버전 1 사용 설명서

AWS Command Line Interface

AWS Command Line Interface: 버전 1 사용 설명서

Table of Contents

이 문서는 AWS CLI의 버전 1에만 해당합니다. AWS CLI의 버전 2와 관련된 문서는 버전 2 사용 설명서를 참조하세요.

AWS Command Line Interface 버전 1이란 무엇인가요?

> **Note**
>
> AWS CLI 버전 1은 AWS CLI의 최신 버전이 아닙니다. 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. 최소한의 구성으로 AWS CLI를 사용하면 터미널 프로그램에 있는 명령 프롬프트에서 브라우저 기반 AWS Management Console에서 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 - Windows의 경우 PowerShell 또는 Windows 명령 프롬프트에서 명령을 실행합니다.
- 원격 – PuTTY 또는 SSH와 같은 원격 터미널 프로그램이나 AWS Systems Manager를 통해 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스에서 명령을 실행합니다.

AWS Management Console의 모든 IaaS(서비스로의 인프라) AWS 관리, 관리 및 액세스 함수는 AWS API 및 AWS CLI에서 사용 가능합니다. 새 AWS IaaS 기능 및 서비스는 출시할 때 또는 출시 후 180일 이내에 API 및 CLI를 통해 전체 AWS Management Console 기능을 제공합니다.

AWS CLI를 사용하면 AWS 서비스의 퍼블릭 API를 직접 액세스할 수 있습니다. AWS CLI를 사용하여 서비스의 기능을 살펴보고 리소스를 관리할 셸 스크립트를 개발할 수 있습니다. 하위 수준 API와 상응한 명령 외에 여러 AWS 서비스에서도 AWS CLI에 대한 사용자 지정 기능을 제공합니다. 사용자 지정에는 복잡한 API와 서비스의 사용을 간소화하는 상위 수준 명령이 포함될 수 있습니다.

AWS CLI 버전 1 정보

AWS CLI 버전 1은 원래 AWS CLI이며, 계속 지원됩니다. 그러나 AWS CLI 버전 2에 도입된 중요한 새로운 기능은 AWS CLI 버전 1로 백포트되지 않을 수 있습니다. 이러한 기능을 사용하려면 AWS CLI 버

전 2를 설치해야 합니다. AWS CLI 버전 1은 Python용 SDK를 사용하여 빌드되므로 호환되는 버전의 Python을 설치해야 합니다.

AWS CLI 버전 1을 설치하려면 AWS CLI 설치 섹션을 참조하세요.

다음 명령을 사용하여 현재 설치된 버전을 점검하세요.

```
$ aws --version
aws-cli/1.40.41 Python/3.11.6 Linux/5.10.205-195.807.amzn2.x86_64 botocore/1.18.6
```

버전 기록은 GitHub의 AWS CLI 버전 1 변경 로그를 참조하세요.

SDK 메이저 버전에 대한 유지 관리 및 지원

SDK 메이저 버전 및 기본 종속성의 유지 관리 및 지원에 대한 자세한 내용은 AWS SDK 및 도구 참조 안내서에서 다음 내용을 참조하세요.

- AWS SDK 및 도구 유지 관리 정책
- AWS SDK 및 도구 버전 지원 매트릭스

Amazon Web Services에 대하여

Amazon Web Services(AWS)는 애플리케이션을 개발할 때 개발자들이 활용할 수 있는 디지털 인프라 서비스의 컬렉션입니다. 이러한 서비스에는 컴퓨팅, 스토리지, 데이터베이스, 애플리케이션 동기화(메시징 및 대기열)가 포함됩니다. AWS에서는 선불형 종량제 서비스 모델을 사용합니다. 사용자 또는 애플리케이션이 사용하는 서비스에 대해서만 청구됩니다. 또한 AWS를 프로토타입 생성 및 실험용 플랫폼으로 더욱 쉽게 이용할 수 있도록 AWS는 프리 티어를 제공합니다. 이 계층에서 특정 사용 수준 미만의 서비스는 무료입니다. AWS 비용 및 프리 티어에 대한 자세한 내용은 AWS 프리 티어를 참조하세요. AWS 계정을 가져오려면 AWS 홈 페이지를 방문하여 AWS 계정 생성을 선택합니다.

AWS CLI 사용 설명서의 예제 관련 정보

이 안내서의 AWS Command Line Interface(AWS CLI) 예제는 다음과 같은 규칙에 따라 서식이 지정됩니다.

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

- 디렉터리 - 특정 디렉터리에서 명령을 실행해야 하는 경우 프롬프트 기호 앞에 디렉터리 이름이 표시됩니다.
- 사용자 입력 - 명령줄에 입력하는 명령 텍스트는 **`user input`**으로 형식이 지정됩니다.
- 대체 가능한 텍스트 – 선택하는 리소스의 이름 또는 명령에 포함시켜야 하는 AWS 서비스에서 생성된 ID를 포함한 변수 텍스트는 *`## ### ###`*로 서식 지정됩니다. 특정 키보드 입력이 필요한 여러 줄 명령에서는 키보드 명령도 대체 가능한 텍스트로 표시될 수 있습니다.
- 출력 – AWS 서비스에서 반환되는 출력은 사용자 입력 아래에 `computer output` 형식으로 표시됩니다.

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

1. 명령줄에서 **`aws configure`**를 입력한 다음 Enter 키를 누릅니다.
2. AWS CLI는 추가 정보를 입력하라고 알리는 텍스트 줄을 출력합니다.
3. 각 액세스 키를 차례로 입력한 다음 Enter(입력)를 누릅니다.
4. 그런 다음, 표시된 형식으로 AWS 리전 이름을 입력하고 Enter를 누른 다음 마지막으로 Enter를 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 그룹 이름 *`my-sg`*를 원하는 보안 그룹 이름으로 바꿀 수 있습니다. 중괄호를 포함한 JSON 문서는 출력입니다. 텍스트 또는 테이블 형식으로 출력할 CLI를 구성하는 경우 출력이 다르게 서식 지정됩니다. 기본 출력 형식은 JSON입니다.

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

AWS CLI에 대한 추가 설명서 및 리소스

AWS CLI 설명서 및 리소스

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

- AWS CLI 버전 1 참조 안내서
- AWS CLI Bash 스크립팅 코드 예제 리포지토리. 오픈 소스 Bash 스크립팅 예제. Bash 스크립팅 예제는 GitHub의 AWS 코드 예제 리포지토리에서 호스팅됩니다.
- AWS CLI GitHub 리포지토리 GitHub에서 AWS CLI의 소스 코드를 보고 포크할 수 있습니다. GitHub의 사용자 커뮤니티에 참여하여 피드백을 제공하고 기능을 요청하며 자체 코드를 기고할 수 있습니다! 여기에는 AWS CLI 설명서의 명령 예제 보기 및 제공이 포함됩니다.
- AWS CLI 별칭 예제 리포지토리 GitHub에서 AWS CLI 별칭 예제를 보고 포크할 수 있습니다.
- AWS CLI 버전 1 변경 로그
- AWS CLI 버전 2 변경 로그

기타 AWS SDK 및 도구

사용 사례에 따라 필요에 더 적합한 AWS SDK 또는 도구 중 하나를 선택할 수 있습니다.

- AWS SDK 및 도구 참조 안내서
- 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

AWS CLI 설치, 업데이트 및 제거

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 새로운 기능은 AWS CLI 버전 1에 추가되어 있지 않을 수 있습니다. 이러한 기능을 사용하려면 AWS CLI 버전 2를 설치해야 합니다. 버전 2 설치 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 2 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 Python용 SDK를 사용하여 빌드되므로 호환되는 버전의 Python을 설치해야 합니다.

Python 버전 지원 매트릭스

AWS CLI version	지원되는 Python 버전
1.39.0~현재	Python 3.9 이상
1.32.0~1.38.38	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

AWS CLI version	지원되는 Python 버전
1.0 ~ 1.16.x	Python 2.6 이상, Python 3.3 이상

AWS CLI의 최신 릴리스에 대한 자세한 내용은 GitHub에서 AWS CLI 버전 2 변경 로그를 참조하세요.

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/1.40.41 Python/3.11.6 Linux/5.10.205-195.807.amzn2.x86_64 botocore/1.18.6
```

Amazon Linux 인스턴스를 언제 생성했는지에 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 version	지원되는 Python 버전
1.39.0~현재	Python 3.9 이상
1.32.0~1.38.38	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 version	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

pip를 사용한 설치, 업데이트 또는 제거

대부분의 Amazon Linux 인스턴스에서는 pip를 사용하여 AWS CLI 버전 1을 사전에 설치합니다.

pip를 사용하여 AWS CLI 버전 1 설치 또는 업데이트

현재 사용자에 대해 AWS CLI 버전 1의 최신 버전을 설치하려면 다음 지침을 사용합니다.

1. Python 버전 3 이상이 설치되어 있으면 `pip3`을 사용하는 것이 좋습니다. `pip3 install`을 사용하여 최신 버전의 AWS CLI 버전 1을 설치하거나 업데이트합니다. 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/1.40.41 Python/3.11.6 Linux/5.10.205-195.807.amzn2.x86_64 botocore/1.18.6
```

pip를 사용하여 AWS CLI 버전 1 제거

AWS CLI를 제거해야 하는 경우 `pip uninstall`을 사용하세요.

```
$ pip3 uninstall awscli
```

yum을 사용한 설치, 업데이트 또는 제거

대부분의 Amazon Linux 2 인스턴스에서는 yum을 사용하여 AWS CLI 버전 1을 사전에 설치합니다.

Amazon Linux에서 yum을 사용하여 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/1.40.41 Python/3.11.6 Linux/5.10.205-195.807.amzn2.x86_64 botocore/1.18.6
```

yum을 사용하여 AWS CLI 버전 1 제거

AWS CLI를 제거하려면 `yum remove`을 사용합니다.

```
$ sudo yum remove awscli
```

AWS CLI 설치 및 제거 오류 문제 해결

AWS CLI를 설치하거나 제거한 후 문제가 발생할 경우 오류 해결에 나온 문제 해결 단계를 참조하세요. 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 CLI를 제거한 후 'aws --version' 명령이 버전을 반환함" 섹션을 참조하세요.

Linux에서 AWS CLI 버전 1 설치, 업데이트 및 제거

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

awscli 패키지는 apt 및 yum과 같은 다른 패키지 관리자를 위한 리포지토리에서 사용할 수 있지만 AWS에서 생성, 관리 또는 지원하지 않습니다. 이 가이드에 설명된 대로 공식 AWS CLI 배포 지점을 통해서 만 AWS를 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 version	지원되는 Python 버전
1.39.0~현재	Python 3.9 이상

AWS CLI version	지원되는 Python 버전
1.32.0~1.38.38	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

번들 설치 관리자를 사용하여 Linux에서 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

- sudo와 함께 제공되는 번들 설치 관리자를 사용하여 AWS CLI 버전 1 설치
- sudo가 없는 번들 설치 관리자를 사용하여 AWS CLI 버전 1 설치
- AWS CLI 버전 1 번들 설치 관리자 제거

sudo와 함께 제공되는 번들 설치 관리자를 사용하여 AWS CLI 버전 1 설치

다음 단계를 수행하여 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 AWS CLI의 경우 다음 명령 블록을 사용하세요.

```
$ 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 AWS CLI의 경우 파일 이름에 하이픈과 버전 번호를 추가하세요. 이 예제의 경우 버전 *1.16.312*의 파일 이름은 `awscli-bundle-1.16.312.zip`이므로 명령은 다음과 같습니다.

```
$ 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 명령줄에서 다음 단계를 수행합니다.

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 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 예제의 경우 버전 *1.16.312*의 파일 이름은 `awscli-bundle-1.16.312.zip`이므로 명령은 다음과 같습니다.

     ```
     $ 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 AWS CLI의 경우 파일 이름에 하이픈과 버전 번호를 추가하세요. 이 예제의 경우 버전 *1.16.312*의 파일 이름은 `awscli-bundle-`*`1.16.312`*`.zip`이므로 URL은 다음과 같습니다. 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 기본 제공된 패키지 관리자를 사용하여 설치하세요.

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

3. 설치 프로그램을 실행합니다. 설치 관리자는 AWS CLI에서 `/usr/local/aws`를 설치하고 aws 디렉터리에 `/usr/local/bin`를 생성합니다. `-b` 옵션을 사용하여 symlink를 생성하면 사용자의 $PATH 변수에 설치 디렉터리를 지정할 필요가 없습니다. 이렇게 하면 모든 사용자가 임의 디렉터리에서 AWS CLI를 입력하여 `aws`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 이 버전을 사용하여 AWS CLI를 설치하려는 경우, 다음과 같이 Python 실행 파일에 대한 절대 경로로 해당 버전의 설치 스크립트를 실행합니다.

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

4. AWS CLI가 올바르게 설치되었는지 확인하세요.

   ```
   $ aws --version
   aws-cli/1.40.41 Python/3.11.6 Linux/5.10.205-195.807.amzn2.x86_64 botocore/1.18.6
   ```

   오류가 발생한 경우 AWS CLI에 대한 오류 문제 해결 섹션을 참조하세요.

sudo가 없는 번들 설치 관리자를 사용하여 AWS CLI 버전 1 설치

sudo 권한이 없거나 현재 사용자에 대해서만 AWS CLI를 설치할 경우, 위 명령의 수정된 버전을 사용할 수 있습니다. 처음 두 명령은 동일합니다.

최신 버전의 AWS CLI의 경우 다음 명령 블록을 사용하세요.

```
$ 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 AWS CLI의 경우 파일 이름에 하이픈과 버전 번호를 추가하세요. 이 예제의 경우 버전 *1.16.312*의 파일 이름은 `awscli-bundle-1.16.312.zip`이므로 명령은 다음과 같습니다.

```
$ 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 대해 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 예제의 경우 버전 *1.16.312*의 파일 이름은 `awscli-bundle-1.16.312.zip`이므로 명령은 다음과 같습니다.

     ```
     $ 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 AWS CLI의 경우 파일 이름에 하이픈과 버전 번호를 추가하세요. 이 예제의 경우 버전 *1.16.312*의 파일 이름은 `awscli-bundle-1.16.312.zip`이므로 URL은 다음과 같습니다. 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 CLI에서 `/usr/local/aws`를 설치하고 aws 디렉터리에 `/usr/local/bin`를 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 하여 AWS CLI를 기본값 위치(`~/.local/lib/aws`)에 설치하고 `~/bin/aws`에 심볼 링크(symlink)를 생성합니다. 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 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. AWS CLI가 올바르게 설치되었는지 확인하세요.

```
$ aws --version
aws-cli/1.40.41 Python/3.11.6 Linux/5.10.205-195.807.amzn2.x86_64 botocore/1.18.6
```

오류가 발생한 경우 AWS CLI에 대한 오류 문제 해결 섹션을 참조하세요.

AWS CLI 버전 1 번들 설치 관리자 제거

1. 번들 설치 관리자를 사용하여 AWS CLI을(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 SDK 및 AWS CLI에서 공유됩니다. 이 폴더를 제거하면 아직 시스템에 있는 AWS SDK에서 액세스할 수 없습니다.

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

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

pip를 사용하여 AWS CLI 버전 1 설치 및 제거

주제

- PIP 설치
- pip를 사용하여 AWS CLI 버전 1 설치 및 업데이트
- pip를 사용하여 AWS CLI 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 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 pip를 ~/.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 $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

   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 스위치, pip는 AWS CLI를 ~/.local/bin에 설치합니다.

   최신 버전의 AWS CLI의 경우 다음 명령 블록을 사용하세요.

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

   특정 버전의 AWS CLI의 경우 파일 이름에 두 개의 등호(=)와 버전 번호를 추가합니다. 이 예제의 경우 버전 *1.16.312*의 파일 이름은 *==1.16.312*이므로 명령은 다음과 같습니다.

   ```
   $ 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. AWS CLI가 올바르게 설치되었는지 확인하세요.

   ```
   $ aws --version
   aws-cli/1.40.41 Python/3.11.6 Linux/5.10.205-195.807.amzn2.x86_64 botocore/1.18.6
   ```

   오류가 발생한 경우 AWS CLI에 대한 오류 문제 해결 섹션을 참조하세요.

pip를 사용하여 AWS CLI 제거

1. pip를 사용하여 AWS CLI 버전 1을 설치한 경우 제거할 때도 pip를 사용해야 합니다.

   ```
   $ pip uninstall awscli
   ```

Python 2 또는 3 버전을 사용하는 경우 `pip2` 또는 `pip3` 명령을 사용해야 할 수 있습니다. `aws --version` 명령을 사용하여 설치된 AWS CLI 버전 1과 관련된 Python 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 SDK 및 AWS CLI에서 공유됩니다. 이 폴더를 제거하면 아직 시스템에 있는 AWS SDK에서 액세스할 수 없습니다.

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

```
$ 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 사용하여 AWS CLI 제거

스냅 설치

아직 `snap`이 설치되어 있지 않은 경우 Canonical Snapcraft에서 제공하는 지침을 사용하여 설치할 수 있습니다. `snap version`을 실행하여 해당 버전의 Linux에 `snap`이 이미 포함되어 있는지 확인합니다.

1. 플랫폼에 Snapcraft를 설치합니다. Snapcraft 설치에 대한 자세한 내용은 Snap 설명서의 데몬 설치를 참조하세요.

2. PATH 변수가 올바르게 업데이트되도록 시스템을 다시 시작합니다. 설치에 문제가 있는 경우 Snap 설명서의 일반적인 문제 해결에 나와 있는 단계를 따르세요.

3. snap이 올바르게 설치되었는지 확인하려면 다음 명령을 실행합니다.

```
$ snap version
```

스냅을 사용하여 AWS CLI 버전 1 설치 및 업데이트

1. AWS CLI 버전 1의 경우 다음 `snap install` 명령을 실행합니다.

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

   권한에 따라 명령에 sudo를 추가해야 할 수도 있습니다.

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

2. AWS CLI가 올바르게 설치되었는지 확인하세요.

```
$ aws --version
aws-cli/1.40.41 Python/3.11.6 Linux/5.10.205-195.807.amzn2.x86_64 botocore/1.18.6
```

   오류가 발생한 경우 AWS CLI에 대한 오류 문제 해결 섹션을 참조하세요.

스냅을 사용하여 AWS CLI 제거

1. snap를 사용하여 AWS CLI 버전 1을 설치한 경우 제거할 때도 snap를 사용해야 합니다.

```
$ 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 SDK 및 AWS CLI에서 공유됩니다. 이 폴더를 제거하면 아직 시스템에 있는 AWS SDK에서 액세스할 수 없습니다.

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

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

AWS CLI 버전 1 실행 파일을 명령줄 경로에 추가

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

다음 명령을 실행하여 pip가 AWS CLI를 설치한 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

AWS CLI를 설치하거나 제거한 후 문제가 발생할 경우 오류 해결에 나온 문제 해결 단계를 참조하세요. 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 CLI를 제거한 후 'aws --version' 명령이 버전을 반환함" 섹션을 참조하세요.

macOS에서 AWS CLI 버전 2 설치, 업데이트 및 제거

번들 설치 관리자 또는 pip를 사용하여 macOS에 AWS Command Line Interface(AWS CLI) 버전 1 및 해당 종속성을 설치할 수 있습니다.

Sections

사전 조건

macOS에 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 version	지원되는 Python 버전
1.39.0~현재	Python 3.9 이상
1.32.0~1.38.38	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

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

- sudo와 함께 제공되는 번들 설치 관리자를 사용하여 AWS CLI 버전 1 설치
- sudo가 없는 번들 설치 관리자를 사용하여 AWS CLI 버전 1 설치
- AWS CLI 버전 1 번들 설치 관리자 제거

sudo와 함께 제공되는 번들 설치 관리자를 사용하여 AWS CLI 버전 1 설치

다음 단계를 수행하여 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 AWS CLI의 경우 다음 명령 블록을 사용하세요.

```
$ 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 AWS CLI의 경우 파일 이름에 하이픈과 버전 번호를 추가하세요. 이 예제의 경우 버전 *1.16.312*의 파일 이름은 `awscli-bundle-1.16.312.zip`이므로 명령은 다음과 같습니다.

```
$ 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 예제의 경우 버전 *1.16.312*의 파일 이름은 `awscli-bundle-1.16.312.zip`이므로 명령은 다음과 같습니다.

     ```
     $ 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 AWS CLI의 경우 파일 이름에 하이픈과 버전 번호를 추가하세요. 이 예제의 경우 버전 *1.16.312*의 파일 이름은 `awscli-bundle-1.16.312.zip`이므로 URL은 다음과 같습니다. https://s3.amazonaws.com/aws-cli/awscli-bundle-1.16.312.zip

2. 패키지에서 파일을 추출(압축 해제)합니다. `unzip`이 없는 경우 macOs 배포의 기본 제공된 패키지 관리자를 사용하여 설치하세요.

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

3. 설치 프로그램을 실행합니다. 설치 관리자는 AWS CLI에서 `/usr/local/aws`를 설치하고 `aws` 폴더에 `/usr/local/bin` symlink를 생성합니다. `-b` 옵션을 사용하여 symlink를 생성하면 사용자의 $PATH 변수에 설치 폴더를 지정할 필요가 없습니다. 이렇게 하면 모든 사용자가 임의 디렉터리에서 AWS CLI를 입력하여 `aws`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 AWS CLI를 설치하려는 경우, 다음과 같이 Python 실행 파일에 대한 절대 경로로 해당 버전의 설치 스크립트를 실행합니다.

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

4. AWS CLI가 올바르게 설치되었는지 확인하세요.

```
$ aws --version
aws-cli/1.40.41 Python/3.11.6 Linux/5.10.205-195.807.amzn2.x86_64 botocore/1.18.6
```

오류가 발생한 경우 AWS CLI에 대한 오류 문제 해결 섹션을 참조하세요.

sudo가 없는 번들 설치 관리자를 사용하여 AWS CLI 버전 1 설치

sudo 권한이 없거나 현재 사용자에 대해서만 AWS CLI를 설치할 경우, 위 명령의 수정된 버전을 사용할 수 있습니다. 처음 두 명령은 동일합니다.

최신 버전의 AWS CLI의 경우 다음 명령 블록을 사용하세요.

```
$ 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 AWS CLI의 경우 파일 이름에 하이픈과 버전 번호를 추가하세요. 이 예제의 경우 버전 *1.16.312*의 파일 이름은 `awscli-bundle-1.16.312.zip`이므로 명령은 다음과 같습니다.

```
$ 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 대해 AWS CLI 버전 1을 설치하는 방법

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 예제의 경우 버전 *1.16.312*의 파일 이름은 `awscli-bundle-1.16.312.zip`이므로 명령은 다음과 같습니다.

     ```
     $ 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 AWS CLI의 경우 파일 이름에 하이픈과 버전 번호를 추가하세요. 이 예제의 경우 버전 *1.16.312*의 파일 이름은 `awscli-bundle-1.16.312.zip`이므로 URL은 다음과 같습니다. https://s3.amazonaws.com/aws-cli/awscli-bundle-1.16.312.zip

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

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

3. 설치 프로그램을 실행합니다. 설치 관리자는 AWS CLI에서 `/usr/local/aws`를 설치하고 aws 디렉터리에 `/usr/local/bin`를 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 하여 AWS CLI를 기본값 위치(`~/.local/lib/aws`)에 설치하고 `~/bin/aws`에 심볼 링크(symlink)를 생성합니다. 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 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. AWS CLI가 올바르게 설치되었는지 확인하세요.

```
$ aws --version
aws-cli/1.40.41 Python/3.11.6 Linux/5.10.205-195.807.amzn2.x86_64 botocore/1.18.6
```

오류가 발생한 경우 AWS CLI에 대한 오류 문제 해결 섹션을 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 SDK 및 AWS CLI에서 공유됩니다. 이 폴더를 제거하면 아직 시스템에 있는 AWS SDK에서 액세스할 수 없습니다.

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

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

pip를 사용하여 AWS CLI 버전 1 설치, 업데이트 및 제거

pip를 직접 사용하여 AWS CLI를 설치할 수 있습니다.

주제

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

PIP 설치

pip가 아직 설치되지 않은 경우 Python Packaging Authority에서 제공하는 스크립트를 사용하여 설치할 수 있습니다. `pip --version`을 실행하여 해당 버전의 Linux에 Python과 pip가 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 pip를 `~/.local/bin` 경로에 설치합니다.

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

pip를 사용하여 AWS CLI 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 AWS CLI의 경우 다음 명령 블록을 사용하세요.

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

   특정 버전의 AWS CLI의 경우 파일 이름에 두 개의 등호(=)와 버전 번호를 추가합니다. 이 예제의 경우 버전 *1.16.312*의 파일 이름은 *==1.16.312*이므로 명령은 다음과 같습니다.

```
$ 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. AWS CLI이 올바르게 설치되었는지 확인합니다.

```
$ aws --version
aws-cli/1.40.41 Python/3.11.6 Darwin/23.3.0 botocore/1.18.6
```

프로그램을 찾을 수 없는 경우 [프로그램을 명령줄 경로에 추가](#)합니다.

AWS CLI 버전 1 실행 파일을 macOS 명령줄 경로에 추가

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 - Python 3.6 및 **pip**가 포함된 macOS(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
```

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 $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 사용하여 AWS CLI 제거

1. `pip`를 사용하여 AWS CLI 버전 1을 설치한 경우 제거할 때도 `pip`를 사용해야 합니다.

```
$ pip uninstall awscli
```

Python 2 또는 3 버전을 사용하는 경우 `pip2` 또는 `pip3` 명령을 사용해야 할 수 있습니다. `aws --version` 명령을 사용하여 설치된 AWS CLI 버전 1과 관련된 Python 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 SDK 및 AWS CLI에서 공유됩니다. 이 폴더를 제거하면 아직 시스템에 있는 AWS SDK에서 액세스할 수 없습니다.

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

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

AWS CLI 설치 및 제거 오류 문제 해결

AWS CLI를 설치하거나 제거한 후 문제가 발생할 경우 오류 해결에 나온 문제 해결 단계를 참조하세요. 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 CLI를 제거한 후 'aws --version' 명령이 버전을 반환함" 섹션을 참조하세요.

Windows에서 AWS CLI 버전 1 설치, 업데이트 및 제거

독립 실행형 설치 관리자(권장) 또는 Python용 패키지 관리자인 `pip`를 사용하여 Windows에 AWS Command Line Interface(AWS CLI) 버전 1을 설치할 수 있습니다.

명령을 입력할 때 프롬프트 기호(`C:\>`)를 포함시키지 마세요. 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 설치할 수 있는 친숙하고 편리한 방법을 제공합니다.

MSI 설치 관리자를 사용하여 AWS CLI 버전 1 설치 및 업데이트

최신 버전이 언제 릴리스되었는지를 알아보려면 GitHub의 릴리스 페이지를 확인하세요. 업데이트가 릴리스되면 설치 프로세스를 반복하여 최신 버전의 AWS CLI 버전 1을 가져와야 합니다.

1. 적절한 MSI 설치 관리자를 다운로드합니다.

   - Windows용 AWS CLI MSI 설치 관리자(64비트): https://s3.amazonaws.com/aws-cli/AWSCLI64PY3.msi
   - Windows용 AWS CLI MSI 설치 관리자(32비트): https://s3.amazonaws.com/aws-cli/AWSCLI32PY3.msi
   - Windows용 AWS CLI 통합 설정 파일: https://s3.amazonaws.com/aws-cli/AWSCLISetup.exe(32비트 및 64비트 MSI 설치 관리자가 모두 포함되어 있으며 자동으로 올바른 버전 설치)
2. 다운로드한 MSI 설치 관리자 또는 설정 파일을 실행하세요.
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 cmd를 검색하여 명령 프롬프트를 시작할 수 있습니다.

   ```
   C:\> aws --version
   aws-cli/1.40.41 Python/3.11.6 Windows/10 botocore/1.18.6
   ```

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

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. AWS CLI를 제거할 것인지 확인합니다.
4. (선택 사항) `.aws` 폴더에서 공유 AWS SDK 및 AWS CLI 설정 정보를 제거합니다.

> **Warning**
>
> 이러한 구성 및 자격 증명 설정은 모든 AWS SDK 및 AWS CLI에서 공유됩니다. 이 폴더를 제거하면 아직 시스템에 있는 AWS SDK에서 액세스할 수 없습니다.

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

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

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

Python Software Foundation은 `pip`가 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 AWS CLI의 경우 다음 명령 블록을 사용하세요.

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

특정 버전의 AWS CLI의 경우 파일 이름에 < 기호와 버전 번호를 추가합니다. 이 예제의 경우 버전 *1.16.312*의 파일 이름은 *<1.16.312*이므로 명령은 다음과 같습니다.

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

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

```
C:\> aws --version
aws-cli/1.40.41 Python/3.11.6 Windows/10 botocore/1.18.6
```

pip를 사용하여 AWS CLI 버전 1 제거

1. `pip`를 사용하여 AWS CLI 버전 1을 설치한 경우 제거할 때도 `pip`를 사용해야 합니다.

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

Python 2 또는 3 버전을 사용하는 경우 `pip2` 또는 `pip3` 명령을 사용해야 할 수 있습니다. `aws --version` 명령을 사용하여 설치된 AWS CLI 버전 1과 관련된 Python 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 SDK 및 AWS CLI에서 공유됩니다. 이 폴더를 제거하면 아직 시스템에 있는 AWS SDK에서 액세스할 수 없습니다.

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

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

AWS CLI 버전 1 실행 파일을 명령줄 경로에 추가

pip를 사용하여 AWS CLI 버전 1을 설치한 후 aws 프로그램을 운영 체제의 PATH 환경 변수에 추가합니다. 이는 MSI 설치를 통해 자동으로 수행됩니다. 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 관리자(64비트) - `C:\Program Files\Amazon\AWSCLI\bin`
- MSI 설치 관리자(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. 경로를 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

AWS CLI를 설치하거나 제거한 후 문제가 발생할 경우 오류 해결에 나온 문제 해결 단계를 참조하세요. 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 CLI를 제거한 후 'aws --version' 명령이 버전을 반환함” 섹션을 참조하세요.

가상 환경에서 AWS CLI 버전 1 설치 및 업데이트

가상 환경에 AWS Command Line Interface(AWS CLI) 버전 1을 설치하면 요구 사항 버전이 다른 `pip` 패키지와 충돌하는 것을 방지할 수 있습니다.

주제

사전 조건

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

  Python 버전 지원 매트릭스

AWS CLI version	지원되는 Python 버전
1.39.0~현재	Python 3.9 이상
1.32.0~1.38.38	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

- `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. 가상 환경에서 AWS CLI 버전 1을 설치하거나 업데이트합니다.

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

5. AWS CLI 버전 1이 올바르게 설치되었는지 확인합니다.

```
$ aws --version
aws-cli/1.40.41 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

AWS CLI를 설치하거나 제거한 후 문제가 발생할 경우 오류 해결에 나온 문제 해결 단계를 참조하세요. 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 CLI를 제거한 후 '`aws --version`' 명령이 버전을 반환함” 섹션을 참조하세요.

AWS CLI 설정 구성

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

- 보안 인증은 API를 호출하는 사람을 식별합니다. 액세스 보안 인증은 AWS 서버에 대한 요청을 암호화하여 ID를 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 섹션을 참조하세요.
- 요청을 처리하는 방법 AWS CLI을 알려주는 기타 구성 세부 정보(예: 기본 출력 형식 및 기본 AWS 리전)

> **Note**
>
> AWS에서는 모든 수신 요청이 암호화 방식으로 서명되어야 합니다. AWS CLI에서 이 작업을 수행합니다. "서명"에는 날짜/시간 스탬프가 포함됩니다. 따라서 컴퓨터의 날짜 및 시간이 올바르게 설정되어야 합니다. 잘못 설정되면 서명의 날짜/시간과 AWS 서비스에서 인식한 날짜/시간의 차이가 극심하여 AWS에서 요청을 거부합니다.

구성 및 보안 인증 우선 순위

보안 인증 및 구성 설정은 시스템 또는 사용자 환경 변수, 로컬 AWS 구성 파일 또는 명령줄에서 파라미터로 명시적으로 선언된 위치 등 다양한 장소에 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 위임 - 구성 또는 `assume-role` 명령을 통해 IAM 역할의 권한을 위임합니다.
4. 웹 ID로 역할 위임 - 구성 또는 `assume-role-with-web-identity` 명령을 통해 웹 ID를 사용하여 IAM 역할의 권한을 위임합니다.
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(Amazon ECS) 태스크 정의에 연결할 수 있습니다. 그러면 작업의 컨테이너에 대해 해당 역할의 임시 보안 인증을 사용할 수 있습니다. 자세한 내용은 Amazon Elastic Container Service 개발자 안내서에서 태스크에 대한 IAM 역할을 참조하세요.
9. Amazon EC2 인스턴스 프로파일 보안 인증 - IAM 역할을 각 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스에 연결할 수 있습니다. 그러면 인스턴스에서 실행되는 코드에 대해 해당 역할의 임시 보안 인증을 사용할 수 있습니다. 보안 인증은 Amazon EC2 메타데이터 서비스를 통해 전달됩니다. 자세한 내용은 Amazon EC2 사용 설명서의 Amazon EC2에 대한 IAM 역할과 IAM 사용 설명서의 인스턴스 프로파일 사용을 참조하세요.

이 섹션의 추가 주제

- the section called “구성 설정”
- the section called “환경 변수”
- the section called “명령줄 옵션”
- the section called “ 명령 완성”
- the section called “재시도”
- the section called “HTTP 프록시”

AWS CLI의 구성 및 보안 인증 파일 설정

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

파일은 `profiles`로 나뉩니다. 기본적으로 AWS CLI는 `default`라는 프로파일에서 확인된 설정을 사용합니다. 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 AWS CLI 설정 구성 섹션을 참조하세요.

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

주제

- 구성 및 보안 인증 파일의 형식
- 구성 설정이 저장되는 장소
- 명명된 프로파일 사용
- 구성 설정 지정 및 보기
- 새 구성 및 보안 인증 설정 명령 예제
- 지원되는 config 파일 설정

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**

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` 파라미터와 함께 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" 섹션을 참조하세요.

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 예는 IAM 역할을 맡기 위한 것입니다. IAM 역할을 사용하는 프로필은 다른 프로필에서 보안 인증을 가져온 다음 IAM 역할 권한을 적용합니다. 다음 예에서 `default`는 보안 인증의 소스 프로

필이며 user1는 동일한 보안 인증을 차용한 다음 새 역할을 맡습니다. 자세한 내용은 the section called "IAM 역할" 섹션을 참조하세요.

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 예는 호스팅 Amazon EC2 인스턴스 메타데이터에서 가져온 보안 인증에 대한 예입니다. 자세한 내용은 the section called "Amazon EC2 메타데이터" 섹션을 참조하세요.

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 목적별 소프트웨어를 개발하거나 실제 데이터로 작업할 때 IAM 사용자를 인증에 사용하지 마세요. 대신 AWS IAM Identity Center과 같은 보안 인증 공급자를 통한 페더레이션을 사용하세요.

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

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**

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 서비스 식별자는 모든 공백을 밑줄로 바꾸고 모든 문자를 소문자로 바꾸는 방식으로 API 모델의 serviceId를 기반으로 합니다. services 섹션에서 사용할 모든 서비스 식별자 키 목록은 AWS CLI에서 엔드포인트 사용을 참조하세요. 서비스 식별자 키 뒤에는 각각 고유한 줄에 공백 두 개로 들여쓰기하여 중첩된 설정이 이어집니다.

다음 예제에서는 *dev* 프로파일에 사용되는 *my-services* 섹션에서 Amazon DynamoDB 서비스에 대한 요청에 사용할 엔드포인트를 구성합니다. 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 AWS CLI에서 엔드포인트 사용을 참조하세요.

프로파일에 역할 기반 보안 인증 정보가 IAM 가정 역할 기능에 대한 source_profile 파라미터를 통해 구성된 경우 SDK는 지정된 프로파일에 대한 서비스 구성만 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
```

프로파일 B를 사용하고 코드에서 Amazon EC2로 호출하는 경우 엔드포인트는 `https://profile-b-ec2-endpoint.aws`로 확인됩니다. 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 장소

AWS CLI는 `aws configure`를 사용하여 지정하는 민감한 보안 인증 정보를 홈 디렉터리의 `credentials`라는 폴더에 있는 `.aws`라는 로컬 파일에 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**
>
> AWS CLI는 `config` 파일에서 보안 인증을 읽을 수 있으므로 모든 프로파일 설정을 단일 파일에 보관할 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. 이들 파일은 다양한 언어의 소프트웨어 개발 키트(SDK)에서도 사용됩니다. SDK 중 하나를 사용하는 경우 AWS CLI는 또한 보안 인증을 자체 파일에 저장해야 하는지 여부를 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 AWS CLI에 대한 환경 변수 구성 섹션을 참조하세요.

AWS Identity and Access Management(IAM) 역할을 지정하는 공유 프로파일을 사용할 때 AWS CLI에서 AWS STS `AssumeRole` 작업을 호출하여 임시 보안 인증을 검색합니다. 이러한 보안 인증은

(~/.aws/cli/cache)에 저장됩니다. 후속 AWS CLI 명령은 캐시된 임시 보안 인증이 만료될 때까지 사용하고 해당 시점에서 AWS CLI가 보안 인증을 자동으로 새로 고칩니다.

명명된 프로파일 사용

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

명명된 프로필을 사용하려면 `--profile` *`profile-name`* 옵션을 명령에 추가합니다. 다음은 `user1` 프로파일에 정의된 보안 인증 및 설정을 사용하여 Amazon EC2 인스턴스를 모두 나열하는 예입니다.

```
$ 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 AWS CLI에 대한 환경 변수 구성 섹션을 참조하세요.

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 set`를 사용하여 보안 인증 또는 구성 설정을 지정할 수 있습니다. `--profile` 설정으로 보거나 수정하려는 프로파일을 지정합니다.

예를 들어 다음 명령은 `region`프로파일에서 `integ`을 설정합니다.

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

설정을 제거하려면 텍스트 편집기에서 `config` 및 `credentials` 파일의 설정을 수동으로 삭제합니다.

aws configure get

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

예를 들어 다음 명령은 `region`프로파일에서 `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 변수에 AWS 리전을 입력하면 이 명령은 사용자가 구성한 리전의 이름, 이 값이 환경 변수에서 가져온 것이라는 점, 환경 변수의 이름을 보여줍니다.

역할 및 IAM Identity Center와 같은 임시 보안 인증 방법의 경우 이 명령은 임시로 캐시된 액세스 키와 비밀 액세스 키를 표시합니다.

```
$ 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" 섹션을 참조하세요.

```
$ 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 예는 IAM 역할을 맡기 위한 것입니다. IAM 역할을 사용하는 프로필은 다른 프로필에서 보안 인증을 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" 섹션을 참조하세요.

```
$ 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

호스팅 Amazon EC2 인스턴스 메타데이터에서 획득한 보안 인증을 사용하는 예입니다. 이 프로세스에는 마법사가 없으므로 `aws configure set` 명령을 사용하여 각 값을 설정합니다. 자세한 내용은 the section called “Amazon EC2 메타데이터” 섹션을 참조하세요.

```
$ 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 목적별 소프트웨어를 개발하거나 실제 데이터로 작업할 때 IAM 사용자를 인증에 사용하지 마세요. 대신 AWS IAM Identity Center과 같은 보안 인증 공급자를 통한 페더레이션을 사용하세요.

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

```
$ 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

- Global settings(글로벌 설정)
- S3 사용자 지정 명령 설정

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 AWS CLI 설정 구성 섹션을 참조하세요.

Global settings(글로벌 설정)

account_id_endpoint_mode

지원되는 AWS 서비스에 대한 호출에 AWS 계정 기반 엔드포인트 ID를 사용할지 여부를 지정합니다. 계정 기반 엔드포인트에 대한 자세한 내용은 the section called “계정 기반 엔드포인트” 섹션을 참조하시기 바랍니다.

이 설정에서는 다음을 설정할 수 있습니다.

- (기본값) **preferred** - 엔드포인트에는 사용 가능한 경우 계정 ID가 포함되어야 합니다.
- **disabled** - 확인된 엔드포인트에는 계정 ID가 포함되지 않습니다.
- **required** - 엔드포인트에는 계정 ID가 포함되어야 합니다. 계정 ID를 사용할 수 없는 경우 SDK에서 오류가 발생합니다.

AWS_ACCOUNT_ID_ENDPOINT_MODE 환경 변수로 재정의할 수도 있습니다. 계정 기반 엔드포인트를 사용하려면 AWS_ACCOUNT_ID 환경 변수 또는 aws_account_id 설정에서 ID를 설정해야 합니다.

```
account_id_endpoint_mode = preferred
```

> 엔드포인트 우선순위
>
> 엔드포인트 구성 설정은 시스템 또는 사용자 환경 변수, 로컬 AWS 구성 파일 또는 명령줄에서 파라미터로 명시적으로 선언된 위치 등 다양한 장소에 있습니다. AWS CLI는 이러한 엔드포인트 설정을 특정 순서로 확인하고 우선순위가 가장 높은 엔드포인트 설정을 사용합니다. 엔드포인트 우선순위 목록은 the section called “엔드포인트 구성 및 설정 우선 순위” 섹션을 참조하시기 바랍니다.

api_versions

일부 AWS 서비스는 이전 버전과의 호환성을 지원하기 위해 여러 개의 API 버전을 유지하고 있습니다. 기본적으로 AWS CLI 명령은 가장 최근에 출시된 API 버전을 사용합니다. config 파일에 api_versions 설정을 포함시켜 프로파일에서 사용할 API 버전을 지정할 수 있습니다.

이는 뒤에 들여 쓴 줄이 하나 이상 나오는 "중첩" 설정으로, 각각의 줄은 하나의 AWS 서비스와 사용할 API 버전을 식별합니다. 사용 가능한 API 버전을 알아보려면 각 서비스의 설명서를 참조하세요.

다음 예제는 두 AWS 서비스에 대한 API 버전을 지정하는 방법을 보여줍니다. 이러한 API 버전은 이러한 설정이 포함된 프로파일 하에서 실행되는 명령에서만 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 보안 인증의 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_account_id

지원되는 AWS 서비스에 대한 호출에 사용할 AWS 계정 기반 엔드포인트 ID를 지정합니다. 계정 기반 엔드포인트에 대한 자세한 내용은 the section called “계정 기반 엔드포인트” 섹션을 참조하시기 바랍니다.

`AWS_ACCOUNT_ID` 환경 변수로 재정의할 수도 있습니다. 이 설정을 사용하려면 `AWS_ACCOUNT_ID_ENDPOINT_MODE` 환경 변수 또는 `account_id_endpoint_mode` 설정을 `preferred` 또는 `required`로 설정해야 합니다.

```
aws_account_id = 123456789EXAMPLE
```

엔드포인트 우선순위

엔드포인트 구성 설정은 시스템 또는 사용자 환경 변수, 로컬 AWS 구성 파일 또는 명령줄에서 파라미터로 명시적으로 선언된 위치 등 다양한 장소에 있습니다. AWS CLI는 이러한 엔드포인트 설정을 특정 순서로 확인하고 우선순위가 가장 높은 엔드포인트 설정을 사용합

니다. 엔드포인트 우선순위 목록은 the section called “엔드포인트 구성 및 설정 우선 순위” 섹션을 참조하시기 바랍니다.

`aws_secret_access_key`

명령 요청을 인증하기 위한 보안 인증의 일부로 사용되는 AWS 비밀 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`

SSL 인증서를 확인하는 데 사용되는 CA 인증서 번들(확장자가 `.pem`인 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`

AWS CLI가 `http://` 또는 `https://`로 시작하는 명령줄 파라미터에서 URL 링크를 따르려고 하는지를 지정합니다. 활성화하면 검색된 콘텐츠가 URL 대신 파라미터 값으로 사용됩니다.

- 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 CLI는 `http://` 또는 `https://`로 시작하는 파라미터 문자열 값을 다른 문자열과 다르게 취급하지 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 AWS CLI에서 `aws` 명령 내역을 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 AWS CLI는 ISO 8601에 따라 모든 타임스탬프의 형식을 변경합니다.

  ISO 8601 형식의 타임스탬프는 다음 예제와 같습니다. 첫 번째 예에서는 시간 뒤에 `Z`를 포함하여 UTC(협정 세계시)로 시간을 보여 줍니다. 날짜와 시간은 `T`로 구분됩니다.

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

  다른 시간대를 지정하려면 `Z` 대신 `+` 또는 `-`를 지정하고 원하는 시간대가 UTC를 기준으로 앞이나 뒤로 몇 시간 차이가 있는지를 두 자리 값으로 지정합니다. 다음 예제에서는 이전 예제와 동일한 시간을 보여 주지만 UTC보다 8시간 뒤인 태평양 표준시로 조정했습니다.

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

- wire – AWS CLI 버전 1의 기본값입니다. 지정된 경우 AWS CLI는 HTTP 쿼리 응답에서 수신된 것과 똑같이 모든 타임스탬프 값을 표시합니다.

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

```
cli_timestamp_format = iso8601
```

credential_process

이 명령에서 사용할 인증 보안 인증을 생성 또는 수신하기 위해 AWS CLI가 실행하는 외부 명령을 지정합니다. 이 명령은 특정 형식으로 보안 인증을 반환해야 합니다. 이 설정을 사용하는 방법에 대한 자세한 내용은 AWS CLI에서 외부 프로세스를 통해 자격 증명 소싱 섹션을 참조하세요.

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

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

credential_source

AWS CLI에서 role_arn 파라미터로 지정된 역할을 수임하는 데 사용하는 보안 인증을 찾을 수 있는 위치를 지정하기 위해 Amazon EC2 인스턴스 또는 컨테이너 내에서 사용됩니다. source_profile과 credential_source 모두를 동일한 프로파일에서 지정할 수 없습니다.

이 파라미터는 다음 세 가지 값 중 하나를 가질 수 있습니다.

- Environment – AWS CLI가 환경 변수의 소스 보안 인증을 검색하도록 지정합니다.
- Ec2InstanceMetadata – AWS CLI가 EC2 인스턴스 프로파일에 연결된 IAM 역할을 사용하여 소스 보안 인증을 가져오도록 지정합니다.
- EcsContainer – AWS CLI가 ECS 컨테이너에 연결된 IAM 역할을 소스 보안 인증으로 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. 자세한 내용은 the section called “모든 AWS 서비스 서비스에 대한 글로벌 엔드포인트 설정” 섹션을 참조하세요.

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

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

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

> **엔드포인트 우선순위**
>
> 엔드포인트 구성 설정은 시스템 또는 사용자 환경 변수, 로컬 AWS 구성 파일 또는 명령줄에서 파라미터로 명시적으로 선언된 위치 등 다양한 장소에 있습니다. AWS CLI는 이러한 엔드포인트 설정을 특정 순서로 확인하고 우선순위가 가장 높은 엔드포인트 설정을 사용합니다. 엔드포인트 우선순위 목록은 [the section called “엔드포인트 구성 및 설정 우선 순위”](#) 섹션을 참조하시기 바랍니다.

`ignore_configure_endpoint_urls`

사용 설정된 경우 AWS CLI는 `config` 파일에 지정된 모든 사용자 지정 엔드포인트 구성을 무시합니다. 유효 값은 **`true`**및 **`false`**입니다.

```
ignore_configure_endpoint_urls = true
```

> **엔드포인트 우선순위**
>
> 엔드포인트 구성 설정은 시스템 또는 사용자 환경 변수, 로컬 AWS 구성 파일 또는 명령줄에서 파라미터로 명시적으로 선언된 위치 등 다양한 장소에 있습니다. AWS CLI는 이러한 엔드포인트 설정을 특정 순서로 확인하고 우선순위가 가장 높은 엔드포인트 설정을 사용합니다. 엔드포인트 우선순위 목록은 [the section called “엔드포인트 구성 및 설정 우선 순위”](#) 섹션을 참조하시기 바랍니다.

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 AWS 리소스에 대한 액세스 권한을 서드 파티에 부여할 때 외부 ID를 사용하는 방법을 참조하세요.

max_attempts

AWS CLI 재시도 핸들러에서 사용하는 최대 재시도 시도 횟수 값을 지정합니다. 여기서 초기 호출은 사용자가 제공한 max_attempts 값에 포함됩니다.

AWS_MAX_ATTEMPTS 환경 변수를 사용하여 이 값을 재정의할 수 있습니다.

```
max_attempts = 3
```

mfa_serial

역할 수임 시 사용하는 MFA 디바이스의 식별 번호입니다. 이는 수임 중인 역할의 신뢰 정책에 MFA 인증을 필요로 하는 조건이 포함된 경우에만 필수입니다. 이 값은 하드웨어 디바이스용 일련 번호(예: GAHT12345678) 또는 가상 MFA 디바이스용 Amazon 리소스 이름(ARN)(예: arn:aws:iam::123456789012:mfa/*user*)일 수 있습니다.

output

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

- **json** - 출력은 JSON 문자열로 형식이 지정됩니다.
- **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 AWS 서비스 엔드포인트에 전달하기 전에 파라미터를 검증할지 여부를 지정합니다.

- true - 기본값입니다. 지정된 경우 AWS CLI는 명령줄 파라미터를 로컬로 검증합니다.
- false – 지정된 경우 AWS CLI는 AWS 서비스 엔드포인트에 전달하기 전에 명령줄 파라미터를 검증하지 않습니다.

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

```
parameter_validation = false
```

`region`

이 프로파일을 사용하여 요청된 명령에서 요청을 전송할 AWS 리전을 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`을 사용하면 리전 엔드포인트 외에 AWS Security Token Service (AWS STS) 및 Amazon Simple Storage Service(Amazon S3)와 같이 글로벌 엔드포인트를 지원하는 서비스에 글로벌 엔드포인트를 지정할 수 있습니다.

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

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

`request_checksum_calculation`

요청 페이로드에 대한 체크섬이 계산되는 시기를 지정하며, 다음과 같은 옵션이 있습니다.

- `when_supported` – (기본값) 요청 페이로드 체크섬은 작업이 서비스 모델에서 체크섬 알고리즘을 지정하거나 요청 체크섬이 필요할 때 계산됩니다.
- `when_required` - 요청 페이로드 체크섬은 작업에 요청 체크섬이 필요하거나 사용자가 AWS 서비스에서 모델링한 `requestAlgorithmMember`를 제공할 때 계산됩니다.

```
request_checksum_calculation = when_supported
```

환경 변수 `AWS_REQUEST_CHECKSUM_CALCULATION`이 이 설정을 재정의합니다.

response_checksum_validation

응답 페이로드에 대한 체크섬 검증이 수행되는 시기를 지정하며, 다음과 같은 옵션이 있습니다.

- when_supported – (기본값) 응답 페이로드 체크섬 검증은 작업이 AWS CLI가 지원하는 서비스 모델에서 응답 알고리즘을 지정할 때 수행됩니다.
- when_required – 응답 페이로드 체크섬 검증은 작업이 AWS CLI가 지원하는 서비스 모델에서 응답 알고리즘을 지정하고 작업 입력에서 모델링된 requestValidationModeMember를 ENABLED로 설정할 때 수행됩니다.

```
response_checksum_validation = when_supported
```

환경 변수 AWS_RESPONSE_CHECKSUM_VALIDATION이 이 설정을 재정의합니다.

retry_mode

AWS CLI에 사용될 재시도 모드를 지정합니다. 레거시(기본값), 표준 및 적응형, 이렇게 3가지 재시도 모드를 사용할 수 있습니다. 재시도에 대한 자세한 내용은 AWS CLI에서 AWS CLI 재시도을 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 RoleSessionName가 AWS CLI 작업을 호출할 때 AssumeRole 파라미터에 제공되고 수임한 역할 사용자 ARN의 일부가 됩니다. 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” 섹션을 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” 섹션을 참조하세요.

sdk_ua_app_id

여러 고객 애플리케이션에서 단일 AWS 계정을 사용하여 AWS 서비스를 호출할 수 있습니다. 애플리케이션 ID는 어떤 소스 애플리케이션이 AWS 서비스를 사용하여 일련의 호출을 수행했는지 식별합니다. AWS SDK와 서비스는 이 값을 고객 커뮤니케이션에 다시 표시하는 것 외에는 이 값을 사용하거나 해석하지 않습니다. 예를 들어, 이 값을 운영 이메일에 포함시켜 알림과 관련된 애플리케이션을 고유하게 식별할 수 있습니다.

애플리케이션 ID는 최대 길이가 50자인 문자열입니다. 문자, 숫자 및 `! $ % & * + - . , ^ _ ` | ~` 등의 특수 문자가 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.

sigv4a_signing_region_set

쉼표로 구분된 목록을 사용하여 SigV4a로 서명할 때 사용할 리전을 지정합니다. 이 변수가 설정되지 않은 경우 AWS CLI는 AWS 서비스에서 사용하는 기본값을 사용합니다. AWS 서비스에 기본값이 없는 경우 `*` 값을 사용하여 모든 리전에서 요청 서명이 유효합니다.

```
sigv4a_signing_region_set = us-west-2, us-east-1
```

SigV4a에 대한 자세한 내용은 IAM 사용 설명서의 AWS Signature Version 4 for API 요청을 참조하세요.

AWS_SIGV4A_SIGNING_REGION_SET 환경 변수를 사용하여 이 설정을 덮어쓸 수 있습니다. 이 값은 명령줄 파라미터로 설정할 수 없습니다.

source_profile

AWS CLI에서 `role_arn` 파라미터로 지정한 역할을 수임하는 데 사용할 수 있는 장기 보안 인증으로 명명된 프로파일을 지정합니다. `source_profile`과 `credential_source` 모두를 동일한 프로파일에서 지정할 수 없습니다.

```
source_profile = production-profile
```

sts_regional_endpoints

AWS CLI에서 AWS 클라이언트가 AWS CLI(AWS Security Token Service)와 통신하는 데 사용하는 AWS STS 서비스 엔드포인트를 결정하는 방법을 지정합니다. AWS CLI 버전 1의 기본값은 `legacy`입니다.

다음 두 값 중 하나를 지정할 수 있습니다.

- **`legacy`** – 글로벌 STS 엔드포인트와 `sts.amazonaws.com`를 다음 AWS 리전에 사용합니다.`ap-northeast-1`, `ap-south-1`, `ap-southeast-1`, `ap-southeast-2`, `aws-global`, ca-central-1, eu-central-1, eu-north-1, eu-west-1, eu-west-2, eu-west-3, sa-east-1, us-east-1, us-east-2, us-west-1, us-west-2. 다른 모든 리전은 해당 리전 엔드포인트를 자동으로 사용합니다.

- **regional** – AWS CLI에서는 현재 구성된 리전에 항상 AWS STS 엔드포인트를 사용합니다. 예를 들어 클라이언트가 us-west-2를 사용하도록 구성된 경우, AWS STS에 대한 모든 호출은 글로벌 엔드포인트 sts.amazonaws.com 대신 리전 엔드포인트 sts.us-west-2.amazonaws.com에 대해 전송됩니다. 이 설정이 활성화된 상태에서 전역 엔드포인트에 요청을 보내려면 Region을 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

이중 스택 엔드포인트를 사용하여 AWS 요청을 보내도록 설정합니다. IPv4 및 IPv6 트래픽을 모두 지원하는 이중 스택 엔드포인트에 대한 자세한 내용은 Amazon 심플 스토리지 서비스 사용 설명서의 Amazon S3 이중 스택 엔드포인트 사용을 참조하세요. 이중 스택 엔드포인트는 일부 리전에 사용할 수 있는 서비스입니다. 서비스 또는 AWS 리전에 대한 이중 스택 엔드포인트가 없는 경우 요청이 실패합니다. 유효한 설정은 true 및 false입니다. 이 옵션은 기본적으로 비활성화되어 있습니다. 자세한 내용은 the section called “모든 AWS 서비스에 이중 스택 엔드포인트를 사용하도록 설정” 섹션을 참조하세요.

이 옵션은 use_accelerate_endpoint 설정에서 함께 사용할 수 없습니다.

> **엔드포인트 우선순위**
>
> 엔드포인트 구성 설정은 시스템 또는 사용자 환경 변수, 로컬 AWS 구성 파일 또는 명령줄에서 파라미터로 명시적으로 선언된 위치 등 다양한 장소에 있습니다. AWS CLI는 이러한 엔드포인트 설정을 특정 순서로 확인하고 우선순위가 가장 높은 엔드포인트 설정을 사용합니다. 엔드포인트 우선순위 목록은 the section called “엔드포인트 구성 및 설정 우선 순위” 섹션을 참조하시기 바랍니다.

use_fips_endpoint

일부 AWS 서비스는 일부 AWS 리전에서 FIPS(Federal Information Processing Standard) 140-2를 지원하는 엔드포인트를 제공합니다. AWS 서비스에서 FIPS를 지원하는 경우 이 설정은 AWS CLI에서 사용해야 하는 FIPS 엔드포인트를 지정합니다. 표준 AWS 엔드포인트와 달리 FIPS 엔드포인트에서는 FIPS 140-2를 준수하는 TLS 소프트웨어 라이브러리를 사용합니다. 이러한 엔드포인트

는 미국 정부와 상호 작용하는 기업에 필요할 수 있습니다. 자세한 내용은 the section called “모든 AWS 서비스에 FIPS 엔드포인트를 사용하도록 설정” 단원을 참조하십시오.

이 설정이 활성화되어 있지만 해당 서비스에 대한 FIPS 엔드포인트가 AWS 리전에 없는 경우 AWS 명령이 실패할 수 있습니다. 이 경우 `--endpoint-url` 옵션을 사용하여 명령에 사용할 엔드포인트를 수동으로 지정하거나 서비스별 엔드포인트를 사용합니다.

> **엔드포인트 우선순위**
>
> 엔드포인트 구성 설정은 시스템 또는 사용자 환경 변수, 로컬 AWS 구성 파일 또는 명령줄에서 파라미터로 명시적으로 선언된 위치 등 다양한 장소에 있습니다. AWS CLI는 이러한 엔드포인트 설정을 특정 순서로 확인하고 우선순위가 가장 높은 엔드포인트 설정을 사용합니다. 엔드포인트 우선순위 목록은 the section called “엔드포인트 구성 및 설정 우선 순위” 섹션을 참조하시기 바랍니다.

`web_identity_token_file`

보안 인증 공급자가 제공하는 OAuth 2.0 액세스 토큰 또는 OpenID Connect ID 토큰을 포함하는 파일의 경로를 지정합니다. AWS CLI에서 이 파일의 내용을 로드하고 해당 파일을 `WebIdentityToken` 작업에 대한 `AssumeRoleWithWebIdentity` 인수로 전달합니다.

환경 변수 `AWS_WEB_IDENTITY_TOKEN_FILE`이 이 설정을 재정의합니다.

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

`tcp_keepalive`

AWS CLI 클라이언트가 TCP keep-alive 패킷을 사용할 것인지 여부를 지정합니다.

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

```
tcp_keepalive = false
```

S3 사용자 지정 명령 설정

Amazon S3는 AWS CLI에서 Amazon S3 작업을 수행하는 방법을 구성하는 몇 가지 설정을 지원합니다. 일부 설정은 `s3api` 및`s3` 네임스페이스의 모든 S3 명령에 적용됩니다. 다른 설정은 특별히 공통 작

업을 추상화하고 API 작업에 대한 일대일 매핑을 한 번 이상 수행하는 S3 "사용자 지정" 명령을 위한 것입니다. `aws s3` 이전 명령인 `cp`, `sync`, `mv` 및 `rm`에는 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. 버킷 이름이 호스트 이름에 있는지 또는 URL의 일부분인지 여부를 제어합니다. 유효 값은 `path`, `virtual` 및 `auto`입니다. 기본값은 `auto`입니다.

Amazon S3 엔드포인트는 두 가지 방식으로 구성할 수 있습니다. 첫 번째는 `virtual`이라고 하는 방식으로 호스트 이름의 일부로 버킷 이름을 포함하고 있습니다. 예를 들면 `https://bucketname.s3.amazonaws.com`입니다. 아니면 `path` 방식을 사용하여 마치 URI의 경로인 것처럼 버킷 이름을 취급합니다(예: `https://s3.amazonaws.com/bucketname`). CLI은 `auto`를 사용하도록 기본 설정되어 있는데, 가능한 경우 `virtual` 방식을 사용하되, 필요할 때는 `path` 방식으로 돌아갑니다. 예를 들어 DNS를 준수하지 않은 버킷 이름은 호스트 이름의 일부가 될

수 없고 해당 경로에 있어야 합니다. auto에서는 CLI가 이 조건을 탐지하여 사용자를 대신해 path 방식으로 자동 전환합니다. 주소 지정 방식을 path로 설정한 경우에는 AWS에서 구성한 AWS CLI 리전이 버킷의 리전과 일치하는지 확인해야 합니다.

payload_signing_enabled

SHA256이 sigv4 페이로드를 서명하는지 여부를 지정합니다. HTTPS를 사용할 때 업로드(UploadPart 및 PutObject)를 스트리밍하기 위해 기본적으로 이 옵션이 비활성화되어 있습니다. 업로드(UploadPart 및 PutObject) 스트리밍을 위해 이 값이 기본적으로 false로 설정됩니다. 단, ContentMD5가 존재하고(기본적으로 생성) 엔드포인트가 HTTPS를 사용하는 경우에 한합니다.

true로 설정되어 있는 경우, S3 요청은 사용자를 위해 계산되어 요청 서명에 포함되어 있는 SHA256 체크섬의 형태로 추가적인 내용 확인을 받게 됩니다. 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 설정하는 경우 AWS CLI는 모든 Amazon S3 요청을 s3-accelerate.amazonaws.com의 S3 Accelerate 엔드포인트로 전달됩니다. 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.

use_dualstack_endpoint

이중 스택 엔드포인트를 사용하여 s3 및 s3api 요청을 보내도록 사용 설정합니다. IPv4 및 IPv6 트래픽을 모두 지원하는 이중 스택 엔드포인트에 대한 자세한 내용은 Amazon 심플 스토리지 서비스 사용 설명서의 Amazon S3 이중 스택 엔드포인트 사용을 참조하세요. 이중 스택 엔드포인트는 일부 리전에 사용할 수 있는 서비스입니다. 서비스 또는 AWS 리전에 대한 이중 스택 엔드포인트가 없는 경우 요청이 실패합니다. 유효한 설정은 true 및 false입니다. 이 옵션은 기본적으로 비활성화되어 있습니다. 자세한 내용은 the section called “모든 AWS 서비스에 이중 스택 엔드포인트를 사용하도록 설정” 섹션을 참조하세요.

이 옵션은 `use_accelerate_endpoint` 설정에서 함께 사용할 수 없습니다.

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

일반적으로 먼저 `max_concurrent_requests`를 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_requests`가 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, `aws s3 cp localdir s3://bucket/ --recursive` 명령을 사용하여 S3 버킷에 파일을 업로드하면 AWS CLI는 `localdir/file1`, `localdir/file2` 및 `localdir/file3` 파일을 동시에 업로드할 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.

AWS CLI는 내부적으로 Amazon S3 태스크를 대기열에 추가한 후 소비자를 통해 작업을 실행하는 모델을 사용합니다. 태스크 수는 `max_concurrent_requests`로 제한됩니다. 태스크는 보통 단일 Amazon S3 작업에 매핑됩니다. 예를 들어 작업은 `PutObjectTask`, `GetObjectTask` 또는 `UploadPartTask`가 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. 또한 이 설정은 AWS CLI가 실행이 필요하다고 인식하고 있는 태스크 수에 해당됩니다. 이는 곧 AWS CLI가 기본적으로 1,000건의 태스크만 미리 볼 수 있다는 것을 의미합니다. 이 값을 늘리면 AWS CLI에서 대기열 저장 속도가 태스크 완료 속도보다 빠르다는 가정하에 필요한 총 태스크 수를 보다 신속하게 알 수 있습니다. max_queue_size가 커질수록 메모리가 더 필요하게 된다는 단점이 있습니다.

multipart_chunksize

개별 파일의 멀티파트 이전을 위해 AWS CLI가 사용하는 청크 크기를 지정합니다. 기본값은 8MB이며 최소 5MB입니다.

파일 전송이 `multipart_threshold`를 초과하면 AWS CLI는 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 이전을 위해 AWS CLI가 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. 예를 들어 `1048576`입니다.

- 두 번째는 크기 접미사가 포함된 파일 크기입니다. KB, MB, GB 또는 TB를 사용할 수 있습니다. 예시: 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.

AWS CLI에 대한 환경 변수 구성

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 명령줄에서 파라미터를 사용하여 옵션을 지정할 경우, 구성 파일에서 해당하는 환경 변수 또는 프로파일의 값을 재정의합니다.

우선 순위 및 AWS CLI에서 사용할 보안 인증을 결정하는 방법에 대한 자세한 내용은 AWS CLI 설정 구성 섹션을 참조하세요.

주제

- 환경 변수를 설정하는 방법
- AWS CLI 지원되는 환경 변수

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 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. 설정을 로드하려면 터미널을 다시 시작해야 할 수 있습니다.

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 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 예에 표시된 대로 PowerShell 프롬프트에서 환경 변수를 설정하면 현재 세션 기간에만 해당 값이 저장됩니다. 모든 PowerShell 및 명령 프롬프트 세션에서 환경 변수 설정이 영구적으로 적용되도록 하려면 제어판에서 시스템 애플리케이션을 사용하여 해당 설정을 저장합니다. 또는 PowerShell 프로파일에 변수를 추가하여 향후 모든 PowerShell 세션에 적용되도록 변수를 설정할 수 있습니다. 환경 변수 저장 또는 세션에 영구적 적용에 대한 자세한 내용은 PowerShell 설명서를 참조하십시오.

AWS CLI 지원되는 환경 변수

AWS CLI는 다음과 같은 환경 변수를 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_ACCOUNT_ID

지원되는 AWS 서비스에 대한 호출에 사용할 AWS 계정 기반 엔드포인트 ID를 지정합니다. 계정 기반 엔드포인트에 대한 자세한 내용은 the section called "계정 기반 엔드포인트" 섹션을 참조하시기 바랍니다.

이 설정은 aws_account_id 설정을 재정의합니다. 이 설정을 사용하려면 AWS_ACCOUNT_ID_ENDPOINT_MODE 환경 변수 또는 account_id_endpoint_mode 설정을 preferred 또는 required로 설정해야 합니다.

> 엔드포인트 우선순위
>
> 엔드포인트 구성 설정은 시스템 또는 사용자 환경 변수, 로컬 AWS 구성 파일 또는 명령줄에서 파라미터로 명시적으로 선언된 위치 등 다양한 장소에 있습니다. AWS CLI는 이러한 엔드포인트 설정을 특정 순서로 확인하고 우선순위가 가장 높은 엔드포인트 설정을 사용합니다. 엔드포인트 우선순위 목록은 the section called "엔드포인트 구성 및 설정 우선 순위" 섹션을 참조하시기 바랍니다.

AWS_ACCOUNT_ID_ENDPOINT_MODE

지원되는 AWS 서비스에 대한 호출에 AWS 계정 기반 엔드포인트 ID를 사용할지 여부를 지정합니다. 계정 기반 엔드포인트에 대한 자세한 내용은 the section called "계정 기반 엔드포인트" 섹션을 참조하시기 바랍니다.

이 설정에서는 다음을 설정할 수 있습니다.

- (기본값) **preferred** - 엔드포인트에는 사용 가능한 경우 계정 ID가 포함되어야 합니다.
- **disabled** - 확인된 엔드포인트에는 계정 ID가 포함되지 않습니다.

- **required** - 엔드포인트에는 계정 ID가 포함되어야 합니다. 계정 ID를 사용할 수 없는 경우 SDK에서 오류가 발생합니다.

이 설정은 `account_id_endpoint_mode` 설정을 재정의합니다. 계정 기반 엔드포인트를 사용하려면 `AWS_ACCOUNT_ID` 환경 변수 또는 `aws_account_id` 설정에서 ID를 설정해야 합니다.

> 엔드포인트 우선순위
>
> 엔드포인트 구성 설정은 시스템 또는 사용자 환경 변수, 로컬 AWS 구성 파일 또는 명령줄에서 파라미터로 명시적으로 선언된 위치 등 다양한 장소에 있습니다. AWS CLI는 이러한 엔드포인트 설정을 특정 순서로 확인하고 우선순위가 가장 높은 엔드포인트 설정을 사용합니다. 엔드포인트 우선순위 목록은 the section called “엔드포인트 구성 및 설정 우선 순위” 섹션을 참조하시기 바랍니다.

`AWS_CA_BUNDLE`

HTTPS 인증서 확인에 사용할 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`

`s3 mv` 명령을 사용할 때 소스 버킷과 대상 버킷이 동일한 경우 소스 파일이나 객체를 그 자체로 옮길 수 있어 실수로 소스 파일이나 객체가 삭제될 수 있습니다. `AWS_CLI_S3_MV_VALIDATE_SAME_S3_PATHS` 환경 변수 및 `--validate-same-s3-paths` 옵션은 Amazon S3 소스 또는 대상 URI에서 액세스 포인트 ARN 또는 액세스 포인트 별칭의 유효성을 검사할지 여부를 지정합니다.

> Note
>
> `s3 mv`에 대한 경로 유효성 검사에는 추가 API 호출이 필요합니다.

`AWS_CONFIG_FILE`

AWS CLI에서 구성 프로파일을 저장하는 데 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 때 기본 제공 검색 경로인 `~/.aws/models` 외부에서 확인할 추가 디렉터리의 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

`Default region name`은 기본적으로 요청을 전송할 서버가 있는 AWS 리전을 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**
>
> AWS를 사용하여 명시적으로 또는 기본 리전을 설정하여 AWS CLI 리전을 지정해야 합니다. 사용 가능한 리전 목록은 리전 및 엔드포인트를 참조하세요. AWS CLI에서 사용하는 리전 표기는 AWS Management Console URL 및 서비스 엔드포인트에서 사용하는 것과 동일한 이름입니다.

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

AWS_EC2_METADATA_DISABLED

Amazon EC2 인스턴스 메타데이터 서비스(IMDS)사용을 비활성화합니다.

true로 설정하면 IMDS에서 사용자 보안 인증 또는 구성(예: 리전)을 요청하지 않습니다.

AWS_ENDPOINT_URL

모든 서비스 요청에 사용되는 엔드포인트를 지정합니다. 자세한 내용은 the section called “모든 AWS 서비스 서비스에 대한 글로벌 엔드포인트 설정” 섹션을 참조하세요.

> 엔드포인트 우선순위
>
> 엔드포인트 구성 설정은 시스템 또는 사용자 환경 변수, 로컬 AWS 구성 파일 또는 명령줄에서 파라미터로 명시적으로 선언된 위치 등 다양한 장소에 있습니다. AWS CLI는 이러한 엔드포인트 설정을 특정 순서로 확인하고 우선순위가 가장 높은 엔드포인트 설정을 사용합니다. 엔드포인트 우선순위 목록은 the section called “엔드포인트 구성 및 설정 우선 순위” 섹션을 참조하시기 바랍니다.

`AWS_ENDPOINT_URL_<SERVICE>`

특정 서비스에 사용되는 사용자 지정 엔드포인트를 지정하며, 여기서 `<SERVICE>`가 AWS 서비스 식별자로 대체됩니다. 예를 들어 Amazon DynamoDB의 `serviceId`는 `DynamoDB`입니다. 이 서비스의 경우 엔드포인트 URL 환경 변수는 `AWS_ENDPOINT_URL_DYNAMODB`입니다.

모든 서비스별 환경 변수 목록은 서비스별 식별자 목록을 참조하세요.

> 엔드포인트 우선순위
>
> 엔드포인트 구성 설정은 시스템 또는 사용자 환경 변수, 로컬 AWS 구성 파일 또는 명령줄에서 파라미터로 명시적으로 선언된 위치 등 다양한 장소에 있습니다. AWS CLI는 이러한 엔드포인트 설정을 특정 순서로 확인하고 우선순위가 가장 높은 엔드포인트 설정을 사용합니다. 엔드포인트 우선순위 목록은 the section called “엔드포인트 구성 및 설정 우선 순위” 섹션을 참조하시기 바랍니다.

`AWS_IGNORE_CONFIGURED_ENDPOINT_URLS`

활성화하면 AWS CLI는 모든 사용자 지정 엔드포인트 구성을 무시합니다. 유효 값은 **`true`**및 **`false`**입니다. 자세한 내용은 the section called “모든 AWS 서비스 서비스에 대한 글로벌 엔드포인트 설정” 섹션을 참조하세요.

엔드포인트 구성 설정은 시스템 또는 사용자 환경 변수, 로컬 AWS 구성 파일 또는 명령줄에서 파라미터로 명시적으로 선언된 위치 등 다양한 장소에 있습니다. 엔드포인트 우선순위는 the section called “엔드포인트 구성 및 설정 우선 순위” 섹션을 참조하시기 바랍니다.

> 엔드포인트 우선순위
>
> 엔드포인트 구성 설정은 시스템 또는 사용자 환경 변수, 로컬 AWS 구성 파일 또는 명령줄에서 파라미터로 명시적으로 선언된 위치 등 다양한 장소에 있습니다. AWS CLI는 이러한

엔드포인트 설정을 특정 순서로 확인하고 우선순위가 가장 높은 엔드포인트 설정을 사용합니다. 엔드포인트 우선순위 목록은 the section called “엔드포인트 구성 및 설정 우선 순위” 섹션을 참조하시기 바랍니다.

AWS_MAX_ATTEMPTS

AWS CLI 재시도 핸들러에서 사용하는 최대 재시도 횟수 값을 지정합니다. 여기서 초기 호출은 사용자가 제공한 값에 포함됩니다. 재시도에 대한 자세한 내용은 AWS CLI에서 AWS CLI 재시도을 참조하세요.

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

AWS_METADATA_SERVICE_NUM_ATTEMPTS

IAM 역할로 구성된 Amazon EC2 인스턴스에서 보안 인증을 검색할 때 AWS CLI는 인스턴스 메타데이터 서비스에서 보안 인증의 검색을 한 번 시도한 후 중지합니다. Amazon EC2 인스턴스에서 명령이 실행되는 경우 AWS CLI가 검색을 포기하기 전에 여러 번 재시도하도록 이 값을 늘릴 수 있습니다.

AWS_METADATA_SERVICE_TIMEOUT

인스턴스 메타데이터 서비스에 대한 연결 시간이 초과되기까지 경과하는 시간(초)입니다. IAM 역할로 구성된 Amazon EC2 인스턴스에서 보안 인증을 검색하려는 경우 기본적으로 1초 후에 인스턴스 메타데이터 서비스에 대한 연결 시간이 초과됩니다. IAM 역할이 구성된 Amazon EC2 인스턴스에서 실행 중인 경우 필요에 따라 이 값을 늘릴 수 있습니다.

AWS_PROFILE

사용할 보안 인증 정보와 옵션이 있는 AWS CLI 프로파일의 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_REQUEST_CHECKSUM_CALCULATION

요청 페이로드에 대한 체크섬이 계산되는 시기를 지정하며, 다음과 같은 옵션이 있습니다.

- `when_supported` – (기본값) 요청 페이로드 체크섬은 작업이 서비스 모델에서 체크섬 알고리즘을 지정하거나 요청 체크섬이 필요할 때 계산됩니다.
- `when_required` - 요청 페이로드 체크섬은 작업에 요청 체크섬이 필요하거나 사용자가 AWS 서비스에서 모델링한 `requestAlgorithmMember`를 제공할 때 계산됩니다.

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

AWS_RESPONSE_CHECKSUM_VALIDATION

응답 페이로드에 대한 체크섬 검증이 수행되는 시기를 지정하며, 다음과 같은 옵션이 있습니다.

- `when_supported` – (기본값) 응답 페이로드 체크섬 검증은 작업이 AWS CLI가 지원하는 서비스 모델에서 응답 알고리즘을 지정할 때 수행됩니다.
- `when_required` – 응답 페이로드 체크섬 검증은 작업이 AWS CLI가 지원하는 서비스 모델에서 응답 알고리즘을 지정하고 작업 입력에서 모델링된 `requestValidationModeMember`를 `ENABLED`로 설정할 때 수행됩니다.

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

AWS_RETRY_MODE

AWS CLI에 사용될 재시도 모드를 지정합니다. 레거시(기본값), 표준 및 적응형, 이렇게 3가지 재시도 모드를 사용할 수 있습니다. 재시도에 대한 자세한 내용은 AWS CLI에서 AWS CLI 재시도을 참조하세요.

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

AWS_ROLE_ARN

AWS CLI 명령을 실행하는 데 사용할 IAM 역할(웹 보안 인증 공급자 포함)의 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 RoleSessionName가 AWS CLI 작업을 호출할 때 AssumeRole 파라미터에 제공되고 수임한 역할 사용자 ARN의 일부가 됩니다. `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.

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 AWS 계정을 사용하여 AWS 서비스를 호출할 수 있습니다. 애플리케이션 ID는 어떤 소스 애플리케이션이 AWS 서비스를 사용하여 일련의 호출을 수행했는지 식별합니다. AWS SDK와 서비스는 이 값을 고객 커뮤니케이션에 다시 표시하는 것 외에는 이 값을 사용하거나 해석하지 않습니다. 예를 들어, 이 값을 운영 이메일에 포함시켜 알림과 관련된 애플리케이션을 고유하게 식별할 수 있습니다.

기본적으로 이 값은 없습니다.

애플리케이션 ID는 최대 길이가 50자인 문자열입니다. 문자, 숫자 및 다음과 같은 특수 문자를 사용할 수 있습니다.

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

정의된 경우 이 환경 변수는 sdk_ua_app_id 프로파일 설정 값을 재정의합니다. 명령줄 옵션으로 애플리케이션을 지정할 수는 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`

AWS CLI에서 액세스 키를 저장하는 데 사용하는 파일의 위치를 지정합니다. 기본 경로는 `~/.aws/credentials`입니다.

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

`AWS_SIGV4A_SIGNING_REGION_SET`

쉼표로 구분된 목록을 사용하여 SigV4a로 서명할 때 사용할 리전을 지정합니다. 이 변수가 설정되지 않은 경우 AWS CLI는 AWS 서비스에서 사용하는 기본값을 사용합니다. AWS 서비스에 기본값이 없는 경우 `*` 값을 사용하여 모든 리전에서 요청 서명이 유효합니다.

SigV4a에 대한 자세한 내용은 IAM 사용 설명서의 AWS Signature Version 4 for API 요청을 참조하세요.

정의된 경우 이 환경 변수는 `sigv4a_signing_region_set` 프로파일 설정 값을 재정의합니다.

`AWS_STS_REGIONAL_ENDPOINTS`

AWS CLI에서 AWS 클라이언트가 AWS CLI(AWS Security Token Service)와 통신하는 데 사용하는 AWS STS 서비스 엔드포인트를 결정하는 방법을 지정합니다. AWS CLI 버전 1의 기본값은 `legacy`입니다.

다음 두 값 중 하나를 지정할 수 있습니다.

- **`legacy`** – 글로벌 STS 엔드포인트와 `sts.amazonaws.com`를 다음 AWS 리전에 사용합니다.`ap-northeast-1`, `ap-south-1`, `ap-southeast-1`, `ap-southeast-2`, `aws-global`, `ca-central-1`, `eu-central-1`, `eu-north-1`, `eu-west-1`, `eu-west-2`, `eu-west-3`, `sa-east-1`, `us-east-1`, `us-east-2`, `us-west-1`, `us-west-2`. 다른 모든 리전은 해당 리전 엔드포인트를 자동으로 사용합니다.
- **`regional`** – AWS CLI에서는 현재 구성된 리전에 항상 AWS STS 엔드포인트를 사용합니다. 예를 들어 클라이언트가 `us-west-2`를 사용하도록 구성된 경우, AWS STS에 대한 모든 호출은 글로벌 엔드포인트 `sts.amazonaws.com` 대신 리전 엔드포인트 `sts.us-`

`west-2.amazonaws.com`에 대해 전송됩니다. 이 설정이 활성화된 상태에서 전역 엔드포인트에 요청을 보내려면 Region을 `aws-global`로 설정하면 됩니다.

`AWS_USE_DUALSTACK_ENDPOINT`

이중 스택 엔드포인트를 사용하여 AWS 요청을 보내도록 설정합니다. IPv4 및 IPv6 트래픽을 모두 지원하는 이중 스택 엔드포인트에 대한 자세한 내용은 Amazon 심플 스토리지 서비스 사용 설명서의 Amazon S3 이중 스택 엔드포인트 사용을 참조하세요. 이중 스택 엔드포인트는 일부 리전에 사용할 수 있는 서비스입니다. 서비스 또는 AWS 리전에 대한 이중 스택 엔드포인트가 없는 경우 요청이 실패합니다. 이 옵션은 기본적으로 비활성화되어 있습니다. 자세한 내용은 the section called "모든 AWS 서비스에 이중 스택 엔드포인트를 사용하도록 설정" 섹션을 참조하세요.

> **엔드포인트 우선순위**
>
> 엔드포인트 구성 설정은 시스템 또는 사용자 환경 변수, 로컬 AWS 구성 파일 또는 명령줄에서 파라미터로 명시적으로 선언된 위치 등 다양한 장소에 있습니다. AWS CLI는 이러한 엔드포인트 설정을 특정 순서로 확인하고 우선순위가 가장 높은 엔드포인트 설정을 사용합니다. 엔드포인트 우선순위 목록은 the section called "엔드포인트 구성 및 설정 우선 순위" 섹션을 참조하시기 바랍니다.

`AWS_USE_FIPS_ENDPOINT`

일부 AWS 서비스는 일부 AWS 리전에서 FIPS(Federal Information Processing Standard) 140-2를 지원하는 엔드포인트를 제공합니다. AWS 서비스에서 FIPS를 지원하는 경우 이 설정은 AWS CLI에서 사용해야 하는 FIPS 엔드포인트를 지정합니다. 표준 AWS 엔드포인트와 달리 FIPS 엔드포인트에서는 FIPS 140-2를 준수하는 TLS 소프트웨어 라이브러리를 사용합니다. 이러한 엔드포인트는 미국 정부와 상호 작용하는 기업에 필요할 수 있습니다. 자세한 내용은 the section called "모든 AWS 서비스에 FIPS 엔드포인트를 사용하도록 설정" 단원을 참조하십시오.

이 설정이 활성화되어 있지만 해당 서비스에 대한 FIPS 엔드포인트가 AWS 리전에 없는 경우 AWS 명령이 실패할 수 있습니다. 이 경우 `--endpoint-url` 옵션을 사용하여 명령에 사용할 엔드포인트를 수동으로 지정하거나 서비스별 엔드포인트를 사용합니다.

> **엔드포인트 우선순위**
>
> 엔드포인트 구성 설정은 시스템 또는 사용자 환경 변수, 로컬 AWS 구성 파일 또는 명령줄에서 파라미터로 명시적으로 선언된 위치 등 다양한 장소에 있습니다. AWS CLI는 이러한 엔드포인트 설정을 특정 순서로 확인하고 우선순위가 가장 높은 엔드포인트 설정을 사용합

니다. 엔드포인트 우선순위 목록은 the section called “엔드포인트 구성 및 설정 우선 순위” 섹션을 참조하시기 바랍니다.

AWS_WEB_IDENTITY_TOKEN_FILE

보안 인증 공급자가 제공하는 OAuth 2.0 액세스 토큰 또는 OpenID Connect ID 토큰을 포함하는 파일의 경로를 지정합니다. AWS CLI에서 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” 섹션을 참조하세요.

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

AWS CLI의 명령줄 옵션

AWS CLI에서 명령줄 옵션은 기본 구성 설정, 해당 프로파일 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
- AWS CLI에서 지원되는 전역 명령줄 옵션
- 명령줄 옵션의 일반적인 용도

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 AWS CLI에서 파라미터 값 지정 섹션을 참조하세요.

AWS CLI에서 지원되는 전역 명령줄 옵션

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

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

SSL 인증서를 확인할 때 사용할 CA(인증 기관) 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. 기본값은 60초입니다.

--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. 기본값은 60초입니다.

--color ***<string>***

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

--디버그

디버그 로깅을 활성화하는 부울 스위치입니다. 기본적으로 AWS CLI는 명령 출력의 명령 결과와 관련된 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.txt를 추가합니다. 여기서 debug.txt는 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
```

--endpoint-url *<string>*

요청을 전송할 URL을 지정합니다. 대부분의 명령에서는 AWS CLI가 선택된 서비스 및 지정된 AWS 리전을 기반으로 자동으로 URL을 결정합니다. 하지만 일부 명령에서는 계정별 URL을 지정해야 합니다. 일부 AWS 서비스를 구성하여 프라이빗 VPC 내에서 직접 엔드포인트를 호스팅할 수도 있습니다. 이렇게 하려면 지정이 필요합니다.

다음 명령 예제에서는 사용자 지정 Amazon S3 엔드포인트 URL을 사용합니다.

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

> **엔드포인트 우선순위**
>
> 엔드포인트 구성 설정은 시스템 또는 사용자 환경 변수, 로컬 AWS 구성 파일 또는 명령줄에서 파라미터로 명시적으로 선언된 위치 등 다양한 장소에 있습니다. AWS CLI는 이러한 엔드포인트 설정을 특정 순서로 확인하고 우선순위가 가장 높은 엔드포인트 설정을 사용합니다. 엔드포인트 우선순위 목록은 the section called “엔드포인트 구성 및 설정 우선 순위” 섹션을 참조하시기 바랍니다.

--no-paginate

출력의 페이지 매김을 생성하는 모든 명령 결과를 수신하기 위해 AWS CLI가 자동으로 수행하는 다중 호출을 비활성화하는 부울 스위치입니다. 즉, 출력의 첫 번째 페이지만 표시됩니다.

--no-sign-request

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

--no-verify-ssl

기본적으로 AWS CLI는 AWS 서비스와 통신할 때 SSL을 사용합니다. AWS CLI는 각 SSL 연결 및 호출에 대해 SSL 인증서를 확인합니다. 이 옵션을 사용하면 SSL 인증서를 확인하는 기본 동작이 재정의됩니다.

> **Warning**
>
> 이 옵션은 모범 사례가 아닙니다. `--no-verify-ssl`을 사용하는 경우 클라이언트와 AWS 서비스 간 트래픽이 더 이상 보호되지 않습니다. 즉, 트래픽이 보안 위험이 되며 중간자 공격에 취약합니다. 인증서에 문제가 있는 경우 대신 해당 문제를 해결하는 것이 좋습니다. 인증서 문제 해결 단계는 the section called “SSL 인증서 오류” 섹션을 참조하세요.

--output *<string>*

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

- **`json`** - 출력은 JSON 문자열로 형식이 지정됩니다.
- **`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>
```

--query *<string>*

응답 데이터를 필터링할 때 사용할 JMESPath 쿼리를 지정합니다. 자세한 내용은 AWS CLI에서 출력 필터링 섹션을 참조하세요.

--region *<string>*

이 명령의 AWS 요청을 전송할 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                                       |||
...
```

AWS CLI에서 명령 완성 구성

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

주제

- 작동 방식
- Linux 또는 macOS에서 명령 완성 구성
- Windows에서 명령 완료 구성

작동 방식

명령, 파라미터 또는 옵션을 부분적으로 입력하면 명령 완성 기능이 자동으로 명령을 완성하거나 제안된 명령 목록을 표시합니다. 명령 완성 메시지를 표시하려면 명령을 부분적으로 입력하고 완성 키(대부분의 셸에서 *Tab* 키)를 누릅니다.

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

- 명령을 부분적으로 입력하고 *Tab* 키를 눌러 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 *Tab* 키를 눌러 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 *Tab* 키를 눌러 제안된 리소스 값 목록을 표시합니다. 이 기능은 AWS CLI 버전 2에서만 사용할 수 있습니다.

```
$ aws dynamodb 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**
>
> Amazon Linux를 실행하는 Amazon EC2 인스턴스에서는 기본적으로 명령 완성이 자동으로 구성되고 활성화됩니다.

주제

경로에 completer 폴더가 있는지 확인

AWS completer가 성공적으로 작동하려면 셸의 경로에 `aws_completer`가 있어야 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 completer 찾기

AWS Completer의 위치는 사용한 설치 방법에 따라 다를 수 있습니다.

- 패키지 관리자 - `pip`, `yum`, `brew` 및 `apt-get`과 같은 프로그램은 일반적으로 AWS completer(또는 이 기능에 대한 symlink)를 표준 경로 위치에 설치합니다.
  - `pip`을 `--user` 파라미터 없이 사용한 경우 기본 경로는 `/usr/local/bin/aws_completer`입니다.
  - `pip`을 `--user` 파리미터와 함께 사용한 경우 기본 경로는 `/home/username/.local/bin/aws_completer`입니다.
- 번들에 포함된 설치 관리자 - 번들에 포함된 설치 관리자를 사용한 경우 기본 경로는 `/usr/local/bin/aws_completer`입니다.

다른 모든 방법이 실패하면 `find` 명령을 사용하여 파일 시스템에서 AWS completer를 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.

- echo $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 섹션에서 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/* 경로를 시스템에서 찾은 경로에 completer 폴더가 있는지 확인로 바꿉니다.

- **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
  ```

  `~/.zshrc`에 이전 명령을 추가하여 새 셸을 열 때마다 실행되도록 합니다.

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

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

  `~/.tcshrc`에 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**
>
> 다양한 완성 키를 포함하여 PowerShell이 완성을 처리하는 방법에 대한 자세한 내용은 Microsoft PowerShell 문서에서 about_Tab_Expansion을 참조하세요.

Windows에서 PowerShell에 대한 명령 완성을 활성화하려면 PowerShell에서 다음 단계를 수행합니다.

1. 다음 명령을 사용하여 `$PROFILE`을 엽니다.

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

   `$PROFILE`이 없는 경우 다음 명령을 사용하여 사용자 프로필을 생성합니다.

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

   PowerShell 프로필에 대한 자세한 내용은 Microsoft Docs 웹 사이트에서 Windows PowerShell ISE에서 프로필을 사용하는 방법을 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 CLI에서 AWS CLI 재시도

이 주제에서는 AWS CLI에서 예기치 않은 문제로 인해 AWS 서비스 호출이 실패하는 것을 어떻게 확인할 수 있는지 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. AWS CLI는 이러한 종류의 오류 또는 예외가 발생할 때 AWS 서비스에 대한 클라이언트 호출을 다시 시도하는 데 도움이 되는 여러 기능을 제공합니다.

주제

사용 가능한 재시도 모드

AWS CLI에는 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, 509 등 여러 HTTP 상태 코드에 대한 재시도 횟수.
- 모든 재시도 횟수에는 기본 계수 2의 지수 백오프가 포함됩니다.

표준 재시도 모드

표준 모드는 레거시 모드보다 많은 기능을 가진 AWS SDK 전반의 표준 재시도 규칙의 집합입니다. 표준 모드는 AWS CLI 버전 2를 위해 생성되었으며 AWS CLI 버전 1로 백포트됩니다. 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, 이러한 HTTP 상태 코드는 500, 502, 503, 504입니다.
- 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 재시도의 유연성을 제공합니다.

새로운 재시도를 시도할 때마다 적응형 모드는 AWS 서비스의 응답에 표시된 오류, 예외 또는 HTTP 상태 코드를 기반으로 속도 제한 변수를 수정합니다. 이러한 속도 제한 변수는 클라이언트의 새 호출 속도를 계산하는 데 사용됩니다. AWS 서비스의 각 예외/오류 또는 비 성공 HTTP 응답(위 목록에 제공)은 성공에 도달하거나, 토큰 버킷이 소진되거나, 구성된 최대 시도 값에 도달할 때까지 재시도가 발생할 때 속도 제한 변수를 업데이트합니다.

재시도 모드 구성

AWS CLI에는 클라이언트 객체를 생성할 때 고려해야 할 구성 방법뿐만 아니라 다양한 재시도 구성이 포함됩니다.

사용 가능한 구성 방법

AWS CLI에서 사용자는 다음과 같은 방법으로 재시도를 구성할 수 있습니다.

- 환경 변수
- AWS CLI 구성 파일

사용자는 다음 재시도 옵션을 사용자 지정할 수 있습니다.

- 재시도 모드 - AWS CLI가 사용할 재시도 모드를 지정합니다. 앞에서 설명한 대로 레거시, 표준 및 적응형, 이렇게 3가지 재시도 모드를 사용할 수 있습니다. AWS CLI 버전 1의 기본값은 레거시이며.
- 최대 시도 - AWS CLI 재시도 핸들러에서 사용하는 최대 재시도 횟수 값을 지정합니다. 여기서 초기 호출은 사용자가 제공한 값에 포함됩니다. 기본값은 재시도 모드를 기반으로 합니다.

환경 변수에서 재시도 구성 정의

AWS CLI에 대한 재시도 구성을 정의하려면 운영 체제의 환경 변수를 업데이트합니다.

재시도 환경 변수는 다음과 같습니다.

- `AWS_RETRY_MODE`
- `AWS_MAX_ATTEMPTS`

환경 변수에 대한 자세한 내용은 [AWS CLI에 대한 환경 변수 구성] 섹션을 참조하세요.

재시도 로그 보기

AWS CLI는 Boto3의 재시도 방법과 로깅을 사용합니다. 모든 명령에서 `--debug` 옵션을 사용하여 디버그 로그를 받을 수 있습니다. `--debug` 옵션을 사용하는 방법에 대한 자세한 내용은 [AWS CLI의 명령줄 옵션] 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 리포지토리에서 [_retry.json]을 참조하세요.

AWS CLI에 대한 HTTP 프록시 사용

프록시 서버를 통해 AWS에 액세스하려면 프록시 서버에서 사용되는 DNS 도메인 이름 또는 IP 주소 및 포트 번호로 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 예제는 프록시의 명시적 IP 주소나 프록시의 IP 주소로 확인되는 DNS 이름을 사용할 수 있는 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 CLI는 NTLM 프록시를 지원하지 않습니다. NTLM 또는 Kerberos 프로토콜 프록시를 사용하는 경우 Cntlm과 같은 인증 프록시를 통해 연결할 수 있습니다.

AWS CLI는 HTTP 기본 인증을 지원합니다. 다음과 같이 프록시 URL에 사용자 이름 및 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 역할을 사용하여 시작한 Amazon EC2 인스턴스에서 프록시를 구성하는 경우 인스턴스 메타데이터에 액세스하는 데 사용된 주소를 제외해야 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

AWS CLI에서 문제가 발생할 경우 오류 해결에 나온 문제 해결 단계를 참조하세요. 가장 관련성이 높은 문제 해결 단계는 the section called “SSL 인증서 오류” 섹션을 참조하세요.

AWS CLI에서 엔드포인트 사용

AWS 서비스에 프로그래밍 방식으로 연결하려면 엔드포인트를 사용해야 합니다. 엔드포인트는 AWS 웹 서비스를 위한 진입점의 URL입니다. AWS Command Line Interface(AWS CLI)는 AWS 리전 리전의 각 서비스에 대해 자동으로 기본 엔드포인트를 사용하지만, API 요청에 대해 대체 엔드포인트를 지정할 수 있습니다.

엔드포인트 주제

- 단일 명령에 대한 엔드포인트 설정

- 모든 AWS 서비스 서비스에 대한 글로벌 엔드포인트 설정
- 모든 AWS 서비스에 FIPS 엔드포인트를 사용하도록 설정
- 모든 AWS 서비스에 이중 스택 엔드포인트를 사용하도록 설정
- 서비스별 엔드포인트 설정
  - 서비스별 엔드포인트: 환경 변수
  - 서비스별 엔드포인트: 공유 config 파일
  - 서비스별 엔드포인트: 서비스별 식별자 목록
- 계정 기반 엔드포인트
- 엔드포인트 구성 및 설정 우선 순위

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 URL을 사용합니다.

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

모든 AWS 서비스 서비스에 대한 글로벌 엔드포인트 설정

모든 서비스에 대한 요청을 사용자 지정 엔드포인트 URL로 라우팅하려면 다음 설정 중 하나를 사용하세요.

- 환경 변수:
  - `AWS_IGNORE_CONFIGURED_ENDPOINT_URLS` - 구성된 엔드포인트 URL을 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 - 글로벌 엔드포인트 URL을 설정합니다.

  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 엔드포인트 URL을 무시합니다.

```
ignore_configure_endpoint_urls = true
```

  - endpoint_url - 글로벌 엔드포인트 URL을 설정합니다.

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

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

모든 AWS 서비스에 FIPS 엔드포인트를 사용하도록 설정

모든 서비스에 대한 요청을 FIP 엔드포인트로 라우팅하려면 다음 중 하나를 사용하세요.

- `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 AWS 리전에서 FIPS(Federal Information Processing Standard) 140-2를 지원하는 엔드포인트를 제공합니다. AWS 서비스에서 FIPS를 지원하는 경우 이 설정은 AWS CLI에서 사용해야 하는 FIPS 엔드포인트를 지정합니다. 표준 AWS 엔드포인트와 달리 FIPS 엔드포인트에서는 FIPS 140-2를 준수하는 TLS 소프트웨어 라이브러리를 사용합니다. 이러한 엔드포인트는 미국 정부와 상호 작용하는 기업에 필요할 수 있습니다.

이 설정이 활성화되어 있지만 해당 서비스에 대한 FIPS 엔드포인트가 AWS 리전에 없는 경우 AWS 명령이 실패할 수 있습니다. 이 경우 `--endpoint-url` 옵션을 사용하여 명령에 사용할 엔드포인트를 수동으로 지정하거나 서비스별 엔드포인트를 사용합니다.

AWS 리전별로 FIPS 엔드포인트를 지정하는 방법에 대한 자세한 내용은 서비스별 FIPS 엔드포인트를 참조하세요.

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
  ```

이중 스택 엔드포인트를 사용하여 AWS 요청을 보내도록 설정합니다. IPv4 및 IPv6 트래픽을 모두 지원하는 이중 스택 엔드포인트에 대한 자세한 내용은 Amazon 심플 스토리지 서비스 사용 설명서의 Amazon S3 이중 스택 엔드포인트 사용을 참조하세요. 이중 스택 엔드포인트는 일부 리전에 사용할 수 있는 서비스입니다. 서비스 또는 AWS 리전에 대한 이중 스택 엔드포인트가 없는 경우 요청이 실패합니다. 이 옵션은 기본적으로 비활성화되어 있습니다.

서비스별 엔드포인트 설정

서비스별 엔드포인트 구성은 AWS CLI 요청에 대해 사용자가 선택한 영구 엔드포인트를 사용할 수 있는 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 AWS 서비스 서비스에 대한 엔드포인트 URL을 지정할 수 있습니다.

서비스별 엔드포인트는 다음과 같은 방법으로 지정할 수 있습니다.

- 단일 명령에 대한 명령줄 옵션 `--endpoint-url`.
- 환경 변수:
  - `AWS_IGNORE_CONFIGURED_ENDPOINT_URLS` - 명령줄에 지정하지 않는 한 구성된 모든 엔드포인트 URL을 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 명령줄에서 지정하지 않는 한 구성된 모든 엔드포인트 URL을 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 지정하지 않는 한 구성된 모든 엔드포인트 URL을 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 지정하며, 여기서 AWS 서비스가 `<SERVICE>` 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_url은 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 대한 서비스별 엔드포인트 URL과 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 동일한 프로파일에서 Amazon S3와 AWS Elastic Beanstalk에 대한 서비스별 엔드포인트 URL을 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 서비스 식별자는 모든 공백을 밑줄로 바꾸고 모든 문자를 소문자로 바꾸는 방식으로 API 모델의 serviceId를 기반으로 합니다.

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

계정 기반 엔드포인트

계정 기반 엔드포인트는 다음과 같은 방법으로 지정할 수 있습니다.

- 환경 변수
  - AWS_ACCOUNT_ID - 지원되는 AWS 서비스에 대한 호출에 사용할 AWS 계정 기반 엔드포인트 ID를 지정합니다.

    Linux or macOS

    ```
    $ export AWS_ACCOUNT_ID=<account-id>
    ```

    Windows Command Prompt

    모든 세션에 대해 설정하려면

    ```
    C:\> setx AWS_ACCOUNT_ID <account-id>
    ```

    현재 세션에만 설정하려면

```
C:\> set AWS_ACCOUNT_ID=<account-id>
```

PowerShell

```
PS C:\> $Env:AWS_ACCOUNT_ID="<account-id>"
```

- AWS_ACCOUNT_ID_ENDPOINT_MODE - 지원되는 AWS 서비스에 대한 호출에 AWS 계정 기반 엔드포인트 ID를 사용할지 여부를 지정합니다. preferred, disabled 또는 required로 설정할 수 있습니다. 기본값은 preferred입니다.

  Linux or macOS

  ```
  $ export AWS_ACCOUNT_ID_ENDPOINT_MODE=preferred
  ```

  Windows Command Prompt

  모든 세션에 대해 설정하려면

  ```
  C:\> setx AWS_ACCOUNT_ID_ENDPOINT_MODE preferred
  ```

  현재 세션에만 설정하려면

  ```
  C:\> set AWS_ACCOUNT_ID_ENDPOINT_MODE=preferred
  ```

  PowerShell

  ```
  PS C:\> $Env:AWS_ACCOUNT_ID_ENDPOINT_MODE="preferred"
  ```

- config 파일:
  - aws_account_id - 지원되는 AWS 서비스에 대한 호출에 사용할 AWS 계정 기반 엔드포인트 ID를 지정합니다.

    ```
    aws_account_id = <account-id>
    ```

  - account_id_endpoint_mode - 지원되는 AWS 서비스에 대한 호출에 AWS 계정 기반 엔드포인트 ID를 사용할지 여부를 지정합니다. preferred, disabled 또는 required로 설정할 수 있습니다. 기본값이 선호됩니다.

```
account_id_endpoint_mode = preferred
```

계정 기반 엔드포인트는 AWS 계정 ID를 사용하여 이 기능을 지원하는 서비스에 대한 AWS 서비스 요청 라우팅을 간소화하여 고성능과 확장성을 확보하는 데 도움이 됩니다. 자격 증명 공급자와 계정 기반 엔드포인트를 지원하는 서비스를 사용하는 경우 AWS CLI는 리전 엔드포인트 대신 계정 기반 엔드포인트를 자동으로 구성하고 사용합니다.

계정 기반 엔드포인트는 다음 형식을 사용합니다. 여기서 `<account-id>`는 AWS 계정 ID로 대체되고 `<region>`는 AWS 리전으로 대체됩니다.

```
https://<account-id>.myservice.<region>.amazonaws.com
```

기본적으로 AWS CLI에서 계정 기반 엔드포인트 모드는 `preferred`로 설정됩니다.

엔드포인트 구성 및 설정 우선 순위

엔드포인트 구성 설정은 시스템 또는 사용자 환경 변수, 로컬 AWS 구성 파일 또는 명령줄에서 파라미터로 명시적으로 선언된 위치 등 다양한 장소에 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. `AWS_ACCOUNT_ID_ENDPOINT_MODE` 환경 변수는 `AWS_ACCOUNT_ID` 환경 변수 또는 `aws_account_id` 설정에서 계정 ID를 사용하여 `preferred` 또는 `required`로 설정됩니다.
6. 공유 `config` 파일의 `services` 섹션 내의 `endpoint_url` 설정에서 제공하는 서비스별 엔드포인트 값.
7. 공유 `config` 파일의 `profile` 내에서 `endpoint_url` 설정에 의해 제공되는 값.
8. `use_dualstack_endpoint`, `use_fips_endpoint` 및 `endpoint_url` 설정입니다.

9. `account_id_endpoint_mode` 설정은 `AWS_ACCOUNT_ID` 환경 변수 또는 `aws_account_id` 설정에서 계정 ID를 사용하여 `preferred` 또는 `required`로 설정됩니다.
10. 각 AWS 서비스에 대한 기본 엔드포인트 URL이 마지막에 사용됩니다. 각 리전에서 사용할 수 있는 표준 서비스 엔드포인트 목록은 Amazon Web Services 일반 참조의 AWS 리전 및 엔드포인트를 참조하세요.

AWS CLI에 대한 인증 및 액세스 보안 인증

AWS 서비스를 사용하여 개발할 때 AWS를 사용한 AWS CLI 인증 방법을 설정해야 합니다. AWS CLI에 대한 프로그램 방식의 액세스 보안 인증을 구성하려면 다음 옵션 중 한 가지를 선택합니다. 옵션은 권장 순서입니다.

인증 유형	용도	지침
IAM 사용자 단기 자격 증명	장기 자격 증명보다 더 안전한 IAM 사용자 단기 자격 증명을 사용하세요. 자격 증명이 유출된 경우 만료되기 전까지 사용할 수 있는 시간이 제한되어 있습니다.	the section called “단기 보안 인증”
Amazon EC2 인스턴스의 IAM 또는 IAM Identity Center 사용자.	Amazon EC2 인스턴스 메타데이터를 사용하여 Amazon EC2 인스턴스에 할당된 역할을 사용하여 임시 자격 증명을 쿼리할 수 있습니다.	the section called “Amazon EC2 메타데이터”
권한에 대한 역할 가정	다른 자격 증명 방법을 페어링하고 사용자가 액세스 권한이 없을 수 있는 AWS 서비스에 대한 임시 액세스를 위한 역할을 맡습니다.	the section called “IAM 역할”
IAM 사용자 장기 자격 증명	(권장하지 않음) 만료 기간이 없는 장기 인증 정보를 사용하세요.	the section called “IAM 사용자”
IAM 의 외부 스토리지	(권장되지 않음) 다른 자격 증명 방법을 페어링하되 자격 증명 값을 AWS CLI 외부의 위치에 저장합니다. 이 방법은 자격 증명이 저장된 외부 위치만큼만 안전합니다.	the section called “외부 자격 증명”

구성 및 보안 인증 우선 순위

보안 인증 및 구성 설정은 시스템 또는 사용자 환경 변수, 로컬 AWS 구성 파일 또는 명령줄에서 파라미터로 명시적으로 선언된 위치 등 다양한 장소에 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 위임 - 구성 또는 `assume-role` 명령을 통해 IAM 역할의 권한을 위임합니다.
4. 웹 ID로 역할 위임 - 구성 또는 `assume-role-with-web-identity` 명령을 통해 웹 ID를 사용하여 IAM 역할의 권한을 위임합니다.
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(Amazon ECS) 태스크 정의에 연결할 수 있습니다. 그러면 작업의 컨테이너에 대해 해당 역할의 임시 보안 인증을 사용할 수 있습니다. 자세한 내용은 Amazon Elastic Container Service 개발자 안내서에서 태스크에 대한 IAM 역할을 참조하세요.
9. Amazon EC2 인스턴스 프로파일 보안 인증 - IAM 역할을 각 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스에 연결할 수 있습니다. 그러면 인스턴스에서 실행되는 코드에 대해 해당 역할의 임시 보안 인증을 사용할 수 있습니다. 보안 인증은 Amazon EC2 메타데이터 서비스를 통해 전달됩니다. 자세한 내용은 Amazon EC2 사용 설명서의 Amazon EC2에 대한 IAM 역할과 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 “Amazon EC2 메타데이터”

- the section called “외부 자격 증명”

AWS CLI에 대한 단기 자격 증명으로 인증

세션 기간 연장 옵션과 함께 IAM Identity Center 인증을 사용하도록 SDK 또는 도구를 구성하는 것이 좋습니다. 하지만 AWS 액세스 포털에서 사용할 수 있는 임시 보안 인증을 복사하여 사용할 수 있습니다. 보안 인증이 만료되면 새 보안 인증을 복사해야 합니다. 프로필에서 임시 보안 인증을 사용하거나 이를 시스템 속성 및 환경 변수의 값으로 사용할 수 있습니다.

1. AWS 액세스 포털에 로그인합니다.
2. 다음 지침에 따라 AWS 액세스 포털에서 IAM 역할 보안 인증을 복사하세요.
    1. 링크된 지침의 2단계에서 개발 요구 사항에 대한 액세스 권한을 부여하는 AWS 계정 및 IAM 역할 이름을 선택합니다. 이 역할에는 일반적으로 PowerUserAccess 또는 Developer와 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 “구성 설정” 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
```

SDK는 서비스 클라이언트를 생성할 때 이러한 임시 보안 인증에 액세스하여 각 요청에 사용합니다. 2a단계에서 선택한 IAM 역할 설정에 따라 임시 보안 인증의 유효 기간이 결정됩니다. 최대 유효 기간은 12시간입니다.

보안 인증 정보가 만료될 때마다 이 단계를 반복하세요.

AWS CLI에서 IAM 역할 사용

AWS Identity and Access Management(IAM) 역할은 사용자가 추가(또는 다른) 권한을 얻을 수 있도록 하거나 다른 AWS 계정에서 작업을 수행할 권한을 주는 인증 도구입니다.

주제

사전 조건

`iam` 명령을 실행하려면 AWS CLI를 설치하고 구성해야 합니다. 여기에는 역할이 다른 자격 증명 방법과 페어링되어 있다고 가정하여 구성된 프로필을 설정하는 것도 포함됩니다. 자세한 내용은 AWS CLI 설치 섹션을 참조하세요.

IAM 역할 사용 개요

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

다음 예제는 `marketingadmin`라는 이름의 역할 프로파일을 보여줍니다. `--profile marketingadmin`으로 명령을 실행하거나(AWS_PROFILE 환경 변수로 이를 지정한 경우) AWS CLI가 별도의 `user1` 프로파일에 정의된 자격 증명을 사용하여 Amazon 리소스 이름(ARN)이

arn:aws:iam::*123456789012*:role/*marketingadminrole*인 역할을 수임합니다. 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 자격 증명을 사용하고 있습니다. AWS CLI 명령에 `marketingadmin` 프로파일을 사용하도록 지정하면 AWS CLI가 연결된 `user1` 프로파일에 대한 자격 증명을 자동으로 찾고 이를 사용하여 지정된 IAM 역할에 대한 임시 자격 증명을 요청합니다. CLI는 백그라운드에서 sts:AssumeRole 작업을 사용하여 이를 수행합니다. 이러한 임시 자격 증명은 요청된 AWS CLI 명령을 실행하는 데 사용됩니다. 지정된 역할은 요청된 AWS CLI 명령이 실행되도록 허용하는 IAM 권한 정책에 연결되어야 합니다.

Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스 또는 Amazon Elastic Container Service(Amazon ECS) 컨테이너 안에서 AWS CLI 명령을 실행하려면 인스턴스 프로파일 또는 컨테이너에 연결된 IAM 역할을 사용하면 됩니다. 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.

아래 예제는 Amazon EC2 인스턴스 프로파일을 참조하여 사용한 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 보다 쉽게 감사할 수 있는 고유한 역할 세션 이름을 지정할 수도 있습니다.

역할 구성 및 사용

IAM 역할을 지정하는 프로파일을 사용하여 명령을 실행하면 AWS CLI는 원본 프로파일의 자격 증명을 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 [IAM 사용자에게 권한을 위임할 역할 생성]에 있는 절차에 따라 사용자가 수임할 수 있도록 하는 권한이 있는 역할을 IAM에 생성할 수 있습니다. 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.

JSON

```
{
  "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 방법은 IAM 사용 설명서에서 사용자에게 역할 전환 권한 부여를 참조하세요.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Resource": "arn:aws:iam::123456789012:role/marketingadminrole"
    }
  ]
}
```

사용자는 추가 권한이 없더라도 역할 프로파일을 사용하여 AWS CLI 명령을 실행할 수 있습니다. 대신 명령을 실행하는 데 필요한 권한은 역할에 연결된 권한으로부터 나옵니다. 권한 정책을 역할에 연결하여 어떤 AWS 리소스에 대해 어떤 작업을 수행할 수 있는지를 지정할 수 있습니다. 권한을 역할에 연결(사용자에게 동일하게 적용)하는 방법은 IAM 사용 설명서에서 IAM 사용자의 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 ID(사용자, 사용자 그룹 및 역할) 및 IAM 역할을 참조하세요.

멀티 팩터 인증 사용

보안을 강화하기 위해, 사용자가 역할 프로파일을 사용하여 호출을 수행하려고 할 때 멀티 팩터 인증 (MFA) 디바이스, U2F 디바이스 또는 모바일 앱에서 생성된 일회용 키를 제공하도록 사용자에게 요구할 수 있습니다.

먼저, MFA를 요구하도록 IAM 역할에 대한 신뢰 관계를 수정합니다. 이렇게 하면 MFA를 사용하여 먼저 인증해야 역할을 사용할 수 있게 됩니다. 해당하는 예는 다음 예제의 `Condition` 행을 참조하세요. 이 정책을 사용하면 이름이 `anika`인 사용자가 정책이 연결된 역할을 수임할 수 있지만, MFA를 사용하여 인증하는 경우에만 수임할 수 있습니다.

JSON

```
{
  "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 ARN을 지정하는 줄을 역할 프로파일에 추가합니다. 다음 샘플 `config` 파일 항목은 `anika`라는 사용자가 `cli-role` 역할에 대한 임시 보안 인증 정보를 요청하기 위해 액세스 키를 사용하는 두 가지 역할 프로파일을 보여줍니다. 사용자 `anika`는 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` 설정은 표시된 대로 ARN 또는 하드웨어 MFA 토큰의 일련 번호를 가져올 수 있습니다.

첫 번째 프로파일인 `role-without-mfa`에서는 MFA를 요구하지 않습니다. 그러나 이전 예제에서 역할에 연결된 신뢰 정책은 MFA를 요구하므로 이 프로파일을 사용하여 명령을 실행하면 실패하게 됩니다.

```
$ aws iam list-users --profile role-without-mfa

An error occurred (AccessDenied) when calling the AssumeRole operation: Access denied
```

두 번째 프로파일 항목 `role-with-mfa`에서는 사용할 MFA 디바이스를 식별합니다. 사용자가 이 프로파일을 사용하여 AWS CLI 명령을 실행하면 사용자에게 MFA 디바이스에서 제공하는 OTP(일회용 암호)를 입력하라는 메시지가 AWS CLI에 표시됩니다. MFA 인증이 성공하면 명령이 요청 작업을 수행합니다. OTP는 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. IAM 사용자에게 권한을 위임하는 역할 생성에 설명된 대로 역할 생성 중에 역할 유형을 다른 AWS 계정으로 설정합니다. 경우에 따라 Require MFA(MFA 필요)를 선택합니다. [MFA 필요(Require MFA)]는 멀티 팩터 인증 사용의 설명과 같이 신뢰 관계에서 적절한 조건을 구성합니다.

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` 값이 `AssumeRole` 작업에 전달되고 역할 세션에 대한 ARN의 일부가 됩니다. 이 값은 로그된 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
```

이 결과 다음 ARN을 포함하는 역할 세션이 생성됩니다.

```
arn:aws:iam::234567890123:assumed-role/SomeRole/Session_Maria_Garcia
```

또한 모든 AWS CloudTrail 로그에 각 작업에 대해 캡처된 정보의 역할 세션 이름이 포함됩니다.

웹 자격 증명을 사용한 역할 수임

AWS CLI에서 [웹 자격 증명 연동 및 OIDC(Open ID Connect)](#)를 사용하여 역할을 수임하도록 프로파일을 구성할 수 있습니다. 프로파일에서 이를 지정할 때 AWS CLI에서 자동으로 해당 AWS STS `AssumeRoleWithWebIdentity` 호출을 수행합니다.

> **Note**
>
> IAM 역할을 사용하는 프로파일을 지정하면 AWS CLI가 적절한 호출을 수행하여 임시 자격 증명을 검색합니다. 이러한 자격 증명은 `~/.aws/cli/cache`에 저장됩니다. 동일한 프로파일을

지정하는 후속 AWS CLI 명령은 캐시된 임시 자격 증명을 만료될 때까지 사용합니다. 이 시점에서 AWS CLI는 자동으로 자격 증명을 새로 고칩니다.

웹 자격 증명 연동을 사용하여 임시 자격 증명을 검색하고 사용하려면 공유 프로파일에서 다음 구성 값을 지정할 수 있습니다.

role_arn

수임할 역할의 ARN을 지정합니다.

web_identity_token_file

자격 증명 공급자가 제공하는 OAuth 2.0 액세스 토큰 또는 OpenID Connect ID 토큰을 포함하는 파일의 경로를 지정합니다. AWS CLI에서 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 역할의 ARN

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 AWS CLI는 임시 자격 증명이 만료될 때까지 로컬에서 임시 자격 증명을 캐시합니다. 다음에 이러한 자격 증명을 사용하려고 할 때 AWS CLI에서 해당 자격 증명을 자동으로 갱신하려고 합니다.

역할의 임시 자격 증명이 취소된 경우에는 해당 자격 증명이 자동으로 갱신되지 않고 사용 시도가 실패합니다. 그러나 강제로 AWS CLI에서 새 자격 증명을 검색하도록 하는 캐시를 삭제할 수 있습니다.

Linux 또는 macOS

```
$ rm -r ~/.aws/cli/cache
```

Windows

```
C:\> del /s /q %UserProfile%\.aws\cli\cache
```

IAM 사용자 자격 증명을 사용한 인증

> **Warning**
> 보안 위험을 방지하려면 목적별 소프트웨어를 개발하거나 실제 데이터로 작업할 때 IAM 사용자를 인증에 사용하지 마세요. 대신 AWS IAM Identity Center과 같은 보안 인증 공급자를 통한 페더레이션을 사용하세요.

이 섹션에서는 IAM 사용자를 사용하여 기본 설정을 구성하는 방법을 설명합니다. 여기에는 `config` 및 `credentials` 파일을 사용한 보안 보안 인증이 포함됩니다.

주제

- AWS CLI 구성
  - aws configure 사용하기

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. AWS Management Console에 로그인하여 https://console.aws.amazon.com/iam/ 에서 IAM 콘솔을 엽니다.
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 키 및 보안 액세스 키와 함께 `.csv` 파일을 저장합니다. 나중에 이 정보가 필요합니다.
7. 완료(Done)를 선택합니다.

AWS CLI 구성

일반적인 용도로는 AWS CLI에서 다음과 같은 정보가 필요합니다.

- 액세스 키 ID
- 비밀 액세스 키
- AWS 리전
- 출력 형식

AWS CLI는 이 정보를 default 파일에서 credentials라는 프로파일(설정 모음)에 저장합니다. 기본적으로 이 프로파일의 정보는 사용할 프로파일을 명시적으로 지정하지 않는 AWS CLI 명령이 실행될 때 사용됩니다. credentials 파일에 대한 자세한 내용은 AWS CLI의 구성 및 보안 인증 파일 설정 섹션을 참조하세요.

AWS CLI을 구성하려면 다음 절차 중 하나를 사용합니다.

주제

- aws configure 사용하기

aws configure 사용하기

일반적인 용도에서 aws configure 명령은 AWS CLI 설치를 설정할 수 있는 가장 빠른 방법입니다. 이 구성 마법사는 시작하는 데 필요한 각 정보를 입력하라는 메시지를 표시합니다. --profile 옵션을 사용하여 별도로 지정하지 않는 한 AWS CLI은 이 정보를 default 프로파일에 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
```

AWS CLI에서 Amazon EC2 인스턴스 메타데이터의 자격 증명 사용

Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스 안에서 AWS CLI를 실행하는 경우 명령에 제공하는 자격 증명을 간편하게 제공할 수 있습니다. 각 Amazon EC2 인스턴스에는 AWS CLI가 임시

자격 증명을 직접 쿼리할 수 있는 메타데이터가 포함되어 있습니다. IAM 역할이 인스턴스에 연결되면 AWS CLI가 임시 자격 증명을 직접 쿼리할 수 있는 메타데이터가 포함되어 있습니다.

이 서비스를 비활성화하려면 AWS_EC2_METADATA_DISABLED 환경 변수를 사용합니다.

주제

사전 조건

AWS CLI에서 Amazon EC2 자격 증명을 사용하려면 다음을 완료해야 합니다.

- AWS CLI를 설치하고 구성합니다. 자세한 내용은 AWS CLI 설치 및 AWS CLI에 대한 인증 및 액세스 보안 인증 섹션을 참조하세요.
- 구성 파일과 명명된 프로파일을 파악합니다. 자세한 내용은 AWS CLI의 구성 및 보안 인증 파일 설정 단원을 참조하십시오.
- 필요한 리소스에 대한 액세스 권한이 있는 AWS Identity and Access Management(IAM) 역할을 생성하고 해당 역할을 시작할 때 Amazon EC2 인스턴스에 연결했습니다. 자세한 내용은 Amazon EC2 사용 설명서의 Amazon EC2에 대한 IAM 정책과 IAM 사용 설명서의 Amazon EC2 인스턴스에서 실행되는 애플리케이션에 AWS 리소스에 대한 액세스 권한 부여를 참조하세요.

Amazon EC2 메타데이터에 대한 프로파일 구성

호스팅 Amazon EC2 인스턴스 프로파일에서 제공되는 자격 증명을 사용할지 여부를 지정하려면 구성 파일의 명명된 프로파일에서 다음 구문을 사용합니다. 자세한 지침은 다음 단계를 참조하세요.

```
[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 대한 액세스 권한이 있는 IAM arn 역할을 추가합니다.

```
role_arn = arn:aws:iam::123456789012:role/rolename
```

3. 자격 증명 소스로 Ec2InstanceMetadata를 지정합니다.

```
credential_source = Ec2InstanceMetadata
```

4. 리전을 설정합니다.

```
region = region
```

예

다음 예제에서는 *marketingadminrole* 역할을 가정하고 *marketingadmin*이라는 Amazon EC2 인스턴스 프로파일에서 *us-west-2* 리전을 사용합니다.

```
[profile marketingadmin]
role_arn = arn:aws:iam::123456789012:role/marketingadminrole
credential_source = Ec2InstanceMetadata
region = us-west-2
```

AWS CLI에서 외부 프로세스를 통해 자격 증명 소싱

> **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 증명의 손상 위험을 줄이려면 AWS CLI 및 AWS에서 제공되고 지원되는 보안 자격 증명을 대신 사용하는 것이 좋습니다. `config` 파일과 유출 방지를 도와주는 모든 파일 및 도구를 보호하고 있는지 확인합니다.
> 사용자 지정 자격 증명 도구가 `StdErr`에 어떤 암호 정보도 기록하지 않아야 합니다. SDK 및 AWS CLI가 그러한 정보를 캡처 및 기록할 수 있으므로 승인되지 않은 사용자에게 노출될 위험이 있기 때문입니다.

AWS CLI에서 직접 지원되지 않는 자격 증명을 생성 또는 조회할 수 있는 방법이 있는 경우에는 `config` 파일에서 AWS CLI 설정을 구성하여 이를 사용하도록 `credential_process`를 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 따르세요.

- 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 묶으세요.
- 문자열 안에 환경 변수를 포함하지 마세요. 예를 들어 $HOME 또는 %USERPROFILE%를 포함할 수 없습니다.
- 홈 폴더를 ~로 지정하지 마세요. 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

AWS CLI는 프로파일에 지정된 대로 명령을 실행한 다음, STDOUT에서 데이터를 읽습니다. 지정한 명령이 다음 구문과 일치하는 STDOUT에서 JSON 출력을 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 존재하지 않으면 CLI는 자격 증명이 새로 고침이 되지 않는 장기 자격 증명이라고 가정합니다. 그렇지 않은 경우 자격 증명은 임시 자격 증명으로 간주되며, 기간이 만료되기 전에 `credential_process` 명령을 다시 실행하면 자동으로 새로 고침됩니다.

Note

AWS CLI는 역할 자격 증명을 수임하는 방법으로 외부 프로세스 자격 증명을 캐싱하지 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.

AWS CLI 사용

이 섹션에서는 구성 the section called "엔드포인트" 섹션에서 다룬 세부 사항을 넘어 AWS Command Line Interface(AWS CLI)에서 사용할 수 있는 일반적인 용도, 일반적인 기능 및 옵션에 대한 포괄적인 개요를 제공합니다.

이 안내서에서는 AWS CLI 명령어의 기본 구조, 형식 지정, 필터링 기능 등 명령어 작성의 기본적인 측면에 대해 자세히 설명합니다. 이러한 핵심 요소를 이해하면 복잡한 웹 기반 콘솔을 탐색할 필요 없이 필요한 리소스와 작업을 정확하게 타겟팅하는 명령을 구성할 수 있습니다.

또한 AWS CLI에 사용할 수 있는 도움말 콘텐츠 및 문서가 강조 표시됩니다. 기본 제공되는 명령줄 도움말부터 종합적인 AWS CLI 참조 안내서까지, AWS CLI의 기능을 탐색하는 데 도움이 되는 정보에 액세스할 수 있습니다.

AWS 서비스별 예제 및 사용 사례는 코드 예제 또는 AWS CLI 참조 안내서를 참조하세요. 여기에서는 명령어별 정보를 제공하고 다양한 AWS 서비스에서 AWS CLI를 활용하는 방법에 대한 예제를 보여 줍니다.

> **Note**
>
> 기본적으로 AWS CLI는 TCP 포트 443에서 HTTPS를 사용하여 AWS 서비스에 요청을 보냅니다. AWS CLI를 성공적으로 사용하려면 이 포트에서 아웃바운드 연결을 할 수 있어야 합니다.

이 안내서의 주제

AWS CLI에 대한 도움말 및 리소스 액세스

이 주제에서는 AWS Command Line Interface(AWS CLI)에 대한 도움말 콘텐츠에 액세스하는 방법을 설명합니다.

주제

- [기본 제공 AWS CLI help 명령](#)
- [AWS CLI 참조 안내서](#)
- [API 설명서](#)
- [오류 해결](#)
- [추가 도움말](#)

기본 제공 AWS CLI help 명령

AWS Command Line Interface(AWS CLI) 사용 중 모든 명령에 대한 도움을 얻을 수 있습니다. 이를 위해 명령 이름 끝에 `help`를 입력하기만 하면 됩니다.

예를 들어, 다음 명령은 일반 AWS CLI 옵션에 대한 도움말과 사용 가능한 최상위 명령을 표시합니다.

```
$ aws help
```

다음 명령은 사용 가능한 Amazon Elastic Compute Cloud(Amazon EC2) 특정 명령을 표시합니다.

```
$ aws ec2 help
```

다음 예제는 Amazon EC2 `DescribeInstances` 작업에 대한 자세한 도움말을 표시합니다. 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 대해 사용 가능한 예제가 없는 경우 이 페이지의 피드백 링크를 사용하여 요청하거나 명령의 도움말 페이지에 있는 AWS CLI 명령 참조에서 요청할 수 있습니다.

```
     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 설명입니다AWS

`describe-instances`의 경우 출력은 예약 객체의 목록이며, 각 객체에는 연관된 인스턴스에 대한 정보를 포함하는 여러 필드 및 객체가 포함됩니다. 이 정보는 Amazon EC2에 사용된 예약 데이터 형식에 대한 API 설명서에서 가져오는 것입니다.

```
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.
...
```

AWS CLI가 출력을 JSON으로 렌더링하는 경우 다음 샘플과 유사한 예약 객체의 배열이 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. 문서를 더 보려면 스페이스바 또는 PgDn 키를 누르고, 중지하려면 **q**를 누릅니다.

```
C:\> aws ec2 describe-instances help | more
```

AWS CLI 참조 안내서

도움말 파일에는 명령줄에서 탐색하거나 볼 수 없는 링크가 포함됩니다. 온라인 AWS CLI 버전 1 참조 가이드를 사용하여 이러한 링크를 보고 상호 작용할 수 있습니다. 참조 가이드에는 모든 AWS CLI 명령에 대한 도움말 콘텐츠도 포함되어 있습니다. 이 설명은 쉽게 탐색하고 모바일, 태블릿 또는 데스크톱 화면에서 볼 수 있도록 제공됩니다.

API 설명서

AWS CLI의 모든 명령은 AWS 서비스의 퍼블릭 API에 대해 이루어진 요청에 해당합니다. 퍼블릭 API를 사용하는 각 서비스에는 AWS 설명서 웹사이트의 해당 서비스 홈페이지에서 찾을 수 있는 API 참조가 있습니다. API 참조 콘텐츠는 API 구성 방법 및 사용되는 프로토콜에 따라 다릅니다. 일반적으로 API 참조에는 API에서 지원하는 작업에 대한 세부 정보, 서비스와 주고 받은 데이터 및 서비스에서 보고할 수 있는 오류 조건이 포함됩니다.

API 설명서 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 AWS CLI의 하위 명령에 해당합니다.
- 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 CLI**
>
> 모든 서비스에서 작동하는 단일 AWS CLI가 작성되기 전에 일부 서비스에 별도의 CLI가 있습니다. 이 서비스별 CLI에는 해당 서비스의 설명서 페이지에서 링크되는 별도의 설명서가 있습니다. 서비스별 CLI에 대한 문서는 AWS CLI에 적용되지 않습니다.

오류 해결

AWS CLI 오류의 진단 및 수정에 대한 자세한 내용은 오류 해결 섹션을 참조하세요.

추가 도움말

AWS CLI 문제에 추가적인 도움이 필요하면 GitHub의 AWS CLI커뮤니티를 방문하세요.

AWS CLI의 명령 구조

이 주제에서는 AWS Command Line Interface(AWS CLI) 명령을 구성하는 방법과 wait 명령을 사용하는 방법에 대해 설명합니다.

주제

- 명령 구조
- wait 명령

명령 구조

AWS CLI는 다음 순서로 지정되어야 하는 명령줄에서 멀티파트 구조를 사용합니다.

1. aws 프로그램에 대한 기본 호출.
2. 최상위 명령: 일반적으로 AWS에서 지원하는 AWS CLI 서비스에 해당합니다.
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, JSON 구조와 같은 다양한 유형의 입력 값을 가져올 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` 섹션을 참조하세요.

AWS CloudFormation

다음 `create-change-set`명령 예제에서는 cloudformation 스택 이름을 *my-change-set*로 변경합니다.

```
$ aws cloudformation create-change-set --stack-name my-stack --change-set-name my-change-set
```

AWS CloudFormation 명령에 대한 자세한 내용은 AWS CLI 명령 참조의 `aws cloudformation` 섹션을 참조하세요.

wait 명령

일부 AWS 서비스에는 사용 가능한 `wait` 명령이 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.

AWS CLI는 다음 순서로 지정되어야 하는 `wait` 명령의 경우 명령줄에서 멀티파트 구조를 사용합니다.

1. `aws` 프로그램에 대한 기본 호출.
2. 최상위 명령: 일반적으로 AWS에서 지원하는 AWS CLI 서비스에 해당합니다.
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, JSON 구조와 같은 다양한 유형의 입력 값을 가져올 수 있습니다. 무엇이 지원되는지는 지정하는 명령 및 하위 명령에 따라 달라집니다.

> **Note**
>
> 모든 AWS 서비스가 `wait` 명령을 지원하는 것은 아닙니다. 서비스가 `wait` 명령을 지원하는지 확인하려면 AWS CLI 참조 가이드를 참조하세요.

예시

AWS CloudFormation

다음 wait change-set-create-complete 명령 예제는 *my-stack* 스택의 *my-change-set* 변경 세트가 실행될 준비가 되었음을 확인한 후에만 일시 중지하고 다시 시작합니다.

```
$ 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 완료될 때까지 일시 중지합니다.

```
$ aws deploy wait deployment-successful --deployment-id d-A1B2C3111
```

AWS CodeDeploy `wait` 명령에 대한 자세한 내용은 AWS CLI 명령 참조의 wait 섹션을 참조하세요.

AWS CLI에서 파라미터 값 지정

AWS Command Line Interface(AWS CLI)에 사용되는 대부분의 파라미터는 간단한 문자열 또는 숫자 값입니다(예: 다음 `aws ec2 create-key-pair` 예제의 `my-key-pair` 키 페어 이름).

```
$ 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*.txt`와 `myfile*.txt`가 서로 다른 파라미터로 간주되기 때문입니다.

그러나 PowerShell에서는 `MyFile*.txt`와 `myfile*.txt`가 동일한 파라미터로 간주되므로 두 요청이 동일하게 처리됩니다. 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 대소문자 구분에 대한 자세한 내용은 PowerShell 설명서에서 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 AWS CLI에서 문자열에 따옴표와 리터럴 사용 섹션을 참조하세요.

이러한 토픽에서는 가장 일반적인 터미널 형식 지정 규칙을 다룹니다. 터미널에서 파라미터 값을 인식하는 데 문제가 있는 경우 이 섹션의 토픽을 검토하고 터미널의 문서에서 특정 구문 규칙을 확인해야 합니다.

매개 변수 주제

- AWS CLI에서 공통 파라미터 유형
- AWS CLI에서 문자열에 따옴표와 리터럴 사용
- AWS CLI에서 파일의 파라미터 로드
- AWS CLI에서 AWS CLI 스켈레톤 및 입력 파일
- AWS CLI에서 간편 구문 사용

AWS CLI에서 공통 파라미터 유형

이 섹션에서는 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” 섹션을 참조하세요.

파라미터 유형에는 다음이 포함됩니다.

- String
- Timestamp
- 나열
- 불
- Integer
- 이진/blob(이진 대용량 객체) 및 스트리밍 blob
- 맵
- 문서

String

문자열 파라미터에는 영숫자 문자, 기호 및 ASCII 문자 세트의 공백이 포함될 수 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 섹션을 참조하세요.

Timestamp

타임스탬프는 ISO 8601 표준에 따른 형식입니다. 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(### ##)*, 예: 2014-10-01T12:30:00.000-08:00
- *YYYY*-*MM*-*DD*, 예: 2014-10-01
- 초 단위의 Unix 시간, 예: 1412195400 경우에 따라 Unix Epoch 시간이라고도 하며 1970년 1월 1일 자정 UTC 이후 경과된 초 수를 나타냅니다.

`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

AWS CLI에서는 명령줄에서 직접 문자열로 이진 값을 전달할 수 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.json을 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 CLI의 간편 구문으로 지정된 키-값 페어 집합입니다. 다음 JSON 예제에서는 --key 맵 파라미터가 포함된 my-table이라는 Amazon DynamoDB 테이블에서 항목을 읽습니다. 파라미터는 중첩된 JSON 구조로 1 숫자 값을 가진 id라는 기본 키를 지정합니다.

명령줄에서 고급 JSON을 사용하려면 명령줄 JSON 프로세서(예: jq)를 사용하여 JSON 문자열을 생성하는 것이 좋습니다. jq에 대한 자세한 내용은 GitHub에서 jq 리포지토리를 참조하세요.

```
$ 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 내에 JSON을 포함할 필요 없이 데이터를 전송하는 데 사용됩니다. 문서 유형을 사용하면 서비스에서 보다 유연한 데이터 유형을 사용할 수 있도록 임의의 스키마를 제공할 수 있습니다.

이를 통해 값을 이스케이프 처리하지 않고도 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"}'
```

AWS CLI에서 문자열에 따옴표와 리터럴 사용

AWS CLI에는 주로 두 가지 방법으로 작은따옴표와 큰따옴표가 사용됩니다.

- 공백이 포함된 문자열 주위에 따옴표 사용
- 문자열 안에 따옴표 사용

공백이 포함된 문자열 주위에 따옴표 사용

파라미터 이름과 그 값은 명령줄에서 공백으로 구분됩니다. 문자열 값에 공백이 있는 경우, AWS CLI에서 그 공백을 값과 다음 파라미터 이름 사이의 구분자로 잘못 해석하지 않도록 전체 문자열을 따옴표로 묶어야 합니다. 사용하는 따옴표 유형은 AWS CLI를 실행 중인 운영 체제에 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 참조하세요.

PowerShell

작은따옴표(권장)

작은따옴표 ' '는 `verbatim` 문자열이라고 합니다. 문자열은 입력한 그대로 명령에 전달됩니다. 즉, PowerShell 변수는 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"
```

따옴표 사용에 대한 자세한 내용은 Microsoft PowerShell 문서에서 인용 규칙 정보를 참조하세요.

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. JSON 구조의 각 요소 이름과 값 주위에 공백과 큰따옴표(" ")가 포함되어 있기 때문에 복잡합니다. JSON 형식의 파라미터를 명령줄에 입력하는 방법은 운영 체제에 따라 다릅니다.

명령줄에서 고급 JSON을 사용하려면 명령줄 JSON 프로세서(예: jq)를 사용하여 JSON 문자열을 생성하는 것이 좋습니다. jq에 대한 자세한 내용은 GitHub에서 jq 리포지토리를 참조하세요.

Linux and macOS

Linux 및 macOS가 문자열을 문자 그대로 해석하는 경우 다음 예제와 같이 작은따옴표(' ')를 사용하여 JSON 데이터 구조를 묶습니다. 문자 그대로 처리되므로 JSON 문자열에 포함 된 큰따옴표를 이스케이프 처리할 필요가 없습니다. JSON은 작은따옴표로 묶여 있으므로 문자열의 모든 작은따옴표를 이스케이프 처리해야 합니다. 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 참조하세요.

PowerShell

작은따옴표(' ') 또는 큰따옴표(" ")를 사용합니다.

작은따옴표(권장)

작은따옴표 `' '`는 `verbatim` 문자열이라고 합니다. 문자열은 입력한 그대로 명령에 전달됩니다. 즉, PowerShell 변수는 전달되지 않습니다.

JSON 데이터 구조에는 큰따옴표가 포함되어 있으므로 작은따옴표(`' '`)로 묶는 것이 좋습니다. 작은따옴표를 사용하는 경우 JSON 문자열에 포함된 큰따옴표를 이스케이프 처리할 필요가 없습니다. 그러나 JSON 구조 내에서 각 작은따옴표를 백틱(`)으로 이스케이프 처리해야 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 이스케이프 처리할 필요가 없습니다. 그러나 다음 예제와 같이 JSON 구조 내에서 각 큰따옴표를 백틱(`)으로 이스케이프 처리해야 합니다.

```
PS C:\> 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 Microsoft PowerShell 문서에서 [인용 규칙 정보](#)를 참조하세요.

> **Warning**
>
> PowerShell은 AWS CLI에 명령을 보내기 전에 일반적인 PowerShell 또는 `CommandLineToArgvW` 따옴표 넣기 규칙을 사용하여 명령을 해석할지 결정합니다. PowerShell이 `CommandLineToArgvW`를 사용하여 처리하는 경우 백슬래시(\)로 문자를 이스케이프 처리해야 합니다.
>
> PowerShell의 `CommandLineToArgvW`에 대한 자세한 내용은 Microsoft DevBlogs의 What's up with the strange treatment of quotation marks and backslashes by CommandLineToArgvW, Microsoft Docs Blog의 Everyone quotes command line arguments the wrong way 및 Microsoft Docs의 CommandLineToArgvW 함수를 참조하세요.
>
> 작은따옴표

작은따옴표 ' '는 verbatim 문자열이라고 합니다. 문자열은 입력한 그대로 명령에 전달됩니다. 즉, PowerShell 변수는 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. 큰 따옴표로 묶인 문자열의 경우 백틱만 사용하는 대신 각 따옴표에 대해 '\를 사용하여 두 번 이스케이프 처리해야 합니다. 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 따옴표 넣기 규칙을 무시하려면 Blob을 사용하여 JSON 데이터를 AWS CLI에 직접 전달합니다. Blob에 대한 자세한 내용은 Blob 섹션을 참조하세요.

Windows command prompt

Windows 명령 프롬프트에서는 JSON 데이터 구조를 묶을 큰따옴표(" ")가 필요합니다. 또한 명령 프로세서가 JSON에 포함된 큰따옴표를 잘못 해석하지 않도록 다음 예제와 같이 JSON 데이터 구조 자체에서도 각각의 큰따옴표(\)를 이스케이프 처리해야 합니다(앞에 백슬래시["] 문자 추가).

```
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.

AWS CLI에서 파일의 파라미터 로드

어떤 파라미터는 파일 이름을 인수로 예상하며 AWS CLI가 데이터를 로드합니다. 다른 어떤 파라미터는 파라미터 값을 명령줄에 입력된 텍스트 또는 파일에서 읽은 텍스트로 지정할 수 있습니다. 파일이 필수인지 또는 선택 사항인지에 관계없이 AWS CLI에서 파일을 이해할 수 있도록 파일을 올바르게 인코딩해야 합니다. 파일의 인코딩은 읽는 시스템의 기본 로캘과 일치해야 합니다. Python `locale.getpreferredencoding()` 메서드를 사용하여 이를 확인할 수 있습니다.

이 방법은 단일 파라미터에 대한 파일을 로드하기 위한 것입니다. 단일 파일로 여러 파라미터를 로드하는 방법에 대한 자세한 내용은 the section called “CLI 스켈레톤 템플릿 생성” 섹션을 참조하세요.

> **Note**
>
> 기본적으로 Windows PowerShell은 텍스트를 UTF-16으로 출력하며, 이는 JSON 파일과 여러 Linux 시스템에서 사용하는 UTF-8 인코딩과 충돌합니다. `-Encoding ascii`에서 결과 파일을 읽을 수 있도록 PowerShell `Out-File` 명령과 함께 AWS CLI를 사용하는 것이 좋습니다.

주제

파일에서 파라미터를 로드하는 방법

모두 명령줄 파라미터 값으로 입력하려고 하는 것보다 파일에서 파라미터 값을 가져오는 것이 편리할 때가 있습니다(예: 파라미터가 복잡한 JSON 문자열일 때). 값을 포함하는 파일을 지정하려면 파일 URL을 다음 형식으로 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.
- URL은 실제 파라미터 내용이 포함된 파일로의 경로를 제공합니다.
- 공백이나 특수 문자가 있는 파일을 사용하는 경우 터미널의 인용 및 이스케이프 규칙을 따릅니다.

> **Note**
>
> 이미 URL을 예측한 파라미터(예: AWS CloudFormation 템플릿 URL를 식별하는 파라미터)에서는 이 동작이 자동으로 비활성화됩니다. 또한 AWS CLI 구성 파일에서 `cli_follow_urlparam` 설정을 사용 중지하여 이 동작을 사용 중지할 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 `%USERPROFILE%` 아래에 사용자 디렉터리가 있습니다.

다른 JSON 문서 값으로 포함되는 JSON 문서는 계속 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 이진 256비트 AES 키를 생성한 다음, 이를 Amazon S3에 제공하여 업로드된 파일 서버측을 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

또한 AWS CLI는 `http://` 또는 `https://` URL을 사용하여 인터넷에서 호스트되는 파일에서 파라미터 로드를 지원합니다. 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
```

위의 예제에서는 `filename.json` 파일에 다음 JSON 데이터가 포함된 것으로 가정합니다.

```
[
  {
    "DeviceName": "/dev/sdb",
    "Ebs": {
      "VolumeSize": 20,
      "DeleteOnTermination": false,
      "VolumeType": "standard"
    }
  }
]
```

JSON 형식의 파라미터를 포함하는 파일을 참조하는 다른 예제는 사용자에게 IAM 관리형 정책 연결 섹션을 참조하세요.

파일을 간편 구문 값으로 로드

값이 크거나 복잡한 간편 구문을 사용하는 경우 파일에 값으로 로드하는 것이 더 쉬운 경우가 많습니다. 파일을 간편 구문 값으로 로드하려면 형식이 약간 변경됩니다. `key=value` 대신 `=` 연산자 대신 `@=` 연산자를 사용합니다. `@=`는 AWS CLI에 값을 문자열이 아닌 파일 경로로 읽어야 함을 보여줍니다. 다음 예제에서는 해당 값에 대한 파일을 로드하는 키-값 페어를 보여줍니다.

Linux or macOS

```
--option key@=file://template.txt
```

Windows

```
--option "key1@=file://template.txt"
```

다음 예제에서는 `aws rolesanywhere create-trust-anchor` 명령에 대한 인증서 파일을 로드하는 방법을 보여줍니다.

```
$ aws rolesanywhere create-trust-anchor --name TrustAnchor \
    --source sourceData={x509CertificateData@=file://root-
ca.crt},sourceType="CERTIFICATE_BUNDLE"  \
```

```
    --enabled
```

간편 구문에 대한 자세한 내용은 the section called “간편 구문” 섹션을 참조하세요.

AWS CLI에서 AWS CLI 스켈레톤 및 입력 파일

대부분의 AWS CLI 명령은 파일의 모든 파라미터 입력 가져오기를 허용합니다. 이러한 템플릿은 `generate-cli-skeleton` 옵션을 사용하여 생성한 다음 `--cli-input-json` 파라미터를 사용하여 가져올 수 있습니다.

주제

AWS CLI 스켈레톤 및 입력 파일 정보

대부분의 AWS Command Line Interface(AWS CLI) 명령은 `--cli-input-json` 파라미터를 사용하여 파일에서 파라미터 입력을 허용하는 기능을 지원합니다.

이러한 명령은 `--generate-cli-skeleton` 파라미터를 사용하므로 모든 파라미터를 편집하거나 입력할 수 있는 JSON 형식의 파일을 만들 수 있습니다. 그러면 `--cli-input-json` 파라미터를 사용하여 명령을 실행하고 입력된 파일을 가리킬 수 있습니다.

> **Important**
>
> `aws s3` 명령 같은 사용자 지정 AWS CLI 명령은 이 주제에서 다루는 `--generate-cli-skeleton` 또는 `--cli-input-json` 파라미터를 지원하지 않습니다. 특정 명령이 이러한 파라미터를 지원하는지 확인하려면 사용하려는 명령에 대한 `help` 명령을 실행하거나 AWS CLI 버전 1 참조 가이드를 참조하세요.

`--generate-cli-skeleton`은 명령에서 사용자 지정하고 입력으로 사용할 수 있는 파라미터 템플릿을 생성하고 표시합니다. 생성된 템플릿에는 명령이 지원하는 모든 파라미터가 포함됩니다.

`--generate-cli-skeleton` 파라미터는 다음 값 중 하나를 허용합니다.

- `input` - 생성된 템플릿에 JSON 형식의 모든 입력 파라미터가 포함됩니다. 이것이 기본값입니다.
- `output` - 생성된 템플릿에 JSON 형식의 모든 출력 파라미터가 포함됩니다.

AWS CLI는 본질적으로 서비스의 API를 둘러싼 "래퍼"이므로, 스켈레톤 파일은 사용자가 모든 파라미터를 기본 API 파라미터 이름으로 참조할 것이라고 예상합니다. 이는 AWS CLI 파라미터 이름과 다를 수 있습니다. 예를 들어 이름이 AWS CLI인 `user-name` 파라미터는 이름이 AWS인 `UserName` 서비스의 API 파라미터에 매핑될 수 있습니다(변경된 대소문자 표시와 누락된 대시에 유의). 실수를 방지하려면 `--generate-cli-skeleton` 옵션을 사용하여 "정확한" 파라미터 이름으로 템플릿을 생성하는 것이 좋습니다. 서비스 API 참조 안내서를 참조하여 예상되는 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 run-instances에 대한 파라미터 템플릿이 생성됩니다.

JSON

다음 예제에서는 `input` 파라미터에 기본값(`--generate-cli-skeleton`)을 사용하여 JSON 형식의 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 및 가져오기

파라미터 스켈레톤 파일을 생성하고 사용하려면

1. `--generate-cli-skeleton` 파라미터와 함께 명령을 실행하여JSON을 생성하고 출력을 파일로 전송하여 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. 불필요한 요소를 제거한 후에도 파일이 여전히 유효한 JSON인지 확인합니다.

   JSON

   ```
   {
       "DryRun": true,
       "ImageId": "",
       "KeyName": "",
       "SecurityGroups": [
           ""
       ],
       "InstanceType": "",
       "Monitoring": {
           "Enabled": true
       }
   }
   ```

   이 예제에서는 Amazon EC2 테스트 실행 기능을 사용하도록 `DryRun` 파라미터를 `true`로 설정된 상태로 유지합니다. 이 기능을 사용하면 실제로 리소스를 생성하거나 수정하지 않고도 명령을 안전하게 테스트할 수 있습니다.

3. 나머지 값을 시나리오에 적합한 값으로 채우세요. 이 예제에서는 사용할 Amazon Machine Image(AMI)의 인스턴스 유형, 키 이름, 보안 그룹 및 식별자를 제공합니다. 이 예제에서는 기본 AWS 리전을 가정합니다. `ami-dfc39aef` AMI는 `us-west-2` 리전에서 호스팅되는 64비트 Amazon Linux 이미지입니다. 다른 리전을 사용하는 경우 사용할 정확한 AMI ID를 찾아야 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. AWS CLI는 경로가 현재 작업 디렉터리를 기준으로 하는 것으로 해석합니다. 다음 예제에서 AWS CLI는 현재 작업 디렉터리에서 파일을 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.
```

   테스트 실행 오류는 JSON 이 올바르게 구성되었으며 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 `false`로 설정할 수 있습니다.

   JSON

```
{
```

```
    "DryRun": false,
    "ImageId": "ami-dfc39aef",
    "KeyName": "mykey",
    "SecurityGroups": [
        "my-sg"
    ],
    "InstanceType": "t2.micro",
    "Monitoring": {
        "Enabled": true
    }
}
```

6. 명령을 실행하면 `run-instances`가 실제로 Amazon EC2 인스턴스를 실행하고 성공적인 실행으로 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": [
   ...
   ```

입력 파일과 명령줄 파라미터 결합

입력 파일은 모든 파라미터에 사용하거나 AWS CLI에 지정된 파라미터와 결합할 수 있습니다. 명령 자체에 개별 설정을 유지하면서 입력 파일에서 자주 재사용하는 설정에이 기능을 사용할 수 있습니다.

다음 `aws ec2 run-instances` 예제에서는 입력 파일과 파라미터의 사용을 결합합니다. 사용할 Amazon Machine Image(AMI)의 인스턴스 유형, 키 이름, 보안 그룹 및 식별자가 제공되고 기본 AWS 리전을 가정합니다. `ami-dfc39aef` AMI는 `us-west-2` 리전에서 호스팅되는 64비트 Amazon Linux 이미지입니다. 다른 리전을 사용하는 경우 [사용할 정확한 AMI ID를 찾아야] 합니다.

JSON

JSON 파일의 콘텐츠:

```
{
    "ImageId": "ami-dfc39aef",
    "KeyName": "mykey",
    "SecurityGroups": [
        "my-sg"
    ],
    "InstanceType": "t2.micro",
    "Monitoring": {
        "Enabled": true
    }
}
```

다음 예제에서는 입력 파일을 `--dry-run` 파라미터와 함께 사용하여 명령의 드라이런을 수행하여 필요한 권한이 있고 파일을 유효한 값으로 입력했는지 확인합니다.

JSON

```
$ aws ec2 run-instances --cli-input-json file://ec2runinst.json --dry-run
```

```
A client error (DryRunOperation) occurred when calling the RunInstances operation:
 Request would have succeeded, but DryRun flag is set.
```

다음 예제에서는 동일한 입력 파일을 사용하지만 `--no-dry-run` 파라미터와 함께 명령을 전체로 수행합니다.

JSON

```
$ aws ec2 run-instances --cli-input-json file://ec2runinst.json --no-dry-run --
output json
```

```
{
    "OwnerId": "123456789012",
    "ReservationId": "r-d94a2b1",
    "Groups": [],
    "Instances": [
...
```

AWS CLI에서 간편 구문 사용

AWS Command Line Interface(AWS CLI)는 JSON 형식으로 많은 옵션 파라미터를 허용할 수 있습니다. 그러나 긴 JSON 목록이나 구조를 명령줄에 입력하려면 지루할 수 있습니다. 입력을 쉽게 하기 위해 AWS CLI는 전체 JSON 형식을 사용하는 것보다 더 간단하게 옵션 파라미터를 표시할 수 있는 간편 구문도 지원합니다.

주제

키-값 페어가 있는 구조 파라미터

AWS CLI에서 간편 구문을 사용하면 사용자가 플랫(중첩되지 않은 구조) 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
```

JSON 형식의 다음 예와 동등합니다.

```
--option '{"key1":"value1","key2":"value2","key3":"value3"}'
```

Windows

```
--option "key1=value1,key2=value2,key3=value3"
```

JSON 형식의 다음 예와 동등합니다.

```
--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
```

이 구문은 JSON 형식의 다음 예와 동등합니다.

```
$ aws dynamodb update-table \
    --provisioned-throughput '{"ReadCapacityUnits":15,"WriteCapacityUnits":10}' \
    --table-name MyDDBTable
```

파일을 간편 구문 값으로 로드

값이 크거나 복잡한 경우 값으로 로드하는 것이 더 쉬운 경우가 많습니다. 파일을 간편 구문 값으로 로드하려면 형식이 약간 변경됩니다. key=value 대신 = 연산자 대신 @= 연산자를 사용합니다. @=는 AWS CLI에게 값을 문자열이 아닌 파일 경로로 읽어야 함을 보여줍니다. 파일을 간편 구문으로 로드할 때는 일반적인 AWS CLI 파일 형식 지정 규칙이 적용됩니다. 다음 예제에서는 해당 값에 대한 파일을 로드하는 키-값 페어를 보여줍니다.

Linux or macOS

```
--option key@=file://template.txt
```

Windows

```
--option "key1@=file://template.txt"
```

다음 예제에서는 aws rolesanywhere create-trust-anchor 명령에 대한 인증서 파일을 로드하는 방법을 보여줍니다.

```
$ aws rolesanywhere create-trust-anchor --name TrustAnchor \
    --source sourceData={x509CertificateData@=file://root-
ca.crt},sourceType="CERTIFICATE_BUNDLE"  \
    --enabled
```

AWS CLI에서 간편 구문 사용

목록 형식의 입력 파라미터는 JSON 또는 간편 구문이라는 두 가지 방법으로 지정할 수 있습니다. AWS CLI의 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
```

이 구문은 JSON 형식의 다음 예와 동등합니다.

```
--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(Amazon EC2)에 대한 `stop-instances` 명령의 예입니다. 여기서 `--instance-ids` 옵션에 대한 입력 파라미터(문자열 목록)는 간편 방식으로 지정됩니다.

```
$ aws ec2 stop-instances \
    --instance-ids i-1486157a i-1286157c i-ec3a7e87
```

이 구문은 JSON 형식의 다음 예와 동등합니다.

```
$ aws ec2 stop-instances \
    --instance-ids '["i-1486157a","i-1286157c","i-ec3a7e87"]'
```

다음 예제는 Amazon EC2 `create-tags` 명령을 보여줍니다. 이 명령은 `--tags` 옵션에 대한 비중첩 구조 목록을 가져옵니다. `--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
```

이 구문은 JSON 형식의 다음 예와 동등합니다. JSON 파라미터는 쉽게 읽을 수 있도록 여러 줄로 작성됩니다.

```
$ aws ec2 create-tags \
    --resources i-1286157c \
    --tags '[
        {"Key": "My1stTag", "Value": "Value1"},
        {"Key": "My2ndTag", "Value": "Value2"},
        {"Key": "My3rdTag", "Value": "Value3"}
    ]'
```

AWS CLI에서 명령 출력 제어

이 섹션에서는 AWS Command Line Interface(AWS CLI)의 출력을 제어하는 다양한 방법에 대해 설명합니다. 터미널에서 AWS CLI 출력을 사용자 지정하면 가독성을 개선하고 스크립팅 자동화를 간소화하며 대규모 데이터세트를 더 쉽게 탐색할 수 있습니다.

AWS CLI는 `json`, `text`, 및 `table`을 비롯한 다양한 출력 형식을 지원합니다. 일부 서비스에는 데이터에 대한 서버 측 페이지 매김이 있으며,.

AWS CLI에는 개별적으로 또는 함께 사용하여 AWS CLI 출력을 필터링할 수 있는 서버 측 필터링과 클라이언트 측 필터링이 둘 다 있습니다.

주제

민감한 출력

AWS CLI의 일부 작업은 환경 변수의 정보를 포함하여 민감한 것으로 간주될 수 있는 정보를 반환할 수 있습니다. 이러한 정보의 노출은 특정 시나리오에서 보안 위험을 나타낼 수 있습니다. 예를 들어, 이러한 정보는 지속적 통합 및 지속적 배포(CI/CD) 로그에 포함될 수 있습니다. 따라서 이러한 출력을 언제 로그의 일부로 포함할지 검토하고 필요하지 않은 경우 출력을 억제하는 것이 중요합니다.

민감한 데이터 보호에 대한 자세한 내용은 the section called “데이터 보호”를 참조하세요.

다음 모범 사례를 고려하세요.

- AWS Secrets Manager와 같은 암호 저장소에서 프로그래밍 방식으로 암호를 검색하는 것을 좋습니다.
- 빌드 로그의 내용을 검토하여 민감한 정보가 포함되어 있지 않은지 확인합니다. `/dev/null`로 파이핑하거나 출력을 bash 또는 PowerShell 변수로 캡처하여 명령 출력을 억제하는 등의 접근 방식을 고려합니다.

  다음은 오류가 아닌 출력을 `/dev/null`로 리디렉션하는 bash 예제입니다.

```
$ aws s3 ls > /dev/null
```

터미널의 출력을 억제하는 방법에 대한 자세한 내용은 사용 중인 터미널의 사용 설명서를 참조하세요.

- 로그의 액세스 권한을 고려하여 사용 사례에 맞게 액세스 범위를 적절히 설정하세요.

서버 측 출력 옵션과 클라이언트 측 출력 옵션 비교

AWS CLI에는 개별적으로 또는 함께 사용하여 AWS CLI 출력을 필터링할 수 있는 서버 측 필터링과 클라이언트 측 필터링이 둘 다 있습니다. 서버 측 필터링이 먼저 처리되고 클라이언트 측 필터링에 대한 출력을 반환합니다. 서버 측 필터링은 서비스 API에서 지원됩니다. 클라이언트 측 필터링은 AWS CLI 매개 변수를 사용하여 `--query` 클라이언트에서 지원됩니다.

서버 측 출력 옵션은 AWS 서비스 API에서 직접 지원합니다. 필터링되거나 페이징된 데이터는 클라이언트로 전송되지 않으므로 HTTP 응답 시간이 빨라지고 대규모 데이터세트의 대역폭을 개선할 수 있습니다.

클라이언트 측 출력 옵션은 AWS CLI에서 만든 기능입니다. 모든 데이터가 클라이언트로 전송되면 AWS CLI가 표시되는 콘텐츠를 필터링하거나 페이지를 매깁니다. 클라이언트 측 작업으로는 대규모 데이터세트의 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.

AWS CLI의 출력 형식 설정

이 주제에서는 AWS Command Line Interface(AWS CLI)의 다양한 출력 형식에 대해 설명합니다. AWS CLI은(는) 다음 출력 형식을 지원합니다.

- **`json`** - 출력은 JSON 문자열로 형식이 지정됩니다.
- **`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 text로 설정합니다.

```
[default]
output=text
```

- **AWS_DEFAULT_OUTPUT** 환경 변수 사용 - 다음 출력은 변수가 변경되거나 세션이 끝날 때까지 이 명령줄 세션의 명령 형식을 table로 지정합니다. 이 환경 변수를 사용하면 config 파일에 설정된 값을 재정의합니다.

```
$ export AWS_DEFAULT_OUTPUT="table"
```

- 명령줄에서 **--output** 옵션 사용 - 다음 예제에서는 이 명령의 출력만 json으로 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 text를 지정하면 --query 필터를 적용하기 전에 해당 출력이 페이지 매김되며, AWS CLI는 해당 출력의 각 페이지에서 해당 쿼리를 한 번 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. AWS CLI은(는) 전체 구조에 대해 한 번만 쿼리를 실행하여 필터링된 결과를 생성한 다음 출력합니다.

JSON 출력 형식

JSON은 AWS CLI의 기본 출력 형식입니다. 대부분의 프로그래밍 언어는 기본 제공된 기능 또는 공개적으로 사용 가능한 라이브러리를 사용하여 JSON 문자열을 쉽게 디코딩할 수 있습니다. JSON 출력을 강력한 방법으로 --query 옵션과 결합하여 AWS CLI JSON 형식의 출력을 필터링하고 서식을 지정할 수 있습니다.

`--query`로 수행할 수 없는 고급 필터링의 경우 명령줄 JSON 프로세서인 `jq`를 고려할 수 있습니다. http://stedolan.github.io/jq/에서 이 처리기를 다운로드하고 공식 자습서를 찾아볼 수 있습니다.

다음은 JSON 출력의 예제입니다.

```
$ 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, sed 및 awk와 같은 기존 Unix 텍스트 도구 및 PowerShell에서 수행하는 텍스트 처리에 효과적입니다.

text 출력 형식은 아래와 같은 기본 구조를 따릅니다. 열은 기본 JSON 객체의 해당 키 이름을 기준으로 알파벳 순서로 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 Console 콘솔에 로그인하지 않기 때문에 마지막 두 항목은 비어 있습니다.

> **Important**
>
> *text* 출력을 지정하는 경우 일관된 동작을 보장하기 위해 항상 *--query* 옵션도 사용하는 것이 좋습니다.
> 텍스트 형식이 출력 열을 AWS 서비스에서 반환하는 기본 JSON 객체의 키 이름에 따라 알파벳 순서로 정렬하고, 유사한 리소스에 항상 동일한 키 이름이 있는 것은 아니기 때문입니다. 예를 들어 Linux 기반 Amazon EC2 인스턴스의 JSON 표시에는 Windows 기반 인스턴스의 JSON 표시에 없는 요소가 있을 수 있으며 반대의 경우도 마찬가지입니다. 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 CLI에서 존재하지 않는 키 값으로 None을 출력하는 방식을 살펴보세요.

```
$ 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-instances`를 사용하는 방법을 보여줍니다. 첫 번째 명령은 `text` 출력에 각 인스턴스의 가용 영역, 현재 상태 및 인스턴스 ID를 표시합니다. 두 번째 명령은 `us-west-2a` 가용 영역에서 실행 중인 모든 인스턴스의 인스턴스 ID만 표시하는 출력을 처리합니다.

```
$ 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 PowerShell에서도 유용할 수 있습니다. text 출력의 열은 탭으로 구분되어 있으므로 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 예제에서 --query 파라미터를 사용하여 기본 JSON 객체를 구문 분석하고 원하는 열을 추출하는 방법을 알아보았지만, 교차 플랫폼 호환성이 문제가 되지 않을 경우 PowerShell 자체 기능으로 JSON을 처리할 수 있습니다. 대부분의 명령 셸에서 요구하듯이 출력을 텍스트로 처리하는 대신, PowerShell에서는 ConvertFrom-JSON cmdlet을 사용하여 계층적으로 구조화된 객체를 생성할 수 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 주의하세요. 첫 번째 예제에서는 열 이름이 알파벳 순서로 정렬되고 두 번째 예제에서는 이름 없는 열이 사용자가 정의한 방식으로 정렬됩니다. --query 옵션에 대한 자세한 내용은 AWS CLI에서 출력 필터링 섹션을 참조하세요.

```
$ 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  |
+--------------+--------------+--------------+-----+
```

AWS CLI의 페이지 매김 옵션 사용

이 주제에서는 AWS Command Line Interface(AWS CLI)의 출력에 페이지 번호를 매기는 다양한 방법에 대해 설명합니다.

서버 측 페이지 매김

많은 항목 목록을 반환하는 대부분의 명령의 경우, AWS CLI에는 AWS CLI에서 서비스의 API를 호출하여 목록을 채울 때 출력에 포함되는 항목의 수를 제어할 수 있는 여러 옵션이 있습니다. AWS CLI의 서버 측 페이지 매김은 AWS 서비스 API에서 활성화되므로 이러한 옵션은 서비스 API에서 활성화한 경우에만 작동합니다.

대부분의 AWS CLI 명령에는 다음 옵션이 포함되어 있습니다.

- --no-paginate 파라미터를 사용하는 방법
- --page-size 파라미터를 사용하는 방법
- --max-items 파라미터를 사용하는 방법
- --starting-token 파라미터를 사용하는 방법

기본적으로 AWS CLI는 개별 서비스에 의해 결정된 페이지 크기를 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 CLI는 백그라운드에서 서비스별 페이지 매김 로직을 처리하고 최종 출력에 3,500개 객체를 모두 반환하면서 Amazon S3에 대한 4개 호출을 자동으로 작성합니다.

특정 명령에 서버 측 페이지 매김이 있는지에 대한 자세한 내용은 AWS CLI 참조 가이드를 참조하세요.

--no-paginate 파라미터를 사용하는 방법

`--no-paginate` 옵션은 클라이언트 측에서 다음 페이지 매김 토큰을 사용 중지합니다. 명령을 사용할 때 기본적으로 AWS CLI는 자동으로 여러 번 호출하여 가능한 모든 결과를 반환해서 페이지 매김을 생성합니다. 각 페이지에 대해 한 번 호출합니다. 페이지 매김을 비활성화하면 AWS CLI가 명령 결과의 첫 페이지에 대해 한 번만 호출합니다.

예를 들어 3,500개의 객체가 포함된 Amazon S3 버킷에서 `aws s3api list-objects`를 실행하는 경우 AWS CLI는 Amazon S3에 대한 첫 번째 호출만 실행하여 최종 출력에서 처음 1,000개의 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. 이 경우 AWS 서비스에 최대 허용 시간을 초과하는 호출이 이루어지고 "시간 초과" 오류가 발생할 수 있습니다. `--page-size` 옵션을 사용하면 AWS CLI가 각 AWS 서비스 호출로부터 더 적은 수의 항목을 요청하도록 지정할 수 있습니다. AWS CLI는 계속 전체 목록을 검색하지만, 백그라운드에서 더 많은 수의 서비스 API 호출을 수행하고 각 호출마다 더 적은 수의 항목을 검색합니다. 그러면 각각의 호출이 시간 초과 없이 성공할 확률이 높아집니다. 페이지 크기를 변경해도 출력에 영향을 주지 않습니다. 출력을 생성하는 데 필요한 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 포함시키려면 `--max-items` 옵션을 사용합니다. AWS CLI는 이전에 설명한 대로 계속 서비스와 함께 페이지 매김을 처리하지만, 사용자가 지정한 항목 수만 한 번에 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 반환하는 전체 항목 수보다 적을 경우 사용자가 다음 항목 세트를 검색하기 위해 후속 명령에 전달할 수 있도록 출력에 `NextToken`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 같은 순서로 항목이 반환되지는 않습니다. `--page-size` 및 `--max-items`에 서로 다른 값을 지정하면 누락되거나 중복된 항목을 포함해 예상치 못한 결과가 발생할 수 있습니다. 이를 방지하려면 `--page-size` 및 `--max-items`에 동일한 번호를 사용하여 AWS CLI의 페이지 매김을 기본 서비스의 페이지 매김과 동기화하세요. 또한 전체 목록을 검색하고 필요한 구문 분석 작업을 로컬에서 수행할 수 있습니다.

AWS CLI에서 출력 필터링

AWS Command Line Interface(AWS CLI)에는 개별적으로 또는 함께 사용하여 AWS CLI 출력을 필터링할 수 있는 서버 측 필터링과 클라이언트 측 필터링이 둘 다 있습니다. 서버 측 필터링이 먼저 처리되고 클라이언트 측 필터링에 대한 출력을 반환합니다.

- 서버 측 필터링은 API에서 지원되며, 일반적으로 `--filter` 매개 변수를 사용하여 구현합니다. 이 서비스는 대량 데이터세트에 대한 HTTP 응답 시간을 단축할 수 있는 일치하는 결과만 반환합니다.
- 클라이언트 측 필터링은 AWS CLI 매개 변수를 사용하여 `--query` 클라이언트에서 지원됩니다. 이 파라미터에는 서버 측 필터링에 없을 수 있는 기능이 있습니다.

주제

- 추가 리소스

서버 측 필터링

AWS CLI의 서버 측 필터링은 AWS 서비스 API에서 제공됩니다. AWS 서비스는 필터와 일치하는 HTTP 응답의 레코드만 반환하며, 이렇게 되면 대량 데이터세트에 대한 HTTP 응답 시간을 단축할 수 있습니다. 서버 측 필터링은 서비스 API에 의해 정의되므로, 매개 변수 이름과 함수는 서비스마다 달라집니다. 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 필터링과 필터링 규칙이 있는지에 대한 자세한 내용은 AWS CLI 참조 가이드를 참조하세요.

클라이언트 측 필터링

AWS CLI은(는) `--query` 매개 변수를 사용하는 기본 제공 JSON 기반 클라이언트 측 필터링 기능을 제공합니다. `--query` 매개 변수는 출력의 내용과 스타일을 사용자 지정하는 데 사용할 수 있는 강력한 도구입니다. `--query` 매개 변수는 서버에서 다시 오는 HTTP 응답을 받아, 결과를 필터링한 후에 이를 표시합니다. 필터링하기 전에 전체 HTTP 응답이 클라이언트로 전송되므로, 클라이언트 측 필터링은 대량 데이터세트에 대한 서버 측 필터링보다 느릴 수 있습니다.

쿼리는 JMESPath 구문을 사용하여 출력을 필터링하는 표현식을 만듭니다. JMESPath 구문에 대해 알아보려면 JMESPath 웹사이트에서 자습서를 참조하세요.

> ⚠ Important
>
> 지정한 출력 유형에 따라 `--query` 옵션 작동 방식이 변경됩니다.
>
> - `--output text`를 지정하면 `--query` 필터를 적용하기 전에 해당 출력이 페이지 매김되며, AWS CLI는 해당 출력의 각 페이지에서 해당 쿼리를 한 번 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. AWS CLI은(는) 전체 구조에 대해 한 번만 쿼리를 실행하여 필터링된 결과를 생성한 다음 출력합니다.

클라이언트 측 필터링 주제

- 시작하기 전에
- 식별자
- 목록에서 선택
- 중첩된 데이터 필터링
- 결과 병합
- 특정 값에 대한 필터링
- 파이핑 표현식
- 여러 식별자 값에 대한 필터링
- 식별자 값에 레이블 추가
- 함수
- 고급 --query 예제

시작하기 전에

> **Note**
>
> 이러한 필터 표현식 예제는 기본 Linux와 유사한 쉘에 대해 작성되었습니다. 이 예제를 사용할 때는 터미널 셸에 올바른 인용 규칙을 사용해야 합니다. 터미널이 입력을 해석하는 방식은 AWS CLI로 전송되는 항목을 크게 변경할 수 있습니다. 터미널이 작은따옴표('), 큰따옴표(") 또는 백틱(`)을 읽는 방법은 콘텐츠 읽기 방법을 변경할 수 있습니다.
> 자세한 내용은 the section called “문자열과 따옴표” 섹션을 참조하세요.

다음 JSON 출력은 `--query` 매개 변수가 생성할 수 있는 결과물의 예를 보여줍니다. 이 출력은 별도의 Amazon EC2 인스턴스에 연결된 세 가지 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 웹사이트의 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 0이고 `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 웹사이트의 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 하위 표현식을 참조하세요.

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 웹사이트의 병합을 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 `VolumeIds``Volumes`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 필터 표현식을 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 VolumeType이(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 State을(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
    }
]
```

최신 AMI를 표시하는 방법

다음 예제에서는 가장 최근에 생성된 5개의 Amazon Machine Image(AMI)를 가장 최근부터 가장 오래된 순으로 정렬하여 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. 대량 데이터세트를 사용하는 경우, 먼저 서버 측 필터링을 사용하면 클라이언트 측 필터링이 제공하는 강력한 사용자 지정을 유지하면서 각 AWS CLI 호출에 대해 클라이언트로 전송되는 데이터 양을 줄일 수 있습니다.

다음 예제에서는 서버 측 필터링과 클라이언트 측 필터링을 둘 다 사용하는 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 `CreationDate`를 기준으로 출력을 정렬하고 가장 최근 항목만 선택합니다. 마지막으로 해당 이미지의 `ImageId`를 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 `length`로 목록의 항목 수를 계산하여 1000 IOPS 이상인 사용 가능한 볼륨 수를 표시합니다.

```
$ aws ec2 describe-volumes \
    --filters "Name=status,Values=available" \
    --query 'length(Volumes[?Iops > `1000`])'
3
```

다음 예시에서는 AWS CloudFormation 스택을 사용하는 지정된 AWS 리전에서 시작 구성을 사용하는 Auto Scaling 그룹의 이름을 검색합니다.

```
$ aws autoscaling describe-auto-scaling-groups --region us-west-2 \
  --filters Name=tag-key,Values=aws:cloudformation:stack-name \
  --query 'AutoScalingGroups[?LaunchConfigurationName!=`null`].AutoScalingGroupName'
[
    "group-1",
    "group-2",
    "group-3"
]
```

추가 리소스

JMESPath Terminal

JMESPath Terminal은 클라이언트 측 필터링에 사용되는 JMESPath 표현식을 실험하는 대화형 터미널 명령입니다. `jpterm` 명령을 사용하면 사용자가 입력할 때 터미널에 즉시 쿼리 결과가 표시됩니다. AWS CLI 출력을 터미널에 직접 파이핑하여 고급 쿼리 실험을 할 수 있습니다.

다음 예제에서는 `aws ec2 describe-volumes` 출력을 JMESPath Terminal에 직접 파이핑합니다.

```
$ aws ec2 describe-volumes | jpterm
```

JMESPath Terminal 및 설치 지침에 대한 자세한 내용은 GitHub의 [JMESPath Terminal]을 참조하세요.

jq 유틸리티

jq 유틸리티는 클라이언트 측의 출력을 사용자가 원하는 출력 형식으로 변환하는 방법을 제공합니다. `jq` 및 설치 지침에 대한 자세한 내용은 GitHub의 [jq]를 참조하세요.

AWS CLI의 명령줄 반환 코드

반환 코드는 대개 명령의 상태를 설명하는 AWS Command Line Interface(AWS CLI) 명령을 실행한 후 전송되는 숨겨진 코드입니다. echo 명령을 사용하여 마지막 AWS CLI 명령에서 보낸 코드를 표시하고, 이러한 코드를 사용하여 명령이 성공했는지 또는 실패했는지, 명령에 오류가 있는 이유를 확인할 수 있습니다. 반환 코드 외에도 --debug 스위치로 명령을 실행하여 실패에 대한 자세한 정보를 볼 수 있습니다. 이렇게 하면 AWS CLI가 명령을 처리하기 위해 사용하는 단계와 각 단계의 결과가 포함된 세부 보고서가 생성됩니다.

AWS CLI 명령의 반환 코드를 확인하려면 CLI 명령을 실행한 즉시 다음 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) 명령의 실행이 끝났을 때 반환될 수 있는 반환 코드 값입니다.

코드	의미
0	서비스가 HTTP 응답 상태 코드 200으로 응답했습니다. 이는 요청이 전송된 AWS CLI 및 AWS 서비스에서 발생한 오류가 없음을 나타냅니다.
1	하나 이상의 Amazon S3 전송 작업이 실패했습니다. S3 명령으로 제한됩니다.
2	이 반환 코드의 의미는 명령에 따라 달라집니다. • 모든 AWS CLI 명령에 적용 가능 – 입력된 명령은 구분 분석할 수 없습니다. 필수적인 하위 명령 또는 인수가 누락되거나 알려지지 않은 명령 또는 인수를 사용한 것이 구문 분석이 실패한 이유 중 하나일 수 있습니다. • S3 명령으로 제한됨 - 전송 프로세스에서 전송 대상으로 표시된 파일을 하나 이상 건너뛰었습니다. 그러나 이전 대상으로 표시된 다른 모든 파일들은 성공적으로 전송되었습니다. 이전 과정 중에 건너 뛴 파일에는 존재하지 않는 파일, 문자 특별 장치, 블록 특별 장치, FIFO 대기열 또는 소켓인 파일, 사용자가 읽기 권한을 가지고 있지 않은 파일이 포함됩니다.
130	SIGINT에 의해 명령이 중단되었습니다. 이것은 `Ctrl+C`로 명령을 취소하기 위해 사용자가 보낸 신호입니다.
255	명령이 실패했습니다. 요청이 전송된 AWS CLI 또는 AWS 서비스에서 오류가 생성되었습니다.

AWS CLI에서 별칭 생성 및 사용

별칭은 자주 사용하는 명령이나 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 AWS CLI 설치 및 AWS CLI에 대한 인증 및 액세스 보안 인증 섹션을 참조하세요.
- 최소 AWS CLI 버전 1.11.24 또는 2.0.0을 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 단계별 절차를 사용하여 원하는 터미널을 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 이름이 `cli`인 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]
```

*aliasname*은 별칭을 말합니다. *command*는 호출할 명령이며, 다른 별칭을 포함할 수 있습니다. 별칭에 옵션 또는 파라미터를 포함하거나 별칭을 호출할 때 추가할 수 있습니다.

다음 예제에서는 [aws sts get-caller-identity](https://docs.aws.amazon.com/cli/latest/reference/sts/get-caller-identity.html) 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 기능을 사용하려면 최소 AWS CLI 버전 1.11.24 또는 2.0.0이 필요합니다.

이전 단계에서 생성한 `alias` 파일에서 다음 구문을 사용해 명령을 추가하여 하위 명령 별칭을 생성할 수 있습니다.

구문

```
[command commandGroup]
aliasname = command [--options]
```

*CommandGroup*은 명령 네임스페이스입니다. 예를 들어 `aws ec2 describe-regions` 명령은 `ec2` 명령 그룹 아래에 있습니다. *aliasname*은 별칭을 말합니다. *command*는 호출할 명령이며, 다른 별칭을 포함할 수 있습니다. 별칭에 옵션 또는 파라미터를 포함하거나 별칭을 호출할 때 추가할 수 있습니다.

다음 예제에서는 [aws ec2 describe-regions](https://docs.aws.amazon.com/cli/latest/reference/ec2/describe-regions.html) 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
```

*`aliasname`*은 별칭을 말하며, *`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 인증합니다.

```
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 별칭](#)을 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 `$phone` 및 `$message`에 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

GitHub의 [AWS CLI 별칭 리포지토리]에는 AWS CLI 개발자 팀 및 커뮤니티에서 생성한 AWS CLI 별칭 예제가 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 GitHub에서 JMESPath (jp) README.md를 참조하세요.
3. `jq` 명령을 설치합니다. `jq` 명령은 `tostring-with-jq` 별칭에 사용됩니다. 설치 지침은 GitHub에서 JSON processor (jq)를 참조하세요.

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 다운로드하여 AWS CLI 구성 폴더의 cli 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

• GitHub의 AWS CLI 별칭 리포지토리에는 AWS CLI 개발자 팀 및 AWS CLI 커뮤니티의 기여로 생성된 AWS CLI 별칭 예제가 포함되어 있습니다.
• YouTube의 AWS re:Invent 2016: The Effective AWS CLI User에서 별칭 기능 발표
• aws sts get-caller-identity
• aws ec2 describe-instances
• aws sns publish

AWS CLI 코드 예제

이 장에서는 AWS 서비스에서 AWS Command Line Interface(AWS CLI)를 사용하는 방법을 보여주는 예제 모음을 제공합니다.

이 가이드에서 AWS CLI에는 다음과 같은 유형의 예제가 있습니다.

- 안내식 명령 예제 - 일부 AWS 서비스에서 AWS CLI를 사용하는 방법에 대한 AWS CLI 사용 설명서의 안내식 명령 예제입니다. 이 예제는 AWS CLI 참조 안내서의 예제보다 더 자세한 내용을 담고 있습니다.
- AWS CLI 명령 예제 - AWS CLI 참조 안내서에서도 사용할 수 있는 오픈 소스 명령 예제입니다. 명령 예제는 GitHub의 AWS CLI 리포지토리에서 호스팅됩니다.
- Bash 스크립팅을 사용하는 AWS CLI 코드 예제 - 오픈 소스 bash 스크립팅 예제입니다. Bash 스크립팅 예제는 GitHub의 AWS 코드 예제 리포지토리에서 호스팅됩니다.

> **예제 피드백**
>
> 필요한 예제를 찾을 수 없습니까? 이 페이지 하단의 피드백 제공 링크를 사용하거나 AWS CLI 참조 안내서의 관련 명령 페이지에서 명령 예제를 요청하세요.
> 기여하고 싶으십니까? GitHub의 AWS 코드 예제 리포지토리에서 AWS CLI 명령어 예제에 기여하세요. 기여에 대한 자세한 내용은 GitHub 페이지에서 AWS CLI 코드 예제 기여 빠른 단계를 참조하세요.

AWS CLI의 안내식 명령 예제

AWS Command Line Interface(AWS CLI)는 명령줄 셸의 명령을 사용하여 다양한 AWS 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다. 이 섹션에서는 AWS CLI를 활용하여 일부 AWS 서비스에 액세스하는 방법을 보여주는 안내 예제를 제공합니다. 여기에는 상위 수준 `aws s3` 명령과 같은 일부 사용자 지정 AWS CLI 명령이 포함됩니다. 이러한 명령 예제는 일부 AWS 서비스에 사용되는 일반적인 작업을 보여 주며 자세한 내용은 추가 리소스를 제공합니다.

숙련된 AWS 사용자든 처음 AWS CLI를 사용하는 사용자든 이 사용 설명서는 AWS 운영을 간소화하는 데 도움이 되는 리소스입니다.

각 AWS 서비스에 대해 사용 가능한 모든 명령어에 대한 전체 참조는 AWS CLI 참조 가이드를 참조하세요. 또한 기본 제공 명령줄 도움말을 활용하여 AWS 서비스, AWS CLI의 명령, 옵션 및 기능 배열을 탐색할 수 있습니다.

이 섹션에서 사용할 수 없는 더 많은 명령 예제는 AWS CLI 명령 예제 섹션을 참조하세요. 이러한 예제는 AWS CLI 참조 가이드에서도 확인할 수 있는 오픈 소스 명령어 예제입니다. 명령 예제는 GitHub의 AWS CLI 리포지토리에서 호스팅됩니다.

오픈 소스 bash 스크립팅 예제는 the section called “Bash 스크립트 예제” 섹션을 참조하세요. Bash 스크립팅 예제는 GitHub의 AWS 코드 예제 리포지토리에서 호스팅됩니다.

서비스

- AWS CLI의 Amazon DynamoDB 사용
- AWS CLI에서 Amazon EC2 사용
- AWS CLI에서 Amazon S3 Glacier 사용
- AWS CLI에서 IAM 사용
- AWS CLI에서 Amazon S3 사용
- AWS CLI에서 Amazon SNS 액세스

AWS CLI의 Amazon DynamoDB 사용

AWS Command Line Interface(AWS CLI)는 Amazon DynamoDB를 포함한 AWS 데이터베이스 서비스를 모두 지원합니다. 테이블 생성과 같이 특별 작업을 수행할 때 AWS CLI를 사용할 수 있습니다. 또한 이를 사용하여 DynamoDB 작업을 유틸리티 스크립트 내에 포함할 수 있습니다.

DynamoDB와 함께 AWS CLI를 사용하는 방법에 대한 자세한 내용은 AWS CLI 명령 참조에서 `dynamodb` 섹션을 참조하세요.

DynamoDB에 대한 AWS CLI 명령을 나열하려면 다음 명령을 사용합니다.

```
$ aws dynamodb help
```

주제

- 사전 조건
- DynamoDB 테이블 생성 및 사용

- DynamoDB Local 사용
- 리소스

사전 조건

dynamodb 명령을 실행하려면 다음을 수행해야 합니다.

- AWS CLI를 설치하고 구성합니다. 자세한 내용은 AWS CLI 설치 및 AWS CLI에 대한 인증 및 액세스 보안 인증(을)를 참조하세요.

DynamoDB 테이블 생성 및 사용

명령줄 형식은 DynamoDB 명령 이름과 해당 명령에 대한 파라미터 순서로 구성됩니다. AWS CLI는 파라미터 값의 CLI 간편 구문과 전체 JSON을 지원합니다.

다음 예제에서는 `MusicCollection`이라는 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. 본 예제에서 간편 구문과 JSON이 함께 사용됩니다.

```
$ 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"
    }
}
```

한 줄 명령에서는 유효한 JSON을 작성하기가 어려울 수 있습니다. 이를 쉽게 하기 위해 AWS CLI는 JSON 파일을 읽을 수 있습니다. 예를 들어 `expression-attributes.json`이라는 파일에 저장된 JSON 코드 조각을 고려해 보세요.

```
{
  ":v1": {"S": "No One You Know"},
  ":v2": {"S": "Call Me Today"}
}
```

이 파일을 사용하면 `query`를 사용하여 AWS CLI 요청을 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

DynamoDB에 더해 DynamoDB Local에도 AWS CLI를 사용할 수 있습니다. DynamoDB Local은 DynamoDB 서비스를 모방하는 클라이언트 측 소형 데이터베이스 및 서버입니다. DynamoDB Local을 사용하면 DynamoDB 웹 서비스의 테이블 또는 데이터를 조작하지 않고도 DynamoDB API를 사용하는 애플리케이션을 쓸 수 있습니다. 그 대신 모든 API 작업이 로컬 데이터베이스로 재라우팅됩니다. 이를 통해 프로비저닝된 처리량, 데이터 스토리지 및 데이터 전송 요금을 절감할 수 있습니다.

DynamoDB Local 및 AWS CLI와 함께 이를 사용하는 방법에 대한 자세한 내용은 Amazon DynamoDB 개발자 안내서에서 다음 섹션을 참조하세요.

- 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 CLI에서 DynamoDB Local 사용

AWS CLI에서 Amazon EC2 사용

Amazon Elastic Compute Cloud 소개
Amazon EC2 소개 - AWS 기반 탄력적 클라우드 서버 및 호스팅

Amazon Elastic Compute Cloud(Amazon EC2)는 확장성과 유연성이 뛰어난 가상 컴퓨팅 환경을 제공합니다. Amazon EC2를 사용하면 다양한 컴퓨팅 요구 사항을 충족하기 위해 Amazon EC2 인스턴스라고 하는 가상 서버를 프로비저닝하고 관리할 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 인스턴스까지 다양한 인스턴스 유형 중에서 선택할 수 있습니다. 이러한 유연성을 통해 컴퓨팅 요구 사항을 충족하여 성능과 비용 효율성을 최적화할 수 있습니다.

또한 Amazon EC2는 컴퓨팅 리소스를 효과적으로 관리할 수 있는 다양한 기능을 제공합니다. 여기에는 새 인스턴스를 빠르게 시작하고, 신속한 배포를 위해 사용자 지정 머신 이미지(AMI)를 생성하고, 필요에 따라 컴퓨팅 용량을 늘리거나 줄일 수 있는 기능이 포함됩니다.

AWS Command Line Interface(AWS CLI)를 사용하여 Amazon EC2의 기능에 액세스할 수 있습니다. Amazon EC2에 대한 AWS CLI 명령을 나열하려면 다음 명령을 사용합니다.

```
aws ec2 help
```

명령을 실행하기 전에 기본 자격 증명을 설정합니다. 자세한 내용은 AWS CLI 설정 구성 섹션을 참조하세요.

이 주제에서는 Amazon EC2에 대한 일반적인 태스크를 수행하는 짧은 형식의 AWS CLI 명령 예제를 보여줍니다.

긴 형식의 AWS CLI 명령 예제는 GitHub에서 AWS CLI 코드 예제 리포지토리를 참조하세요.

주제

- AWS CLI에서 Amazon EC2 키 페어 생성, 표시 및 삭제
- AWS CLI에서 Amazon EC2 보안 그룹 생성, 구성 및 삭제

- AWS CLI에서 Amazon EC2 인스턴스 시작, 나열 및 삭제
- AWS CLI에서 bash 스크립트로 Amazon EC2 인스턴스 유형 변경

AWS CLI에서 Amazon EC2 키 페어 생성, 표시 및 삭제

AWS Command Line Interface(AWS CLI)를 사용하여 Amazon Elastic Compute Cloud(Amazon EC2)에 대한 키 페어를 생성, 표시 및 삭제할 수 있습니다. 키 페어를 사용하여 Amazon EC2 인스턴스에 연결합니다. 인스턴스를 생성할 때 Amazon EC2에 키 페어를 제공한 다음, 인스턴스에 연결할 때 해당 키 페어를 사용하여 인증할 수 있습니다.

> **Note**
>
> 추가 명령 예제는 AWS CLI 참조 가이드를 참조하세요.

주제

사전 조건

`ec2` 명령을 실행하려면 다음을 수행해야 합니다.

- AWS CLI를 설치하고 구성합니다. 자세한 내용은 AWS CLI 설치 및 AWS CLI에 대한 인증 및 액세스 보안 인증(을)를 참조하세요.
- Amazon EC2 액세스를 허용하도록 IAM 권한을 설정합니다. Amazon EC2의 IAM 권한에 대한 자세한 내용은 Amazon EC2 사용 설명서의 Amazon EC2에 대한 IAM 정책을 참조하세요.

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
```

PowerShell의 경우 `> file` 리디렉션이 UTF-8 인코딩으로 기본 지정되는데, 일부 SSH 클라이언트에서는 이 인코딩을 사용할 수 없습니다. 따라서 `out-file` 명령으로 파이프하여 출력을 변환하고, 명시적으로 인코딩을 `ascii`로 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 키가 AWS에 저장되지 않고 이 키가 생성될 때만 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 DER 인코딩된 사본에서 가져온 SHA1 해시입니다. 이 값은 키 페어가 생성될 때 캡처되어 퍼블릭 키와 함께 AWS에 저장됩니다. Amazon EC2 콘솔에서 또는 AWS CLI 명령 `aws ec2 describe-key-pairs`를 실행하여 지문을 볼 수 있습니다.

다음 예제는 `MyKeyPair`의 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 EC2 사용 설명서의 Amazon EC2 키 페어를 참조하세요.

키 페어 삭제

키 페어를 삭제하려면 *MyKeyPair*를 삭제할 페어의 이름으로 바꾸어 `aws ec2 delete-key-pair` 명령을 실행합니다.

```
$ 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
- AWS SDK 및 AWS CLI 코드 예제를 보고 기여하려면 GitHub에서 AWS 코드 예제 리포지토리를 참조하세요.

AWS CLI에서 Amazon EC2 보안 그룹 생성, 구성 및 삭제

기본적으로 방화벽 역할을 하는 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스에 대한 보안 그룹과 함께, 들어오고 나가는 네트워크 트래픽을 결정하는 규칙을 생성할 수 있습니다.

AWS Command Line Interface(AWS CLI)를 사용하여 새 보안 그룹을 만들고, 기존 보안 그룹에 규칙을 추가하고, 보안 그룹을 삭제합니다.

> **Note**
> 추가 명령 예제는 AWS CLI 참조 가이드를 참조하세요.

주제

사전 조건

ec2 명령을 실행하려면 다음을 수행해야 합니다.

- AWS CLI를 설치하고 구성합니다. 자세한 내용은 AWS CLI 설치 및 AWS CLI에 대한 인증 및 액세스 보안 인증(을)를 참조하세요.

- Amazon EC2 액세스를 허용하도록 IAM 권한을 설정합니다. Amazon EC2의 IAM 권한에 대한 자세한 내용은 Amazon EC2 사용 설명서의 Amazon EC2에 대한 IAM 정책을 참조하세요.

보안 그룹 생성

Virtual Private Cloud(VPC)와 관련된 보안 그룹을 생성할 수 있습니다.

다음 `aws ec2 create-security-group` 예제에서는 지정된 VPC에 대한 보안 그룹을 생성하는 방법을 보여줍니다.

```
$ 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 아닌 `vpc-id`를 통해서만 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 이미지에 연결하는 방법에 맞는 수신 네트워크 트래픽을 허용하려면 보안 그룹의 규칙을 활성화해야 합니다.

예를 들어 Windows 인스턴스를 시작할 경우, RDP(Remote Desktop Protocol)를 지원하기 위해 일반적으로 TCP 포트 3389에서 인바운드 트래픽을 허용하는 규칙을 추가합니다. Linux 인스턴스를 시작할 경우, SSH 연결을 지원하기 위해 일반적으로 TCP 포트 22에서 인바운드 트래픽을 허용하는 규칙을 추가합니다.

`aws ec2 authorize-security-group-ingress` 명령을 사용하여 보안 그룹에 규칙을 추가합니다. 이 명령의 필수 파라미터가 컴퓨터 또는 컴퓨터가 연결된 네트워크(주소 범위의 형식)의 퍼블릭 IP 주소(CIDR 표기법 사용)입니다.

> **Note**
>
> 다음 서비스를 제공합니다. https://checkip.global.api.aws/으로 퍼블릭 IP 주소 확인을 활성화할 수 있습니다. IP 주소를 식별하는 데 도움이 되는 다른 서비스를 찾으려면 브라우저를 사용하여 "내 IP 주소"를 검색합니다. 동적 IP 주소를 사용하여(프라이빗 네트워크의 NAT 게이트웨이를 통해) 방화벽 뒤에서 연결하거나 ISP를 통해 연결하면 주소가 주기적으로 변경될 수 있습니다. 이 경우 클라이언트 컴퓨터에서 사용하는 IP 주소 범위를 알아야 합니다.

다음 예제에서는 사용자의 IP 주소를 사용하여 ID가 `sg-903004f8`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 SSH를 활성화하는 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 예제는 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
- AWS SDK 및 AWS CLI 코드 예제를 보고 기여하려면 GitHub에서 AWS 코드 예제 리포지토리를 참조하세요.

AWS CLI에서 Amazon EC2 인스턴스 시작, 나열 및 삭제

AWS Command Line Interface(AWS CLI)를 사용하여 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스를 시작, 나열 및 삭제할 수 있습니다. 시작하는 인스턴스가 AWS 프리 티어에 해당하지 않는 경우 인스턴스를 시작한 후에 요금이 청구되고 유휴 상태를 포함해 인스턴스가 실행된 시간에 대해 과금됩니다.

> **Note**
>
> 추가 명령 예제는 AWS CLI 참조 가이드를 참조하세요.

주제

사전 조건

이 주제의 `ec2` 명령을 실행하려면 다음을 수행해야 합니다.

- AWS CLI를 설치하고 구성합니다. 자세한 내용은 AWS CLI 설치 및 AWS CLI에 대한 인증 및 액세스 보안 인증(을)를 참조하세요.
- Amazon EC2 액세스를 허용하도록 IAM 권한을 설정합니다. Amazon EC2의 IAM 권한에 대한 자세한 내용은 Amazon EC2 사용 설명서의 Amazon EC2에 대한 IAM 정책을 참조하세요.
- 키 페어 및 보안 그룹을 생성합니다.
- Amazon Machine Image(AMI)를 선택하고 AMI ID를 기록합니다. 자세한 내용은 https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/finding-an-ami.htmlAmazon EC2 사용 설명서의 적합한 AMI 찾기를 참조하세요.

인스턴스 시작

선택한 AMI를 사용하여 Amazon EC2 인스턴스를 시작하려면 `aws ec2 run-instances` 명령을 사용합니다. Virtual Private Cloud(VPC)에서 인스턴스를 시작할 수 있습니다.

처음에는 인스턴스가 `pending` 상태로 표시되지만 몇 분 후에 `running` 상태로 변경됩니다.

다음 예제는 지정한 VPC 서브넷에서 `t2.micro` 인스턴스를 시작하는 방법을 보여줍니다. *####* 파라미터 값을 사용자의 값으로 바꾸세요.

```
$ 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 사용하면 실행될 때 인스턴스에 연결할 추가 Amazon Elastic Block Store(Amazon EBS) 볼륨 또는 인스턴스 스토어 볼륨을 지정할 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 `/dev/sdf`를 사용하여 인스턴스에 이를 매핑합니다.

```
--block-device-mappings "[{\"DeviceName\":\"/dev/sdf\",\"Ebs\":{\"VolumeSize\":20,
\"DeleteOnTermination\":false}}]"
```

다음 예제에서는 기존 스냅샷을 기반으로 /dev/sdf에 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 [블록 디바이스 매핑](#)을 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 [리소스에 태그 지정](#)을 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 EC2 사용 설명서의 [Amazon EC2 인스턴스에 연결](#)을 참조하세요.

인스턴스 나열

AWS CLI를 사용하여 인스턴스를 나열하고 정보를 확인할 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, ami-y0123456 및 ami-z0123456 AMI를 사용해 시작된 인스턴스를 나열합니다.

```
$ aws ec2 describe-instances --filters "Name=image-id,Values=ami-x0123456,ami-
y0123456,ami-z0123456"
```

인스턴스 삭제

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-instances](#)를 사용합니다. 자세한 내용은 Amazon EC2 사용 설명서의 [인스턴스 종료](#)를 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
- AWS SDK 및 AWS CLI 코드 예제를 보고 기여하려면 GitHub에서 AWS 코드 예제 리포지토리를 참조하세요.

AWS CLI에서 bash 스크립트로 Amazon EC2 인스턴스 유형 변경

Amazon EC2에 대한 이 bash 스크립팅 예제에서는 AWS Command Line Interface(AWS CLI)를 사용하여 Amazon EC2 인스턴스의 인스턴스 유형을 변경합니다. 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 가이드를 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 AWS CLI 설치 및 AWS CLI에 대한 인증 및 액세스 보안 인증(을)를 참조하세요.
- 사용하는 프로파일에는 예제에서 수행하는 AWS 작업을 허용하는 권한이 있어야 합니다.
- 중지 및 수정 권한이 있는 계정에서 실행 중인 Amazon EC2 인스턴스입니다. 테스트 스크립트를 실행하면 테스트 스크립트가 인스턴스를 시작하고 유형을 변경하여 인스턴스를 테스트한 다음 인스턴스를 종료합니다.
- AWS 모범 사례로서 이 코드에 최소 권한을 부여하거나 태스크를 수행하는 데 필요한 권한만 부여하세요. 자세한 내용은 AWS Identity and Access Management(IAM) 사용 설명서에서 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 발생할 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 GitHub에서 AWS 코드 예제 리포지토리의 Amazon EC2 인스턴스 유형 변경을 참조하세요.

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 있는지 확인합니다.
- `-f`를 선택하지 않으면 인스턴스를 중지하기 전에 사용자에게 경고합니다.
- 인스턴스 유형을 변경합니다.
- `-r`을 설정하면 인스턴스를 다시 시작하고 인스턴스가 실행 중인지 확인합니다.

GitHub에서 `change_ec2_instance_type.sh`에 대한 코드를 확인하세요.

`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 실행 시 통과/실패 상태를 표시합니다. 기본적으로 테스트는 자동으로 실행되며 출력에는 최종 전체 통과/실패 상태만 포함됩니다.

- -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. 프롬프트에서 Enter 키를 누르면 스크립트가 다음 단계로 진행됩니다.

GitHub에서 `test_change_ec2_instance_type.sh`에 대한 코드를 확인하세요.

awsdocs_general.sh

스크립트 파일 `awsdocs_general.sh`에는 AWS CLI에 대한 고급 예제에서 사용되는 범용 함수가 들어 있습니다.

GitHub에서 `awsdocs_general.sh`에 대한 코드를 확인하세요.

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
- AWS SDK 및 AWS CLI 코드 예제를 보고 기여하려면 GitHub에서 AWS 코드 예제 리포지토리를 참조하세요.

AWS CLI에서 Amazon S3 Glacier 사용

Amazon S3 Glacier 소개
Amazon S3 Glacier 소개

이 주제에서는 S3 Glacier에 대한 일반적인 태스크를 수행하는 AWS CLI 명령의 예제를 보여줍니다. 이 예제는 AWS CLI를 사용하여 대용량 파일을 분할하고 명령줄에서 업로드하여 S3 Glacier로 업로드 방법을 보여줍니다.

AWS Command Line Interface(AWS CLI)를 사용하여 Amazon S3 Glacier의 기능에 액세스할 수 있습니다. 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` 섹션을 참조하세요.

주제

- 사전 조건
- Amazon S3 Glacier 볼트 생성
- 파일 업로드 준비
- 멀티파트 업로드 및 파일 업로드 시작
- 업로드 완료
- 리소스

사전 조건

`glacier` 명령을 실행하려면 다음을 수행해야 합니다.

- AWS CLI를 설치하고 구성합니다. 자세한 내용은 AWS CLI 설치 및 AWS CLI에 대한 인증 및 액세스 보안 인증(을)를 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 3MiB의 임의 데이터를 포함하는 *`largefile`*이라는 파일을 생성합니다.

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/urandom`을 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 1MB(1,048,576바이트)의 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. 작업이 완료되면 AWS CLI에서 업로드 ID를 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 Glacier에서 업로드된 모든 조각이 AWS에 도달했는지 확인하려면 원본 파일의 트리 해시가 필요합니다.

트리 해시를 계산하려면 파일을 1MiB 파트로 분할하고 각 조각의 이진 SHA-256 해시를 계산합니다. 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 OpenSSL.org에서 Windows용으로 제공됩니다.

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 이진 SHA-256 해시를 계산 및 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

AWS CLI에서 IAM 사용

AWS Identity and Access Management 소개
AWS Identity and Access Management 소개

AWS Command Line Interface(AWS CLI)를 사용하여 AWS Identity and Access Management(IAM) 기능에 액세스할 수 있습니다. IAM에 대한 AWS CLI 명령을 나열하려면 다음 명령을 사용합니다.

```
aws iam help
```

이 주제에서는 IAM에 대한 일반적인 태스크를 수행하는 AWS CLI 명령의 예제를 보여줍니다.

명령을 실행하기 전에 기본 자격 증명을 설정합니다. 자세한 내용은 AWS CLI 설정 구성 단원을 참조하세요.

IAM 서비스에 대한 자세한 내용은 AWS Identity and Access Management 사용 설명서를 참조하세요.

주제

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 확인합니다. 다음 명령은 `list-policies`를 사용하여 이름이 `PowerUserAccess`인 정책의 ARN을 찾습니다. 그런 다음 해당 ARN을 환경 변수에 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 ARN을 보유하는 환경 변수를 참조합니다.

```
$ 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 리소스](액세스 관리 리소스)를 참조하세요. 이 주제에서는 권한 및 정책 개요에 대한 링크와 Amazon S3, Amazon EC2 및 기타 서비스에 액세스하기 위한 정책 예에 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
```

AWS CLI에서 Amazon S3 사용

Amazon Simple Storage Service(Amazon S3) 소개
Amazon Simple Storage Service(Amazon S3) 소개 - AWS의 클라우드 스토리지

AWS Command Line Interface(AWS CLI)를 사용하여 Amazon Simple Storage Service(Amazon S3)의 기능에 액세스할 수 있습니다. Amazon S3는 확장성과 내구성이 뛰어난 객체 스토리지 서비스입니다. Amazon S3는 사실상 무제한의 스토리지 용량을 제공하도록 설계되어 다양한 데이터 스토리지 및 관리 요구 사항을 충족하는 이상적인 솔루션입니다.

Amazon S3를 사용하면 작은 파일부터 대용량 데이터세트에 이르기까지 모든 종류의 데이터를 객체 형태로 저장하고 검색할 수 있습니다. 각 객체는 버킷이라는 컨테이너에 저장되며, AWS Management Console을 통해 또는 AWS SDK, 도구 및 AWS CLI를 통해 프로그래밍 방식으로 액세스하고 관리할 수 있습니다.

기본 스토리지를 비롯하여 Amazon S3는 수명 주기 관리, 버전 관리, 확장성 및 보안을 비롯한 다양한 기능을 제공합니다. 다른 AWS 서비스와 통합되어 필요에 따라 확장할 수 있는 클라우드 기반 솔루션을 구축할 수 있습니다.

AWS CLI는 Amazon S3에 액세스하기 위한 2가지 티어의 명령을 제공합니다.

- s3 - 객체 및 버킷 생성, 조작, 삭제, 동기화 등 일반적인 태스크 수행을 간소화하는 AWS CLI 전용 사용자 지정 상위 수준 명령입니다.
- s3api - 모든 Amazon S3 API 작업에 대한 직접 액세스를 제공하며, 이를 통해 고급 작업을 수행할 수 있습니다.

이 설명서의 주제는 다음과 같습니다.

- AWS CLI에서 상위 수준(s3) 명령 사용
- AWS CLI에서 API 수준(s3api) 명령 사용
- AWS CLI의 Amazon S3 버킷 수명 주기에 대한 스크립팅 예제

AWS CLI에서 상위 수준(s3) 명령 사용

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 AWS CLI 설치 및 AWS CLI에 대한 인증 및 액세스 보안 인증(을)를 참조하세요.
- 사용하는 프로파일에는 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 CLI에서 자동으로 멀티파트 업로드를 수행합니다. 이러한 `aws s3` 명령을 사용할 때는 실패한 업로드를 재개할 수 없습니다.

시간 초과로 인해 멀티파트 업로드가 실패하거나 AWS CLI에서 수동으로 취소할 경우 AWS CLI는 업로드를 중지하고 생성된 모든 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` 네임스페이스에서 AWS CLI 버전 1 버전의 명령을 사용하여 한 Amazon S3 버킷 위치에서 다른 Amazon S3 버킷 위치로 파일을 복사하고 해당 작업에서 멀티파트 복사를 사용하는 경우 소스 객체의 파일 속성은 대상 객체로 복사되지 않습니다.

버킷 만들기

`s3 mb` 명령을 사용하여 버킷을 만듭니다. 버킷 이름은 글로벌로 고유(모든 Amazon S3에서 고유)해야 하며 DNS를 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-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 *bucket-name/example/*에 있는 객체(즉, 접두사 *example/*을 기준으로 필터링된 *bucket-name*에 있는 객체)를 나열합니다.

```
$ aws s3 ls s3://amzn-s3-demo-bucket/example/
2018-12-06 18:59:32          3 MyFile1.txt
```

특정 리전의 버킷과 객체만 표시하려면 `--region` 옵션을 사용합니다.

```
$ aws s3 ls --region us-east-2
2018-12-06 18:59:32          3 MyFile1.txt
```

버킷 및 객체 목록이 많은 경우 `--max-items` 또는 `--page-size` 옵션을 사용하여 결과를 페이지 매김할 수 있습니다. `--max-items` 옵션은 직접 호출에서 반환되는 총 버킷 및 객체 수를 제한하고 `--page-size` 옵션은 페이지에 나열된 버킷 및 객체 수를 제한합니다.

```
$ aws s3 ls --max-items 100 --page-size 10
```

페이지 매김에 대한 자세한 내용은 the section called “--page-size” 및 the section called “--max-items” 섹션을 참조하세요.

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.txt에서 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 --recursive에서 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 대상에 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 섹션을 참조하세요.

> **Warning**
>
> Amazon S3 소스 또는 대상 URI에서 어떤 유형의 액세스 포인트 ARN 또는 액세스 포인트 별칭을 사용하는 경우, 소스 및 대상 Amazon S3 URI가 서로 다른 기본 버킷으로 확인되도록 각별히 주의해야 합니다. 소스 버킷과 대상 버킷이 동일한 경우 소스 파일이나 객체를 그 자체로 옮길 수 있어 실수로 소스 파일이나 객체가 삭제될 수 있습니다. 소스 버킷과 대상 버킷이 동일하지 않은지 확인하려면 --validate-same-s3-paths 파라미터를 사용하거나 환경 변수 AWS_CLI_S3_MV_VALIDATE_SAME_S3_PATHS를 true로 설정하세요.

s3 mv 예제

다음 예제에서는 s3://amzn-s3-demo-bucket/example에서 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**
>
> PowerShell을 사용하는 경우 셸은 CRLF의 인코딩을 변경하거나, 파이프 입력이나 출력 또는 리디렉션된 출력에 CRLF를 추가할 수 있습니다.

s3 cp 명령은 다음 구문을 사용하여 stdin에서 지정된 버킷으로 파일 스트림을 업로드합니다.

구문

```
$ aws s3 cp - <target> [--options]
```

s3 cp 명령은 다음 구문을 사용하여 stdout에 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 섹션을 참조하세요.

s3 cp 예제

다음 예제에서는 s3://amzn-s3-demo-bucket/example에서 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](https://docs.aws.amazon.com/cli/latest/reference/s3/sync.html) 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` 섹션을 참조하세요.

s3 동기화 예제

다음 예제에서는 amzn-s3-demo-bucket이라는 버킷에 있는 path라는 Amazon S3 접두사의 내용을 현재 작업 디렉터리와 동기화합니다.

`s3 sync`는 대상에 있는 동일한 이름의 파일과 크기 또는 수정 시간이 다른 모든 파일을 업데이트합니다. 출력에는 동기화 중에 수행된 특정 작업이 표시됩니다. 이 작업은 하위 디렉터리 `MySubdirectory`와 해당 내용을 `s3://amzn-s3-demo-bucket/path/MySubdirectory`와 반복적으로 동기화합니다.

```
$ aws s3 sync . s3://amzn-s3-demo-bucket/path
upload: MySubdirectory\MyFile3.txt to s3://amzn-s3-demo-bucket/path/MySubdirectory/
MyFile3.txt
upload: MyFile2.txt to s3://amzn-s3-demo-bucket/path/MyFile2.txt
upload: MyFile1.txt to s3://amzn-s3-demo-bucket/path/MyFile1.txt
```

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 미리 제공된 ACL을 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_ID`를 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 `full`로 설정할 수 있습니다.
- *`Grantee_Type`* - 피부여자 식별 방법을 지정합니다. `uri`, `emailaddress` 또는 `id`로 설정할 수 있습니다.
- *`Grantee_ID`* - *`Grantee_Type`*을 기준으로 피부여자를 지정합니다.
  - `uri` – 그룹의 URI입니다. 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 사용 설명서의 Amazon S3 버킷 작업
- Amazon S3 사용 설명서의 접두사 및 구분 기호를 사용하여 계층적 구조로 키 나열
- Amazon S3 사용 설명서의 AWS SDK for .NET(낮은 수준)를 사용하여 S3 버킷에 대한 멀티파트 업로드 중단

AWS CLI에서 API 수준(s3api) 명령 사용

API 수준 명령(`s3api` 명령 세트에 포함됨)을 사용하면 Amazon Simple Storage Service(Amazon S3) API에 직접 액세스할 수 있으며 상위 수준 `s3` 명령에 표시되지 않는 일부 작업을 활성화할 수 있습니다. 이러한 명령은 서비스의 기능에 대한 API 수준 액세스를 제공하는 다른 AWS 서비스와 동등합니다. `s3` 명령어에 대한 자세한 내용은 AWS CLI에서 상위 수준(s3) 명령 사용 섹션을 참조하세요.

이 주제에서는 Amazon S3 API에 매핑되는 하위 수준 명령을 사용하는 방법을 보여주는 예제를 제공합니다. 또한 각 S3 API 명령에 대한 예제는 AWS CLI 참조 가이드의 `s3api` 섹션에서 찾을 수 있습니다.

주제

- 사전 조건
- 사용자 지정 ACL 적용
- 로깅 정책 구성
- 리소스

사전 조건

`s3api` 명령을 실행하려면 다음을 수행해야 합니다.

- AWS CLI를 설치하고 구성합니다. 자세한 내용은 AWS CLI 설치 및 AWS CLI에 대한 인증 및 액세스 보안 인증(을)를 참조하세요.
- 사용하는 프로파일에는 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 ACL 적용

고급 명령을 사용하면 `--acl` 옵션을 사용하여 Amazon S3 객체에 미리 정의된 액세스 제어 목록(ACL)을 적용할 수 있습니다. 그러나 이 명령을 사용해도 버킷 전체 ACL을 설정할 수는 없습니다. 그러나 put-bucket-acl API 수준 명령을 사용하여 이 작업을 수행할 수 있습니다.

다음 예제에서는 두 명의 AWS 사용자(user1@example.com 및 user2@example.com)에게 전체 제어 권한을 부여하고 모든 사람에게 읽기 권한을 부여하는 방법을 보여줍니다. "모든 사람"의 식별자는 파라미터로 전달하는 특수 URI에서 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"'
```

ACL을 구성하는 방법에 대한 자세한 내용은 Amazon Simple Storage Service API 참조에서 PUT Bucket acl을 참조하세요. CLI에서 `s3api`와 같은 `put-bucket-acl` ACL 명령은 동일한 간편 인수 표기법을 사용합니다.

로깅 정책 구성

API 명령인 `put-bucket-logging`은 버킷 로깅 정책을 구성합니다.

다음 예제에서 AWS 사용자 user@example.com에게는 로그 파일을 완전히 제어하는 권한이 부여되며, 모든 사용자는 읽기 액세스 권한을 갖게 됩니다. 로그를 읽고 버킷에 쓰는 데 필요한 권한을 Amazon S3 로그 전달 시스템(URI로 지정)에 부여하려는 경우에도 `put-bucket-acl` 명령이 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 사용 설명서의 Amazon S3 버킷 작업
- Amazon S3 사용 설명서의 접두사 및 구분 기호를 사용하여 계층적 구조로 키 나열
- Amazon S3 사용 설명서의 AWS SDK for .NET(낮은 수준)를 사용하여 S3 버킷에 대한 멀티파트 업로드 중단

AWS CLI의 Amazon S3 버킷 수명 주기에 대한 스크립팅 예제

이 주제에서는AWS 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 AWS CLI 설치 및 AWS CLI에 대한 인증 및 액세스 보안 인증(을)를 참조하세요.
- 사용하는 프로파일에는 예제에서 수행하는 AWS 작업을 허용하는 권한이 있어야 합니다.

- AWS 모범 사례로서 이 코드에 최소 권한을 부여하거나 태스크를 수행하는 데 필요한 권한만 부여하세요. 자세한 내용은 IAM 사용 설명서에서 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 발생할 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 GitHub에서 AWS 코드 예제 리포지토리의 Amazon S3 버킷 수명 주기 작업을 참조하세요.

파일

예제에는 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

GitHub에서 `bucket-operations.sh`에 대한 코드를 확인하세요.

test-bucket-operations.sh

셸 스크립트 파일 `test-bucket-operations.sh`는 `bucket-operations.sh` 파일을 소싱하고 각 함수를 호출하여 함수를 호출하는 방법을 보여줍니다. 함수를 호출한 후 테스트 스크립트는 생성한 모든 리소스를 제거합니다.

GitHub에서 `test-bucket-operations.sh`에 대한 코드를 확인하세요.

awsdocs-general.sh

스크립트 파일 `awsdocs-general.sh`에는 AWS CLI에 대한 고급 코드 예제에서 사용되는 범용 함수가 들어 있습니다.

GitHub에서 `awsdocs-general.sh`에 대한 코드를 확인하세요.

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 사용 설명서의 Amazon S3 버킷 작업
- AWS SDK 및 AWS CLI 코드 예제를 보고 기여하려면 GitHub에서 AWS 코드 예제 리포지토리를 참조하세요.

AWS CLI에서 Amazon SNS 액세스

AWS Command Line Interface(AWS CLI)를 사용하여 Amazon Simple Notification Service(Amazon SNS)의 기능에 액세스할 수 있습니다. Amazon SNS에 대한 AWS CLI 명령을 나열하려면 다음 명령을 사용합니다.

```
aws sns help
```

명령을 실행하기 전에 기본 자격 증명을 설정합니다. 자세한 내용은 AWS CLI 설정 구성 단원을 참조하세요.

이 주제에서는 Amazon SNS에 대한 일반적인 태스크를 수행하는 AWS CLI 명령의 예제를 보여줍니다.

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 `email`에 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"
}
```

AWS는 `subscribe` 명령으로 지정한 이메일 주소로 즉시 확인 메시지를 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 예에서 AWS는 'Hello World!' 텍스트가 포함된 이메일 메시지를 `saanvi@example.com`으로 전송합니다.

주제에서 구독 취소

주제 구독을 해지하고 해당 주제로 게시된 메시지 수신을 중단하려면 [sns unsubscribe](#) 명령을 사용하고 구독을 해지할 주제의 ARN을 지정합니다.

```
$ aws sns unsubscribe --subscription-arn arn:aws:sns:us-west-2:123456789012:my-
topic:1328f057-de93-4c15-512e-8bb22EXAMPLE
```

구독이 성공적으로 해지되었는지 확인하려면 [sns list-subscriptions](#) 명령을 사용하여 해당 ARN이 목록에 더 이상 나타나지 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
```

AWS가 주제를 성공적으로 삭제했는지 확인하려면 `sns list-topics` 명령을 사용하여 해당 주제가 목록에 더 이상 나타나지 않는지 확인합니다.

```
$ aws sns list-topics
```

AWS CLI 명령 예제

이 항목의 코드 예제에서는 AWS와 함께 AWS Command Line Interface를 사용하는 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.

시나리오는 동일한 서비스 내에서 또는 다른 AWS 서비스와 결합된 상태에서 여러 함수를 호출하여 특정 태스크를 수행하는 방법을 보여주는 코드 예제입니다.

일부 서비스에는 서비스와 관련된 라이브러리 또는 함수를 활용하는 방법을 보여주는 추가 예제 범주가 포함되어 있습니다.

서비스

- AWS CLI를 사용하는 ACM 예제
- AWS CLI를 사용하는 API Gateway 예제
- AWS CLI를 사용하는 API Gateway HTTP 및 WebSocket API 예제
- AWS CLI를 사용하는 API Gateway Management API 예제
- AWS CLI를 사용하는 App Mesh 예제
- AWS CLI를 사용하는 App Runner 예제
- AWS CLI를 사용한 AWS AppConfig 예시
- AWS CLI를 사용하는 Application Auto Scaling 예제
- AWS CLI를 사용하는 Application Discovery Service 예제
- AWS CLI를 사용한 AppRegistry 예시

- AWS CLI를 사용하는 Athena 예제
- AWS CLI를 사용하는 Auto Scaling 예제
- AWS CLI를 사용하는 Auto Scaling Plans 예제
- AWS CLI를 사용한 AWS Backup 예시
- AWS CLI를 사용한 AWS Batch 예시
- AWS CLI를 사용한 AWS Budgets 예시
- AWS CLI를 사용한 Amazon Chime 예시
- AWS CLI를 사용한 Cloud Control API 예제
- AWS CLI를 사용한 AWS Cloud Map 예시
- AWS CLI를 사용한 AWS Cloud9 예시
- AWS CLI를 사용한 AWS CloudFormation 예시
- AWS CLI를 사용한 CloudFront 예시
- AWS CLI를 사용한 Amazon CloudSearch 예제
- CloudTrail examples using AWS CLI
- AWS CLI를 사용하는 CloudWatch 예제
- AWS CLI를 사용하는 CloudWatch Logs 예제
- AWS CLI를 사용한 CloudWatch Network Monitoring 예시
- AWS CLI를 사용한 CloudWatch Observability Access Monitor 예제
- AWS CLI를 사용한 CloudWatch Observability Admin 예제
- AWS CLI를 사용한 CloudWatch Synthetics 예제
- AWS CLI를 사용한 CodeArtifact 예제
- AWS CLI를 사용한 CodeBuild 예시
- AWS CLI를 사용한 CodeCommit 예시
- AWS CLI를 사용한 CodeDeploy 예시
- CodeGuru Reviewer examples using AWS CLI
- AWS CLI를 사용한 CodePipeline 예제
- AWS CodeStar를 사용한 Notifications 예제AWS CLI
- AWS CLI를 사용한 CodeConnections 예제
- AWS CLI를 사용한 Amazon Cognito ID 예제

- AWS CLI를 사용한 AWS IoT Greengrass 예시
- AWS CLI를 사용한 AWS IoT Greengrass V2 예시
- AWS CLI를 사용한 AWS IoT Jobs SDK release 예시
- AWS CLI를 사용한 AWS IoT SiteWise 예시
- AWS CLI를 사용한 AWS IoT Things Graph 예시
- AWS CLI를 사용한 AWS IoT 무선 예시
- AWS CLI를 사용한 Amazon IVS 예시
- AWS CLI를 사용한 Amazon IVS Chat 예시
- AWS CLI를 사용한 Amazon IVS Real-Time Streaming 예시
- AWS CLI를 사용한 Amazon Kendra 예시
- AWS CLI를 사용한 Kinesis 예시
- AWS CLI를 사용한 AWS KMS 예시
- AWS CLI를 사용한 Lake Formation 예시
- AWS CLI를 사용한 Lambda 예시
- AWS CLI를 사용한 License Manager 예시
- AWS CLI를 사용한 Lightsail 예시
- AWS CLI를 사용한 Macie 예시
- AWS CLI를 사용한 Amazon Managed Grafana 예제
- AWS CLI를 사용한 MediaConnect 예시
- AWS CLI를 사용한 MediaConvert 예시
- AWS CLI를 사용한 MediaLive 예시
- AWS CLI를 사용한 MediaPackage 예시
- AWS CLI를 사용한 MediaPackage VOD 예시
- AWS CLI를 사용한 MediaStore 데이터 플레인 예시
- AWS CLI를 사용한 MediaTailor 예시
- AWS CLI를 사용한 MemoryDB 예시
- AWS CLI를 사용한 Amazon MSK 예시
- AWS CLI를 사용한 Network Flow Monitor 예제
- AWS CLI를 사용한 Network Manager 예시
- AWS CLI를 사용한 OpenSearch 서비스 예제

- AWS CLI를 사용한 OpsWorks 예시
- AWS CLI를 사용한 AWS OpsWorks CM 예시
- AWS CLI를 사용한 Organizations 예시
- AWS CLI를 사용한 AWS Outposts 예시
- AWS CLI를 사용한 AWS Payment Cryptography 예시
- AWS CLI를 사용한 AWS Payment Cryptography 데이터 플레인 예시
- AWS CLI를 사용한 Amazon Pinpoint 예시
- AWS CLI를 사용한 Amazon Polly 예시
- AWS CLI를 사용한 AWS 가격표 예시
- AWS CLI를 사용한 AWS Private CA 예시
- AWS CLI를 사용한 AWS Proton 예시
- AWS CLI를 사용한 QLDB 예시
- AWS CLI를 사용한 Amazon RDS 예시
- AWS CLI를 사용한 Amazon RDS 예시
- AWS CLI를 사용한 Amazon RDS Performance Insights 예시
- AWS CLI를 사용한 Amazon Redshift 예시
- AWS CLI를 사용한 Amazon Rekognition 예시
- AWS CLI를 사용한 AWS RAM 예시
- AWS CLI를 사용한 Resource Explorer 예시
- AWS CLI를 사용한 Resource Groups 예시
- AWS CLI를 사용한 Resource Groups Tagging API 예시
- AWS CLI를 사용한 AWS RoboMaker 예시
- AWS CLI를 사용한 Route 53 예시
- AWS CLI를 사용한 Route 53 도메인 등록 예시
- AWS CLI를 사용한 Route 53 Profiles 예시
- AWS CLI를 사용하여 Route 53 Resolver 예시
- AWS CLI를 사용한 Amazon S3 예시
- AWS CLI를 사용한 Amazon S3 Control 예시
- AWS CLI를 사용한 S3 Glacier 예제
- AWS CLI를 사용한 Secrets Manager 예시

- AWS CLI를 사용한 Security Hub 예시
- AWS CLI를 사용한 Security Lake 예시
- AWS CLI를 사용한 AWS Serverless Application Repository 예시
- AWS CLI를 사용한 Service Catalog 예시
- AWS CLI를 사용한 Service Quotas 예시
- AWS CLI를 사용한 Amazon SES 예시
- AWS CLI를 사용한 Shield 예시
- AWS CLI를 사용한 Signer 예시
- AWS CLI를 사용한 Snowball Edge 예시
- AWS CLI를 사용한 Amazon SNS 예제
- AWS CLI를 사용한 Amazon SQS 예시
- AWS CLI를 사용한 Storage Gateway 예시
- AWS CLI를 사용한 AWS STS 예시
- AWS CLI를 사용한 지원 예시
- AWS CLI를 사용한 Amazon SWF 예시
- AWS CLI를 사용한 Systems Manager 예시
- AWS CLI를 사용한 Amazon Textract 예시
- AWS CLI를 사용한 Amazon Transcribe 예시
- AWS CLI를 사용한 Amazon Translate 예시
- AWS CLI를 사용한 Trusted Advisor 예시
- AWS CLI를 사용하여 Verified Permissions 예시
- AWS CLI를 사용한 VPC Lattice 예시
- AWS CLI를 사용한 AWS WAF Classic 예시
- AWS CLI를 사용한 AWS WAF Classic Regional 예시
- AWS CLI를 사용한 AWS WAFV2 예시
- AWS CLI를 사용한 WorkDocs 예제
- AWS CLI를 사용한 Amazon WorkMail 예시
- AWS CLI를 사용한 Amazon WorkMail Message Flow 예제
- AWS CLI를 사용한 WorkSpaces 예제
- AWS CLI를 사용한 X-Ray 예제

AWS CLI를 사용하는 ACM 예제

다음 코드 예제에서는 ACM에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여 줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

작업

add-tags-to-certificate

다음 코드 예시는 `add-tags-to-certificate`의 사용 방법을 보여 줍니다.

AWS CLI

기존 ACM 인증서에 태그 추가

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
```

- API 세부 정보는 AWS CLI 명령 참조의 AddTagsToCertificate를 참조하세요.

delete-certificate

다음 코드 예시는 `delete-certificate`의 사용 방법을 보여 줍니다.

AWS CLI

계정에서 ACM 인증서를 삭제하는 방법

다음 `delete-certificate` 명령은 지정된 ARN이 포함된 인증서를 삭제합니다.

```
aws acm delete-certificate --certificate-
arn arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012
```

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteCertificate](#)를 참조하세요.

`describe-certificate`

다음 코드 예시는 `describe-certificate`의 사용 방법을 보여 줍니다.

AWS CLI

ACM 인증서에 포함된 필드를 검색하는 방법

다음 `describe-certificate` 명령은 지정된 ARN이 포함된 인증서의 모든 필드를 검색합니다.

```
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"
    ]
  }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeCertificate를 참조하세요.

export-certificate

다음 코드 예시는 `export-certificate`의 사용 방법을 보여 줍니다.

AWS CLI

프라이빗 CA에서 발급한 프라이빗 인증서를 내보내는 방법

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
```

- API 세부 정보는 AWS CLI 명령 참조의 ExportCertificate를 참조하세요.

get-certificate

다음 코드 예시는 get-certificate의 사용 방법을 보여 줍니다.

AWS CLI

ACM 인증서를 검색하는 방법

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-----"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetCertificate를 참조하세요.

`import-certificate`

다음 코드 예시는 `import-certificate`의 사용 방법을 보여 줍니다.

AWS CLI

인증서를 ACM으로 가져오는 방법

다음 `import-certificate` 명령은 인증서를 ACM으로 가져옵니다. 실제 파일 이름으로 바꾸세요.

```
aws acm import-certificate --certificate file://Certificate.pem --certificate-chain file://CertificateChain.pem --private-key file://PrivateKey.pem
```

- API 세부 정보는 AWS CLI 명령 참조의 ImportCertificate를 참조하세요.

`list-certificates`

다음 코드 예시는 `list-certificates`의 사용 방법을 보여 줍니다.

AWS CLI

AWS 계정의 ACM 인증서를 나열하는 방법

다음 `list-certificates` 명령은 계정에 있는 인증서의 ARN을 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-certificates`를 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 인증서 ARN과 `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 PENDING_VALIDATION 상태인 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
```

- API 세부 정보는 AWS CLI 명령 참조의 ListCertificates를 참조하세요.

list-tags-for-certificate

다음 코드 예시는 list-tags-for-certificate의 사용 방법을 보여 줍니다.

AWS CLI

ACM 인증서에 적용된 태그를 나열하는 방법

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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [ListTagsForCertificate](https://docs.aws.amazon.com/goto/aws-cli/acm-2015-12-08/ListTagsForCertificate)를 참조하세요.

`remove-tags-from-certificate`

다음 코드 예시는 `remove-tags-from-certificate`의 사용 방법을 보여 줍니다.

AWS CLI

ACM 인증서에서 태그를 제거하는 방법

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
```

- API 세부 정보는 AWS CLI 명령 참조의 [RemoveTagsFromCertificate](https://docs.aws.amazon.com/goto/aws-cli/acm-2015-12-08/RemoveTagsFromCertificate)를 참조하세요.

`request-certificate`

다음 코드 예시는 `request-certificate`의 사용 방법을 보여 줍니다.

AWS CLI

새 ACM 인증서를 요청하는 방법

다음 `request-certificate` 명령은 DNS 검증을 사용하여 www.example.com 도메인의 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
```

- API 세부 정보는 AWS CLI 명령 참조의 [RequestCertificate](#)를 참조하세요.

resend-validation-email

다음 코드 예시는 resend-validation-email의 사용 방법을 보여 줍니다.

AWS CLI

ACM 인증서 요청에 대한 검증 이메일을 재전송하는 방법

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
```

- API 세부 정보는 AWS CLI 명령 참조의 ResendValidationEmail을 참조하세요.

update-certificate-options

다음 코드 예시는 update-certificate-options의 사용 방법을 보여 줍니다.

AWS CLI

인증서 옵션을 업데이트하는 방법

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
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateCertificateOptions를 참조하세요.

AWS CLI를 사용하는 API Gateway 예제

다음 코드 예제는 API Gateway와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

create-api-key

다음 코드 예시는 create-api-key의 사용 방법을 보여 줍니다.

AWS CLI

기존 API 및 스테이지에 대해 활성화된 API 키를 생성하는 방법

명령:

```
aws apigateway create-api-key --name 'Dev API Key' --description 'Used for
 development' --enabled --stage-keys restApiId='a1b2c3d4e5',stageName='dev'
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateApiKey를 참조하세요.

create-authorizer

다음 코드 예시는 create-authorizer의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: API에 대해 토큰 기반 API Gateway 사용자 지정 권한 부여자를 생성하는 방법

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: API에 대해 Cognito 사용자 풀 기반 API Gateway 사용자 지정 권한 부여자를 생성하는 방법

다음 `create-authorizer` 예제에서는 Cognito 사용자 풀 기반 API Gateway 사용자 지정 권한 부여자를 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: API에 대해 요청 기반 API Gateway 사용자 지정 권한 부여자를 생성하는 방법

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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateAuthorizer를 참조하세요.

create-base-path-mapping

다음 코드 예시는 `create-base-path-mapping`의 사용 방법을 보여 줍니다.

AWS CLI

사용자 지정 도메인 이름의 기본 경로 매핑을 생성하는 방법

명령:

```
aws apigateway create-base-path-mapping --domain-name subdomain.domain.tld --rest-
api-id 1234123412 --stage prod --base-path v1
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateBasePathMapping을 참조하세요.

`create-deployment`

다음 코드 예시는 `create-deployment`의 사용 방법을 보여 줍니다.

AWS CLI

API용으로 구성된 리소스를 새 스테이지에 배포하는 방법

명령:

```
aws apigateway create-deployment --rest-api-id 1234123412 --stage-name dev --stage-description 'Development Stage' --description 'First deployment to the dev stage'
```

API용으로 구성된 리소스를 기존 스테이지에 배포하는 방법

명령:

```
aws apigateway create-deployment --rest-api-id 1234123412 --stage-name dev --description 'Second deployment to the dev stage'
```

API용으로 구성된 리소스를 기존 스테이지에 스테이지 변수와 함께 배포하는 방법

aws apigateway create-deployment --rest-api-id 1,234,123,412 --stage-name dev --description 'Third deployment to the dev stage' --variables key='value',otherKey='otherValue'

- API 세부 정보는 AWS CLI 명령 참조의 CreateDeployment를 참조하세요.

`create-domain-name-access-association`

다음 코드 예시는 `create-domain-name-access-association`의 사용 방법을 보여 줍니다.

AWS CLI

도메인 이름 액세스 연결을 생성하려면

다음 `create-domain-name-access-association` 예제에서는 프라이빗 사용자 지정 도메인 이름과 VPC 엔드포인트 간에 도메인 이름 액세스 연결을 생성합니다.

```
aws apigateway create-domain-name-access-association \
    --domain-name-arn arn:aws:apigateway:us-west-2:111122223333:/domainnames/my.private.domain.tld+abcd1234 \
    --access-association-source vpce-abcd1234efg \
    --access-association-source-type VPCE
```

출력:

```
{
    "domainNameAccessAssociationArn": "arn:aws:apigateway:us-west-2:012345678910:/
domainnameaccessassociations/domainname/my.private.domain.tld/vpcesource/vpce-
abcd1234efg
    "accessAssociationSource": "vpce-abcd1234efg",
    "accessAssociationSourceType": "VPCE",
    "domainNameArn" : "arn:aws:apigateway:us-west-2:111122223333:/domainnames/
private.example.com+abcd1234"
}
```

자세한 내용은 Amazon API Gateway 개발자 안내서의 API Gateway에서 프라이빗 API에 대한 사용자 지정 도메인 이름을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateDomainNameAccessAssociation 섹션을 참조하세요.

`create-domain-name`

다음 코드 예시는 `create-domain-name`의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 퍼블릭 사용자 지정 도메인 이름 생성

다음 `create-domain-name` 예제에서는 퍼블릭 사용자 지정 도메인 이름을 생성합니다.

```
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'
```

출력:

```
{
    "domainName": "my.domain.tld",
    "certificateName": "my.domain.tld cert",
    "certificateArn": "arn:aws:acm:us-east-1:012345678910:certificate/fb1b9770-
a305-495d-aefb-27e5e101ff3",
```

```
    "certificateUploadDate": "2024-10-08T11:29:49-07:00",
    "distributionDomainName": "abcd1234.cloudfront.net",
    "distributionHostedZoneId": "Z2FDTNDATAQYW2",
    "endpointConfiguration": {
        "types": [
            "EDGE"
        ]
    },
    "domainNameStatus": "AVAILABLE",
    "securityPolicy": "TLS_1_2"
}
```

자세한 내용은 Amazon API Gateway 개발자 안내서의 API Gateway에서 퍼블릭 REST API에 대한 사용자 지정 도메인 이름을 참조하세요.

예제 2: 프라이빗 사용자 지정 도메인 이름 생성

다음 `create-domain-name` 예제에서는 프라이빗 사용자 지정 도메인 이름을 생성합니다.

```
aws apigateway create-domain-name \
    --domain-name 'my.private.domain.tld' \
    --certificate-name 'my.domain.tld cert' \
    --certificate-arn 'arn:aws:acm:us-east-1:012345678910:certificate/fb1b9770-a305-495d-aefb-27e5e101ff3' \
    --endpoint-configuration '{"types": ["PRIVATE"]}' \
    --security-policy 'TLS_1_2' \
    --policy file://policy.json
```

`policy.json`의 콘텐츠:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": "*",
            "Action": "execute-api:Invoke",
            "Resource": [
                "execute-api:/*"
            ]
        },
        {
```

```
                "Effect": "Deny",
                "Principal": "*",
                "Action": "execute-api:Invoke",
                "Resource": [
                    "execute-api:/*"
                ],
                "Condition" : {
                    "StringNotEquals": {
                        "aws:SourceVpce": "vpce-abcd1234efg"
                    }
                }
            }
        ]
    }
```

출력:

```
{
    "domainName": "my.private.domain.tld",
    "domainNameId": "abcd1234",
    "domainNameArn": "arn:aws:apigateway:us-east-1:012345678910:/domainnames/
my.private.domain.tld+abcd1234",
    "certificateArn": "arn:aws:acm:us-east-1:012345678910:certificate/fb1b9770-
a305-495d-aefb-27e5e101ff3",
    "certificateUploadDate": "2024-09-10T10:31:20-07:00",
    "endpointConfiguration": {
        "types": [
            "PRIVATE"
        ]
    },
    "domainNameStatus": "AVAILABLE",
    "securityPolicy": "TLS_1_2",
    "policy": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":
\"Allow\",\"Principal\":\"*\",\"Action\":\"execute-api:Invoke\",\"Resource\":
\"arn:aws:execute-api:us-east-1:012345678910:/domainnames/my.private.domain.tld
+abcd1234\"},{\"Effect\":\"Deny\",\"Principal\":\"*\",\"Action\":\"execute-
api:Invoke\",\"Resource\":\"arn:aws:execute-api:us-east-1:012345678910:/domainnames/
my.private.domain.tld+abcd1234\",\"Condition\":{\"StringNotEquals\":{\"aws:SourceVpc
\":\"vpc-1a2b3c4d\"}}}]}"
}
```

자세한 내용은 Amazon API Gateway 개발자 안내서의 [API Gateway에서 퍼블릭 REST API에 대한 사용자 지정 도메인 이름](#)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateDomainName을 참조하세요.

create-model

다음 코드 예시는 create-model의 사용 방법을 보여 줍니다.

AWS CLI

API에 대한 모델을 생성하는 방법

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
\" } } } } }"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateModel을 참조하세요.

create-resource

다음 코드 예시는 create-resource의 사용 방법을 보여 줍니다.

AWS CLI

API에서 리소스를 생성하는 방법

명령:

```
aws apigateway create-resource --rest-api-id 1234123412 --parent-id a1b2c3 --path-part 'new-resource'
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateResource를 참조하세요.

create-rest-api

다음 코드 예시는 `create-rest-api`의 사용 방법을 보여 줍니다.

AWS CLI

API를 생성하는 방법

명령:

```
aws apigateway create-rest-api --name 'My First API' --description 'This is my first API'
```

기존 API에서 복제 API를 생성하는 방법

명령:

```
aws apigateway create-rest-api --name 'Copy of My First API' --description 'This is a copy of my first API' --clone-from 1234123412
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateRestApi를 참조하세요.

create-stage

다음 코드 예시는 `create-stage`의 사용 방법을 보여 줍니다.

AWS CLI

API에서 기존 배포를 포함할 스테이지를 생성하는 방법

명령:

```
aws apigateway create-stage --rest-api-id 1234123412 --stage-name 'dev' --description 'Development stage' --deployment-id a1b2c3
```

API에서 기존 배포 및 사용자 지정 스테이지 변수를 포함할 스테이지를 생성하는 방법

명령:

```
aws apigateway create-stage --rest-api-id 1234123412 --stage-name 'dev'
 --description 'Development stage' --deployment-id a1b2c3 --variables
 key='value',otherKey='otherValue'
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateStage를 참조하세요.

`create-usage-plan-key`

다음 코드 예시는 `create-usage-plan-key`의 사용 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
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateUsagePlanKey를 참조하세요.

`create-usage-plan`

다음 코드 예시는 `create-usage-plan`의 사용 방법을 보여 줍니다.

AWS CLI

월초에 재설정되는 스로틀 및 할당량 제한이 있는 사용 계획을 생성하는 방법

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
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateUsagePlan을 참조하세요.

delete-api-key

다음 코드 예시는 `delete-api-key`의 사용 방법을 보여 줍니다.

AWS CLI

API 키를 삭제하는 방법

명령:

```
aws apigateway delete-api-key --api-key 8bklk8bl1k3sB38D9B3l0enyWT8c09B30lkq0blk
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteApiKey를 참조하세요.

delete-authorizer

다음 코드 예시는 `delete-authorizer`의 사용 방법을 보여 줍니다.

AWS CLI

API에서 사용자 지정 권한 부여자를 삭제하는 방법

명령:

```
aws apigateway delete-authorizer --rest-api-id 1234123412 --authorizer-id 7gkfbo
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteAuthorizer를 참조하세요.

delete-base-path-mapping

다음 코드 예시는 `delete-base-path-mapping`의 사용 방법을 보여 줍니다.

AWS CLI

사용자 지정 도메인 이름의 기본 경로 매핑을 삭제하는 방법

명령:

```
aws apigateway delete-base-path-mapping --domain-name 'api.domain.tld' --base-path
 'dev'
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteBasePathMapping을 참조하세요.

`delete-client-certificate`

다음 코드 예시는 `delete-client-certificate`의 사용 방법을 보여 줍니다.

AWS CLI

클라이언트 인증서를 삭제하는 방법

명령:

```
aws apigateway delete-client-certificate --client-certificate-id a1b2c3
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteClientCertificate를 참조하세요.

`delete-deployment`

다음 코드 예시는 `delete-deployment`의 사용 방법을 보여 줍니다.

AWS CLI

API에서 배포를 삭제하는 방법

명령:

```
aws apigateway delete-deployment --rest-api-id 1234123412 --deployment-id a1b2c3
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteDeployment를 참조하세요.

`delete-domain-name-access-association`

다음 코드 예시는 `delete-domain-name-access-association`의 사용 방법을 보여 줍니다.

AWS CLI

도메인 이름 액세스 연결을 삭제하려면

다음 `delete-domain-name-access-association` 예제에서는 프라이빗 사용자 지정 도메인 이름과 VPC 엔드포인트 간의 도메인 이름 액세스 연결을 삭제합니다.

```
aws apigateway delete-domain-name-access-association \
    --domain-name-access-association-arn arn:aws:apigateway:us-west-2:012345678910:/
domainnameaccessassociations/domainname/my.private.domain.tld/vpcesource/vpce-
abcd1234efg
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon API Gateway 개발자 안내서의 [API Gateway에서 프라이빗 API에 대한 사용자 지정 도메인 이름](https://docs.aws.amazon.com/apigateway/latest/developerguide/apigateway-private-custom-domains.html)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteDomainNameAccessAssociation](https://awscli.amazonaws.com/v2/documentation/api/latest/reference/apigateway/delete-domain-name-access-association.html) 섹션을 참조하세요.

delete-domain-name

다음 코드 예시는 `delete-domain-name`의 사용 방법을 보여 줍니다.

AWS CLI

사용자 지정 도메인 이름을 삭제하는 방법

명령:

```
aws apigateway delete-domain-name --domain-name 'api.domain.tld'
```

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteDomainName](https://awscli.amazonaws.com/v2/documentation/api/latest/reference/apigateway/delete-domain-name.html)을 참조하세요.

delete-integration-response

다음 코드 예시는 `delete-integration-response`의 사용 방법을 보여 줍니다.

AWS CLI

API에서 지정된 리소스, 메서드 및 상태 코드에 대한 통합 응답을 삭제하는 방법

명령:

```
aws apigateway delete-integration-response --rest-api-id 1234123412 --resource-
id a1b2c3 --http-method GET --status-code 200
```

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteIntegrationResponse](https://awscli.amazonaws.com/v2/documentation/api/latest/reference/apigateway/delete-integration-response.html)를 참조하세요.

delete-integration

다음 코드 예시는 delete-integration의 사용 방법을 보여 줍니다.

AWS CLI

API에서 지정된 리소스 및 메서드에 대한 통합을 삭제하는 방법

명령:

```
aws apigateway delete-integration --rest-api-id 1234123412 --resource-id a1b2c3 --
http-method GET
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteIntegration을 참조하세요.

delete-method-response

다음 코드 예시는 delete-method-response의 사용 방법을 보여 줍니다.

AWS CLI

API에서 지정된 리소스, 메서드 및 상태 코드에 대한 메서드 응답을 삭제하는 방법

명령:

```
aws apigateway delete-method-response --rest-api-id 1234123412 --resource-id a1b2c3
 --http-method GET --status-code 200
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteMethodResponse를 참조하세요.

delete-method

다음 코드 예시는 delete-method의 사용 방법을 보여 줍니다.

AWS CLI

API에서 지정된 리소스에 대한 메서드를 삭제하는 방법

명령:

```
aws apigateway delete-method --rest-api-id 1234123412 --resource-id a1b2c3 --http-
method GET
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteMethod를 참조하세요.

delete-model

다음 코드 예시는 delete-model의 사용 방법을 보여 줍니다.

AWS CLI

지정된 API에서 모델을 삭제하는 방법

명령:

```
aws apigateway delete-model --rest-api-id 1234123412 --model-name 'customModel'
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteModel을 참조하세요.

delete-resource

다음 코드 예시는 delete-resource의 사용 방법을 보여 줍니다.

AWS CLI

API에서 리소스를 삭제하는 방법

명령:

```
aws apigateway delete-resource --rest-api-id 1234123412 --resource-id a1b2c3
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteResource를 참조하세요.

delete-rest-api

다음 코드 예시는 delete-rest-api의 사용 방법을 보여 줍니다.

AWS CLI

API를 삭제하는 방법

명령:

```
aws apigateway delete-rest-api --rest-api-id 1234123412
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteRestApi를 참조하세요.

delete-stage

다음 코드 예시는 delete-stage의 사용 방법을 보여 줍니다.

AWS CLI

API에서 스테이지를 삭제하는 방법

명령:

```
aws apigateway delete-stage --rest-api-id 1234123412 --stage-name 'dev'
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteStage를 참조하세요.

delete-usage-plan-key

다음 코드 예시는 delete-usage-plan-key의 사용 방법을 보여 줍니다.

AWS CLI

사용 계획에서 API 키를 제거하는 방법

명령:

```
aws apigateway delete-usage-plan-key --usage-plan-id a1b2c3 --key-
id 1NbjQzMReAkeEQPNAW8r3dXsU2rDD7fc7f2Sipnu
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteUsagePlanKey를 참조하세요.

delete-usage-plan

다음 코드 예시는 delete-usage-plan의 사용 방법을 보여 줍니다.

AWS CLI

사용 계획을 삭제하는 방법

명령:

```
aws apigateway delete-usage-plan --usage-plan-id a1b2c3
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteUsagePlan을 참조하세요.

flush-stage-authorizers-cache

다음 코드 예시는 flush-stage-authorizers-cache의 사용 방법을 보여 줍니다.

AWS CLI

스테이지의 모든 권한 부여자 캐시 항목을 비우는 방법

명령:

```
aws apigateway flush-stage-authorizers-cache --rest-api-id 1234123412 --stage-name dev
```

- API 세부 정보는 AWS CLI 명령 참조의 FlushStageAuthorizersCache를 참조하세요.

flush-stage-cache

다음 코드 예시는 flush-stage-cache의 사용 방법을 보여 줍니다.

AWS CLI

API 스테이지에 대한 캐시를 비우는 방법

다음 flush-stage-cache 예제에서는 스테이지의 캐시를 플러시합니다.

```
aws apigateway flush-stage-cache \
    --rest-api-id 1234123412 \
    --stage-name dev
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon API Gateway 개발자 안내서의 API Gateway에서 API 스테이지 캐시 플러시를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 FlushStageCache를 참조하세요.

generate-client-certificate

다음 코드 예시는 generate-client-certificate의 사용 방법을 보여 줍니다.

AWS CLI

클라이언트 측 SSL 인증서를 생성하는 방법

명령:

```
aws apigateway generate-client-certificate --description 'My First Client
 Certificate'
```

- API 세부 정보는 AWS CLI 명령 참조의 GenerateClientCertificate를 참조하세요.

get-account

다음 코드 예시는 get-account의 사용 방법을 보여 줍니다.

AWS CLI

API Gateway 계정 설정을 가져오는 방법

명령:

```
aws apigateway get-account
```

출력:

```
{
    "cloudwatchRoleArn": "arn:aws:iam::123412341234:role/
APIGatewayToCloudWatchLogsRole",
    "throttleSettings": {
        "rateLimit": 500.0,
        "burstLimit": 1000
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetAccount를 참조하세요.

get-api-key

다음 코드 예시는 get-api-key의 사용 방법을 보여 줍니다.

AWS CLI

특정 API 키에 대한 정보를 가져오는 방법

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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetApiKey를 참조하세요.

get-api-keys

다음 코드 예시는 get-api-keys의 사용 방법을 보여 줍니다.

AWS CLI

API 키 목록을 가져오는 방법

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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetApiKeys를 참조하세요.

get-authorizer

다음 코드 예시는 `get-authorizer`의 사용 방법을 보여 줍니다.

AWS CLI

API Gateway API별 권한 부여자 설정을 가져오는 방법

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"
```

```
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetAuthorizer를 참조하세요.

get-authorizers

다음 코드 예시는 `get-authorizers`의 사용 방법을 보여 줍니다.

AWS CLI

REST API에 대한 권한 부여자의 목록을 가져오는 방법

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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetAuthorizers를 참조하세요.

get-base-path-mapping

다음 코드 예시는 `get-base-path-mapping`의 사용 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetBasePathMapping을 참조하세요.

get-base-path-mappings

다음 코드 예시는 `get-base-path-mappings`의 사용 방법을 보여 줍니다.

AWS CLI

사용자 지정 도메인 이름의 기본 경로 매핑들을 가져오는 방법

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"
        }
```

```
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetBasePathMappings를 참조하세요.

get-client-certificate

다음 코드 예시는 `get-client-certificate`의 사용 방법을 보여 줍니다.

AWS CLI

클라이언트 인증서를 가져오는 방법

명령:

```
aws apigateway get-client-certificate --client-certificate-id a1b2c3
```

- API 세부 정보는 AWS CLI 명령 참조의 GetClientCertificate를 참조하세요.

get-client-certificates

다음 코드 예시는 `get-client-certificates`의 사용 방법을 보여 줍니다.

AWS CLI

클라이언트 인증서의 목록을 가져오는 방법

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
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetClientCertificates를 참조하세요.

get-deployment

다음 코드 예시는 `get-deployment`의 사용 방법을 보여 줍니다.

AWS CLI

배포에 대한 정보를 가져오는 방법

명령:

```
aws apigateway get-deployment --rest-api-id 1234123412 --deployment-id ztt4m2
```

출력:

```
{
    "description": "myDeployment",
    "id": "ztt4m2",
    "createdDate": 1455218022
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetDeployment 섹션을 참조하세요.

get-deployments

다음 코드 예시는 `get-deployments`의 사용 방법을 보여 줍니다.

AWS CLI

REST API에 대한 배포의 목록을 가져오는 방법

명령:

```
aws apigateway get-deployments --rest-api-id 1234123412
```

출력:

```
{
    "items": [
        {
            "createdDate": 1453797217,
            "id": "0a2b4c",
            "description": "Deployed my API for the first time"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetDeployments를 참조하세요.

get-domain-name-access-associations

다음 코드 예시는 get-domain-name-access-associations의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 모든 도메인 이름 액세스 연결 나열

다음 get-domain-name-access-associations 예제에서는 모든 도메인 이름 액세스 연결을 나열합니다.

```
aws apigateway get-domain-name-access-associations
```

출력:

```
{
    "items": [
        {
        "domainNameAccessAssociationArn": "arn:aws:apigateway:us-
west-2:012345678910:/domainnameaccessassociations/domainname/my.private.domain.tld/
vpcesource/vpce-abcd1234efg
        "accessAssociationSource": "vpce-abcd1234efg",
        "accessAssociationSourceType": "VPCE",
        "domainNameArn" : "arn:aws:apigateway:us-west-2:111122223333:/domainnames/
private.example.com+abcd1234"
        }
    ]
```

```
}
```

자세한 내용은 Amazon API Gateway 개발자 안내서의 API Gateway에서 프라이빗 API에 대한 사용자 지정 도메인 이름을 참조하세요.

예제 2: 이 AWS 계정이 소유한 모든 도메인 이름 액세스 연결 나열

다음 `get-domain-name-access-associations` 예제에서는 현재 AWS 계정이 소유한 모든 도메인 이름 액세스 연결을 나열합니다.

```
aws apigateway get-domain-name-access-associations \
    --resource-owner SELF
```

출력:

```
{
    "items": [
        {
        "domainNameAccessAssociationArn": "arn:aws:apigateway:us-
west-2:012345678910:/domainnameaccessassociations/domainname/my.private.domain.tld/
vpcesource/vpce-abcd1234efg
        "accessAssociationSource": "vpce-abcd1234efg",
        "accessAssociationSourceType": "VPCE",
        "domainNameArn" : "arn:aws:apigateway:us-west-2:111122223333:/domainnames/
private.example.com+abcd1234"
        }
    ]
}
```

자세한 내용은 Amazon API Gateway 개발자 안내서의 API Gateway에서 프라이빗 API에 대한 사용자 지정 도메인 이름을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetDomainNameAccessAssociations를 참조하세요.

get-domain-name

다음 코드 예시는 `get-domain-name`의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 퍼블릭 사용자 지정 도메인 이름에 대한 정보를 가져오는 방법

다음 get-domain-name 예제에서는 퍼블릭 사용자 지정 도메인 이름에 대한 정보를 가져옵니다.

```
aws apigateway get-domain-name \
    --domain-name api.domain.tld
```

출력:

```
{
    "domainName": "api.domain.tld",
    "distributionDomainName": "d1a2f3a4c5o6d.cloudfront.net",
    "certificateName": "uploadedCertificate",
    "certificateUploadDate": 1462565487
}
```

자세한 내용은 Amazon API Gateway 개발자 안내서의 API Gateway에서 퍼블릭 REST API에 대한 사용자 지정 도메인 이름을 참조하세요.

예제 2: 프라이빗 사용자 지정 도메인 이름에 대한 정보를 가져오는 방법

다음 get-domain-name 예제에서는 프라이빗 사용자 지정 도메인 이름에 대한 정보를 가져옵니다.

```
aws apigateway get-domain-name \
    --domain-name api.private.domain.tld \
    --domain-name-id abcd1234
```

출력:

```
{
    "domainName": "my.private.domain.tld",
    "domainNameId": "abcd1234",
    "domainNameArn": "arn:aws:apigateway:us-east-1:012345678910:/domainnames/
my.private.domain.tld+abcd1234",
    "certificateArn": "arn:aws:acm:us-east-1:012345678910:certificate/fb1b9770-
a305-495d-aefb-27e5e101ff3",
    "certificateUploadDate": "2024-09-10T10:31:20-07:00",
    "endpointConfiguration": {
        "types": [
            "PRIVATE"
        ]
    },
    "domainNameStatus": "AVAILABLE",
```

```
    "securityPolicy": "TLS_1_2",
    "policy": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":
\"Allow\",\"Principal\":\"*\",\"Action\":\"execute-api:Invoke\",\"Resource\":
\"arn:aws:execute-api:us-east-1:012345678910:/domainnames/my.private.domain.tld
+abcd1234\"},{\"Effect\":\"Deny\",\"Principal\":\"*\",\"Action\":\"execute-
api:Invoke\",\"Resource\":\"arn:aws:execute-api:us-east-1:012345678910:/domainnames/
my.private.domain.tld+abcd1234\",\"Condition\":{\"StringNotEquals\":{\"aws:SourceVpc
\":\"vpc-1a2b3c4d\"}}}]}"
}
```

자세한 내용은 Amazon API Gateway 개발자 안내서의 API Gateway에서 퍼블릭 REST API에 대한 사용자 지정 도메인 이름을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetDomainName을 참조하세요.

get-domain-names

다음 코드 예시는 `get-domain-names`의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 사용자 지정 도메인 이름의 목록을 가져오는 방법

다음 `get-domain-names` 명령은 도메인 이름 목록을 가져옵니다.

```
aws apigateway get-domain-names
```

출력:

```
{
    "items": [
        {
            "distributionDomainName": "d9511k3l09bkd.cloudfront.net",
            "certificateUploadDate": 1452812505,
            "certificateName": "my_custom_domain-certificate",
            "domainName": "subdomain.domain.tld"
        }
    ]
}
```

자세한 내용은 Amazon API Gateway 개발자 안내서의 API Gateway에서 프라이빗 API에 대한 사용자 지정 도메인 이름을 참조하세요.

예제 2: 이 AWS 계정이 소유한 사용자 지정 도메인 이름 목록을 가져오는 방법

다음 get-domain-names 명령은 이 AWS 계정이 소유한 도메인 이름 목록을 가져옵니다.

```
aws apigateway get-domain-names \
    --resource-owner SELF
```

출력:

```
{
    "items": [
        {
            "domainName": "my.domain.tld",
            "domainNameArn": "arn:aws:apigateway:us-east-1::/domainnames/
my.private.domain.tld",
            "certificateUploadDate": "2024-08-15T17:02:55-07:00",
            "regionalDomainName": "d-abcd1234.execute-api.us-east-1.amazonaws.com",
            "regionalHostedZoneId": "Z1UJRXOUMOOFQ8",
            "regionalCertificateArn": "arn:aws:acm:us-
east-1:012345678910:certificate/fb1b9770-a305-495d-aefb-27e5e101ff3",
            "endpointConfiguration": {
                "types": [
                    "REGIONAL"
                ]
            },
            "domainNameStatus": "AVAILABLE",
            "securityPolicy": "TLS_1_2"
        },
        {
            "domainName": "my.private.domain.tld",
            "domainNameId": "abcd1234",
            "domainNameArn": "arn:aws:apigateway:us-east-1:012345678910:/
domainnames/my.private.domain.tld+abcd1234",
            "certificateArn": "arn:aws:acm:us-east-1:012345678910:certificate/
fb1b9770-a305-495d-aefb-27e5e101ff3",
            "certificateUploadDate": "2024-11-26T11:44:40-08:00",
            "endpointConfiguration": {
                "types": [
                    "PRIVATE"
                ]
            },
            "domainNameStatus": "AVAILABLE",
            "securityPolicy": "TLS_1_2"
```

```
        }
    ]
}
```

자세한 내용은 Amazon API Gateway 개발자 안내서의 API Gateway에서 프라이빗 API에 대한 사용자 지정 도메인 이름을 참조하세요.

예제 3: 도메인 이름 액세스 연결을 생성할 수 있는 다른 AWS 계정이 소유한 사용자 지정 도메인 이름 목록을 가져오는 방법.

다음 `get-domain-names` 명령은 도메인 이름 액세스 연결을 생성할 수 있는 액세스 권한이 있는 다른 AWS 계정이 소유한 도메인 이름 목록을 가져옵니다.

```
aws apigateway get-domain-names \
    --resource-owner OTHER_ACCOUNTS
```

출력:

```
{
    "items": [
        {
            "domainName": "my.private.domain.tld",
            "domainNameId": "abcd1234",
            "domainNameArn": "arn:aws:apigateway:us-east-1:012345678910:/
domainnames/my.private.domain.tld+abcd1234"
        }
    ]
}
```

자세한 내용은 Amazon API Gateway 개발자 안내서의 API Gateway에서 프라이빗 API에 대한 사용자 지정 도메인 이름을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetDomainNames를 참조하세요.

`get-export`

다음 코드 예시는 `get-export`의 사용 방법을 보여 줍니다.

AWS CLI

스테이지에 대한 JSON Swagger 템플릿을 가져오는 방법

명령:

```
aws apigateway get-export --rest-api-id a1b2c3d4e5 --stage-name dev --export-
type swagger /path/to/filename.json
```

스테이지에 대한 JSON Swagger 템플릿 + API Gateway 확장을 가져오는 방법

명령:

```
aws apigateway get-export --parameters extensions='integrations' --rest-api-
id a1b2c3d4e5 --stage-name dev --export-type swagger /path/to/filename.json
```

스테이지에 대한 JSON Swagger 템플릿 + Postman 확장을 가져오는 방법

명령:

```
aws apigateway get-export --parameters extensions='postman' --rest-api-id a1b2c3d4e5
 --stage-name dev --export-type swagger /path/to/filename.json
```

- API 세부 정보는 AWS CLI명령 참조의 GetExport를 참조하세요.

get-integration-response

다음 코드 예시는 `get-integration-response`의 사용 방법을 보여 줍니다.

AWS CLI

REST API의 리소스에서 정의된 HTTP 메서드에 대한 통합 응답 구성을 가져오는 방법

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
    }
```

```
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetIntegrationResponse를 참조하세요.

get-integration

다음 코드 예시는 get-integration의 사용 방법을 보여 줍니다.

AWS CLI

REST API의 리소스에서 정의된 HTTP 메서드에 대한 통합 구성을 가져오는 방법

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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetIntegration을 참조하세요.

get-method-response

다음 코드 예시는 get-method-response의 사용 방법을 보여 줍니다.

AWS CLI

REST API의 리소스에서 정의된 HTTP 메서드에 대한 메서드 응답 리소스 구성을 가져오는 방법

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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetMethodResponse를 참조하세요.

get-method

다음 코드 예시는 get-method의 사용 방법을 보여 줍니다.

AWS CLI

REST API의 리소스에서 정의된 HTTP 메서드에 대한 메서드 리소스 구성을 가져오는 방법

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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetMethod를 참조하세요.

`get-model-template`

다음 코드 예시는 `get-model-template`의 사용 방법을 보여 줍니다.

AWS CLI

REST API에서 정의된 모델에 대한 매핑 템플릿을 가져오는 방법

명령:

```
aws apigateway get-model-template --rest-api-id 1234123412 --model-name Empty
```

출력:

```
{
```

```
    "value": "#set($inputRoot = $input.path('$'))\n{ }"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetModelTemplate을 참조하세요.

get-model

다음 코드 예시는 get-model의 사용 방법을 보여 줍니다.

AWS CLI

REST API에서 정의된 모델에 대한 구성을 가져오는 방법

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}"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetModel을 참조하세요.

get-models

다음 코드 예시는 get-models의 사용 방법을 보여 줍니다.

AWS CLI

REST API에 대한 모델의 목록을 가져오는 방법

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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI명령 참조의 GetModels를 참조하세요.

get-resource

다음 코드 예시는 get-resource의 사용 방법을 보여 줍니다.

AWS CLI

리소스에 대한 정보를 가져오는 방법

명령:

```
aws apigateway get-resource --rest-api-id 1234123412 --resource-id zwo0y3
```

출력:

```
{
```

```
    "path": "/path",
    "pathPart": "path",
    "id": "zwo0y3",
    "parentId": "uyokt6ij2g"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetResource를 참조하세요.

get-resources

다음 코드 예시는 get-resources의 사용 방법을 보여 줍니다.

AWS CLI

REST API에 사용할 리소스 목록을 가져오는 방법

명령:

```
aws apigateway get-resources --rest-api-id 1234123412
```

출력:

```
{
    "items": [
        {
            "path": "/resource/subresource",
            "resourceMethods": {
                "POST": {}
            },
            "id": "024ace",
            "pathPart": "subresource",
            "parentId": "ai5b02"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetResources를 참조하세요.

get-rest-api

다음 코드 예시는 get-rest-api의 사용 방법을 보여 줍니다.

AWS CLI

API에 대한 정보를 가져오는 방법

명령:

```
aws apigateway get-rest-api --rest-api-id 1234123412
```

출력:

```
{
    "name": "myAPI",
    "id": "o1y243m4f5",
    "createdDate": 1453416433
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetRestApi를 참조하세요.

get-rest-apis

다음 코드 예시는 `get-rest-apis`의 사용 방법을 보여 줍니다.

AWS CLI

REST API 목록을 가져오는 방법

명령:

```
aws apigateway get-rest-apis
```

출력:

```
{
    "items": [
        {
            "createdDate": 1438884790,
            "id": "12s44z21rb",
            "name": "My First API"
        }
    ]
```

```
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetRestApis를 참조하세요.

get-sdk

다음 코드 예시는 get-sdk의 사용 방법을 보여 줍니다.

AWS CLI

REST API 스테이지에 대한 Android SDK를 가져오는 방법

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 스테이지에 대한 IOS SDK를 가져오는 방법

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 스테이지에 대한 Javascript SDK를 가져오는 방법

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
\""
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetSdk를 참조하세요.

get-stage

다음 코드 예시는 get-stage의 사용 방법을 보여 줍니다.

AWS CLI

API의 스테이지에 대한 정보를 가져오는 방법

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
        }
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetStage를 참조하세요.

get-stages

다음 코드 예시는 `get-stages`의 사용 방법을 보여 줍니다.

AWS CLI

REST API에 대한 스테이지의 목록을 가져오는 방법

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
                }
            }
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetStages를 참조하세요.

get-usage-plan-key

다음 코드 예시는 `get-usage-plan-key`의 사용 방법을 보여 줍니다.

AWS CLI

사용 계획과 연결된 API 키의 세부 정보를 가져오는 방법

명령:

```
aws apigateway get-usage-plan-key --usage-plan-id a1b2c3 --key-id 1NbjQzMReAkeEQPNAW8r3dXsU2rDD7fc7f2Sipnu
```

- API 세부 정보는 AWS CLI 명령 참조의 GetUsagePlanKey를 참조하세요.

get-usage-plan-keys

다음 코드 예시는 `get-usage-plan-keys`의 사용 방법을 보여 줍니다.

AWS CLI

사용 계획과 연결된 API 키의 목록을 가져오는 방법

명령:

```
aws apigateway get-usage-plan-keys --usage-plan-id a1b2c3
```

- API 세부 정보는 AWS CLI 명령 참조의 GetUsagePlanKeys를 참조하세요.

get-usage-plan

다음 코드 예시는 `get-usage-plan`의 사용 방법을 보여 줍니다.

AWS CLI

사용 계획의 세부 정보를 가져오는 방법

명령:

```
aws apigateway get-usage-plan --usage-plan-id a1b2c3
```

- API 세부 정보는 AWS CLI 명령 참조의 GetUsagePlan을 참조하세요.

get-usage-plans

다음 코드 예시는 `get-usage-plans`의 사용 방법을 보여 줍니다.

AWS CLI

모든 사용 계획의 세부 정보를 가져오는 방법

명령:

```
aws apigateway get-usage-plans
```

- API 세부 정보는 AWS CLI 명령 참조의 GetUsagePlans를 참조하세요.

get-usage

다음 코드 예시는 get-usage의 사용 방법을 보여 줍니다.

AWS CLI

사용 계획에 대한 사용 세부 정보를 가져오는 방법

명령:

```
aws apigateway get-usage --usage-plan-id a1b2c3 --start-date "2016-08-16" --end-date "2016-08-17"
```

- API 세부 정보는 AWS CLI 명령 참조의 GetUsage를 참조하세요.

import-rest-api

다음 코드 예시는 import-rest-api의 사용 방법을 보여 줍니다.

AWS CLI

Swagger 템플릿을 가져오고 API를 생성하는 방법

명령:

```
aws apigateway import-rest-api --body 'file:///path/to/API_Swagger_template.json'
```

- API 세부 정보는 AWS CLI 명령 참조의 ImportRestApi를 참조하세요.

put-integration-response

다음 코드 예시는 put-integration-response의 사용 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'"'"'"}'
```

- API 세부 정보는 AWS CLI 명령 참조의 PutIntegrationResponse를 참조하세요.

put-integration

다음 코드 예시는 put-integration의 사용 방법을 보여 줍니다.

AWS CLI

모의 통합 요청을 생성하는 방법

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 생성하는 방법

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 엔드포인트를 사용하여 AWS 통합 요청을 생성하는 방법

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'
```

- API 세부 정보는 AWS CLI 명령 참조의 PutIntegration을 참조하세요.

put-method-response

다음 코드 예시는 put-method-response의 사용 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"
```

- API 세부 정보는 AWS CLI 명령 참조의 PutMethodResponse를 참조하세요.

put-method

다음 코드 예시는 put-method의 사용 방법을 보여 줍니다.

AWS CLI

권한 부여, API 키 및 사용자 지정 메서드 요청 헤더가 없는 API의 리소스에 대한 메서드를 생성하는 방법

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"
```

- API 세부 정보는 AWS CLI 명령 참조의 PutMethod를 참조하세요.

put-rest-api

다음 코드 예시는 put-rest-api의 사용 방법을 보여 줍니다.

AWS CLI

Swagger 템플릿을 사용하여 기존 API를 덮어쓰는 방법

명령:

```
aws apigateway put-rest-api --rest-api-id 1234123412 --mode overwrite --body
 'fileb:///path/to/API_Swagger_template.json'
```

Swagger 템플릿을 기존 API에 병합하는 방법

명령:

```
aws apigateway put-rest-api --rest-api-id 1234123412 --mode merge --body 'fileb:///
path/to/API_Swagger_template.json'
```

- API 세부 정보는 AWS CLI 명령 참조의 PutRestApi를 참조하세요.

reject-domain-name-access-association

다음 코드 예시는 reject-domain-name-access-association의 사용 방법을 보여 줍니다.

AWS CLI

도메인 이름 액세스 연결을 거부하려면

다음 reject-domain-name-access-association 예제에서는 프라이빗 사용자 지정 도메인 이름과 VPC 엔드포인트 간의 도메인 이름 액세스 연결을 거부합니다.

```
aws apigateway reject-domain-name-access-association \
    --domain-name-access-association-arn arn:aws:apigateway:us-west-2:012345678910:/
domainnameaccessassociations/domainname/my.private.domain.tld/vpcesource/vpce-
abcd1234efg \
```

```
    --domain-name-arn arn:aws:apigateway:us-east-1:012345678910:/domainnames/
my.private.domain.tld+abcd1234
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon API Gateway 개발자 안내서의 API Gateway에서 프라이빗 API에 대한 사용자 지정 도메인 이름을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RejectDomainNameAccessAssociation 섹션을 참조하세요.

`test-invoke-authorizer`

다음 코드 예시는 `test-invoke-authorizer`의 사용 방법을 보여 줍니다.

AWS CLI

사용자 지정 권한 부여자에게 필수 헤더 및 값을 포함하여 보내는 요청의 호출을 테스트하는 방법

명령:

```
aws apigateway test-invoke-authorizer --rest-api-id 1234123412 --authorizer-
id 5yid1t --headers Authorization='Value'
```

- API 세부 정보는 AWS CLI 명령 참조의 TestInvokeAuthorizer를 참조하세요.

`test-invoke-method`

다음 코드 예시는 `test-invoke-method`의 사용 방법을 보여 줍니다.

AWS CLI

GET 요청을 수행하여 API에서 루트 리소스의 호출을 테스트하는 방법

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 수행하여 API에서 하위 리소스의 호출을 테스트하는 방법

명령:

```
aws apigateway test-invoke-method --rest-api-id 1234123412 --resource-id 3gapai --
http-method GET --path-with-query-string '/pets/1'
```

- API 세부 정보는 AWS CLI 명령 참조의 TestInvokeMethod를 참조하세요.

update-account

다음 코드 예시는 update-account의 사용 방법을 보여 줍니다.

AWS CLI

CloudWatch 로그에 로깅하는 IAM 역할 ARN을 변경하는 방법

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
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateAccount를 참조하세요.

update-api-key

다음 코드 예시는 update-api-key의 사용 방법을 보여 줍니다.

AWS CLI

API 키의 이름을 변경하는 방법

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 비활성화하는 방법

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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateApiKey를 참조하세요.

update-authorizer

다음 코드 예시는 update-authorizer의 사용 방법을 보여 줍니다.

AWS CLI

사용자 지정 권한 부여자의 이름을 변경하는 방법

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 변경하는 방법

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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateAuthorizer를 참조하세요.

update-base-path-mapping

다음 코드 예시는 update-base-path-mapping의 사용 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateBasePathMapping을 참조하세요.

update-client-certificate

다음 코드 예시는 update-client-certificate의 사용 방법을 보여 줍니다.

AWS CLI

클라이언트 인증서의 설명을 업데이트하는 방법

명령:

```
aws apigateway update-client-certificate --client-certificate-id a1b2c3 --patch-operations op='replace',path='/description',value='My new description'
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateClientCertificate를 참조하세요.

`update-deployment`

다음 코드 예시는 `update-deployment`의 사용 방법을 보여 줍니다.

AWS CLI

배포의 설명을 변경하는 방법

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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateDeployment를 참조하세요.

`update-domain-name`

다음 코드 예시는 `update-domain-name`의 사용 방법을 보여 줍니다.

AWS CLI

사용자 지정 도메인 이름에 대한 인증서 이름을 변경하는 방법

다음 `update-domain-name` 예제에서는 사용자 지정 도메인에 대한 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 API Gateway에서 API에 대한 사용자 지정 도메인 이름 설정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateDomainName을 참조하세요.

update-integration-response

다음 코드 예시는 update-integration-response의 사용 방법을 보여 줍니다.

AWS CLI

'*' 정적 매핑을 사용하도록 통합 응답 헤더를 변경하는 방법

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 제거하는 방법

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'
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateIntegrationResponse을 참조하세요.

update-integration

다음 코드 예시는 `update-integration`의 사용 방법을 보여 줍니다.

AWS CLI

Input Passthrough로 구성된 'Content-Type: application/json' 매핑 템플릿을 추가하는 방법

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 'Content-Type: application/json' 매핑 템플릿을 업데이트(교체)하는 방법

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 Passthrough가 있는 'Content-Type: application/json'과 연결된 사용자 지정 템플릿을 업데이트(교체)하는 방법

명령:

```
aws apigateway update-integration \
    --rest-api-id a1b2c3d4e5 \
    --resource-id a1b2c3 \
    --http-method POST \
    --patch-operations "op='replace',path='requestTemplates/application~1json'"
```

'Content-Type: application/json' 매핑 템플릿을 제거하는 방법

명령:

```
aws apigateway update-integration \
    --rest-api-id a1b2c3d4e5 \
    --resource-id a1b2c3 \
    --http-method POST \
    --patch-operations "op='remove',path='/requestTemplates/application~1json'"
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateIntegration을 참조하세요.

update-method-response

다음 코드 예시는 update-method-response의 사용 방법을 보여 줍니다.

AWS CLI

메서드에서 200 응답에 대한 새 메서드 응답 헤더를 생성하고 이를 필요 없음(기본값)으로 정의하는 방법

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 삭제하는 방법

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"
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateMethodResponse를 참조하세요.

update-method

다음 코드 예시는 update-method의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: API 키를 요구하도록 메서드를 수정하는 방법

다음 update-method 예제에서는 API 키를 요구하도록 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 권한 부여를 요구하도록 메서드를 수정하는 방법

다음 update-method 예제에서는 IAM 권한 부여를 요구하도록 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 권한 부여를 요구하도록 메서드를 수정하는 방법

다음 `update-method` 예제에서는 Lambda 권한 부여를 요구하도록 메서드를 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 API Gateway CLI 및 REST API를 사용하여 사용 계획 생성, 구성 및 테스트와 API Gateway에서 REST API에 대한 액세스 제어 및 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateMethod를 참조하세요.

`update-model`

다음 코드 예시는 `update-model`의 사용 방법을 보여 줍니다.

AWS CLI

API에서 모델의 설명을 변경하는 방법

명령:

```
aws apigateway update-model --rest-api-id 1234123412 --model-name 'Empty' --patch-operations op=replace,path=/description,value='New Description'
```

API에서 모델의 스키마를 변경하는 방법

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\" }"'
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateModel을 참조하세요.

`update-resource`

다음 코드 예시는 `update-resource`의 사용 방법을 보여 줍니다.

AWS CLI

API에서 리소스를 이동하고 이를 다른 상위 리소스에 배치하는 방법

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"
}
```

API에서 리소스(pathPart)의 이름을 바꾸는 방법

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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateResource를 참조하세요.

`update-rest-api`

다음 코드 예시는 `update-rest-api`의 사용 방법을 보여 줍니다.

AWS CLI

API의 이름을 변경하는 방법

명령:

```
aws apigateway update-rest-api --rest-api-id 1234123412 --patch-operations
 op=replace,path=/name,value='New Name'
```

API의 설명을 변경하는 방법

명령:

```
aws apigateway update-rest-api --rest-api-id 1234123412 --patch-operations
 op=replace,path=/description,value='New Description'
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateRestApi를 참조하세요.

update-stage

다음 코드 예시는 `update-stage`의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 리소스 및 메서드에 대한 스테이지 설정을 재정의하는 방법

다음 `update-stage` 예제에서는 특정 리소스 및 메서드에 대한 스테이지 설정을 재정의하고 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 API Gateway 개발자 안내서의 REST API에 대한 스테이지 설정을 참조하세요.

예제 2: API 스테이지의 모든 리소스 및 메서드에 대한 스테이지 설정을 업데이트하는 방법

다음 `update-stage` 예제에서는 API 스테이지의 모든 리소스 및 메서드에 대한 전체 요청/응답 로깅을 켭니다.

```
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 API Gateway 개발자 안내서의 REST API에 대한 스테이지 설정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateStage를 참조하세요.

`update-usage-plan`

다음 코드 예시는 `update-usage-plan`의 사용 방법을 보여 줍니다.

AWS CLI

사용 계획에서 정의된 기간을 변경하는 방법

명령:

```
aws apigateway update-usage-plan --usage-plan-id a1b2c3 --patch-operations
 op="replace",path="/quota/period",value="MONTH"
```

사용 계획에서 정의된 할당량 제한을 변경하는 방법

명령:

```
aws apigateway update-usage-plan --usage-plan-id a1b2c3 --patch-operations
 op="replace",path="/quota/limit",value="500"
```

사용 계획에서 정의된 스로틀 요율 제한을 변경하는 방법

명령:

```
aws apigateway update-usage-plan --usage-plan-id a1b2c3 --patch-operations
 op="replace",path="/throttle/rateLimit",value="10"
```

사용 계획에서 정의된 스로틀 버스트 제한을 변경하는 방법

명령:

```
aws apigateway update-usage-plan --usage-plan-id a1b2c3 --patch-operations
 op="replace",path="/throttle/burstLimit",value="20"
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateUsagePlan을 참조하세요.

`update-usage`

다음 코드 예시는 `update-usage`의 사용 방법을 보여 줍니다.

AWS CLI

사용 계획에서 정의된 현재 기간에 대해 API 키의 할당량을 일시적으로 수정하는 방법

명령:

```
aws apigateway update-usage --usage-plan-id a1b2c3 --key-
id 1NbjQzMReAkeEQPNAW8r3dXsU2rDD7fc7f2Sipnu --patch-operations op="replace",path="/
remaining",value="50"
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateUsage를 참조하세요.

AWS CLI를 사용하는 API Gateway HTTP 및 WebSocket API 예제

다음 코드 예제는 API Gateway HTTP 및 WebSocket API와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`create-api-mapping`

다음 코드 예시는 `create-api-mapping`의 사용 방법을 보여 줍니다.

AWS CLI

API에 대한 API 매핑을 생성하는 방법

다음 `create-api-mapping` 예제에서는 API의 `test` 스테이지를 `regional.example.com` 사용자 지정 도메인 이름의 `/myApi` 경로에 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 API Gateway에서 리전 사용자 지정 도메인 이름 설정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateApiMapping을 참조하세요.

`create-api`

다음 코드 예시는 `create-api`의 사용 방법을 보여 줍니다.

AWS CLI

HTTP API를 생성하는 방법

다음 `create-api` 예제에서는 빠른 생성을 사용하여 HTTP API를 생성합니다. 빠른 생성을 사용하여 AWS Lambda 또는 HTTP 통합, 기본 catch-all 경로, 변경 사항을 자동 배포하도록 구성된 기본 스테이지를 통해 API를 생성할 수 있습니다. 다음 명령은 빠른 생성을 사용하여 Lambda 함수와 통합되는 HTTP API를 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 안내서의 API Gateway에서 HTTP API 개발을 참조하세요.

WebSocket API를 생성하는 방법

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 API Gateway에서 WebSocket API 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [CreateApi]를 참조하세요.

`create-authorizer`

다음 코드 예시는 `create-authorizer`의 사용 방법을 보여 줍니다.

AWS CLI

HTTP API에 대한 JWT 권한 부여자를 생성하는 방법

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 사용하여 HTTP API에 대한 액세스 제어]를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [CreateAuthorizer]를 참조하세요.

create-deployment

다음 코드 예시는 create-deployment의 사용 방법을 보여 줍니다.

AWS CLI

API에 대한 배포를 생성하는 방법

다음 create-deployment 예제에서는 API에 대한 배포를 생성하고 이 배포를 API의 dev 스테이지와 연결합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateDeployment를 참조하세요.

create-domain-name

다음 코드 예시는 create-domain-name의 사용 방법을 보여 줍니다.

AWS CLI

사용자 지정 도메인 이름을 생성하는 방법

다음 create-domain-name 예제에서는 API에 대한 리전 사용자 지정 도메인 이름을 생성합니다.

```
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 [API Gateway에서 리전 사용자 지정 도메인 이름 설정]을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [CreateDomainName]을 참조하세요.

`create-integration`

다음 코드 예시는 `create-integration`의 사용 방법을 보여 줍니다.

AWS CLI

WebSocket API 통합을 생성하는 방법

다음 `create-integration` 예제에서는 WebSocket API에 대한 모의 통합을 생성합니다.

```
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 API Gateway에서 WebSocket API 통합 요청 설정을 참조하세요.

HTTP API 통합을 생성하는 방법

다음 `create-integration` 예제에서는 HTTP API에 대한 AWS Lambda 통합을 생성합니다.

```
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 안내서의 HTTP API에 대한 통합 구성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateIntegration을 참조하세요.

`create-route`

다음 코드 예시는 `create-route`의 사용 방법을 보여 줍니다.

AWS CLI

WebSocket 또는 HTTP API에 대한 $default 경로를 생성하는 방법

다음 `create-route` 예제에서는 WebSocket 또는 HTTP API에 대한 `$default` 경로를 생성합니다.

```
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 [WebSocket API에 대한 라우팅 작업](#)을 참조하세요.

HTTP API에 대한 경로를 생성하는 방법

다음 `create-route` 예제에서는 POST 요청을 수락하는 `signup`이라는 이름의 경로를 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 안내서의 HTTP API에 대한 라우팅 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateRoute를 참조하세요.

`create-routing-rule`

다음 코드 예시는 `create-routing-rule`의 사용 방법을 보여 줍니다.

AWS CLI

라우팅 규칙을 만들려면

다음 `create-routing-rule` 예제에서는 우선 순위가 `50`인 라우팅 규칙을 생성합니다.

```
aws apigatewayv2 create-routing-rule \
    --domain-name 'regional.example.com' \
    --priority 50 \
    --conditions '[ \
        { \
            "MatchBasePaths": { \
                "AnyOf": [ \
                    "PetStoreShopper" \
                ] \
            } \
        } \
    ]' \
    --actions '[ \
        { \
            "InvokeApi": { \
                "ApiId": "abcd1234", \
                "Stage": "prod" \
            } \
        } \
    ]'
```

출력:

```
{
    "Actions": [
        {
            "InvokeApi": {
                "ApiId": "abcd1234",
```

```
                "Stage": "prod",
                "StripBasePath": false
            }
        }
    ],
    "Conditions": [
        {
            "MatchBasePaths": {
                "AnyOf": [
                    "PetStoreShopper"
                ]
            }
        }
    ],
    "Priority": 50,
    "RoutingRuleArn": "arn:aws:apigateway:us-east-2:123456789012:/domainnames/
regional.example.com/routingrules/aaa111",
    "RoutingRuleId": "aaa111"
}
```

자세한 내용은 Amazon API Gateway 개발자 안내서의 Routing rules to connect API stages to a custom domain name for REST APIs를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateRoutingRule 섹션을 참조하세요.

`create-stage`

다음 코드 예시는 `create-stage`의 사용 방법을 보여 줍니다.

AWS CLI

스테이지를 생성하는 방법

다음 `create-stage` 예제에서는 API에 대해 dev라는 이름의 스테이지를 생성합니다.

```
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 API Gateway 개발자 안내서의 HTTP API에 대한 스테이지 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateStage를 참조하세요.

`create-vpc-link`

다음 코드 예시는 `create-vpc-link`의 사용 방법을 보여 줍니다.

AWS CLI

HTTP API에 대한 VPC 링크를 생성하는 방법

다음 `create-vpc-link` 예제에서는 HTTP API에 대한 VPC 링크를 생성합니다.

```
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 API Gateway 개발자 안내서의 HTTP API에 대한 VPC 링크 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateVpcLink를 참조하세요.

delete-access-log-settings

다음 코드 예시는 delete-access-log-settings의 사용 방법을 보여 줍니다.

AWS CLI

API에 대한 액세스 로깅을 비활성화하는 방법

다음 delete-access-log-settings 예제에서는 API의 $default 스테이지에 대한 액세스 로그 설정을 삭제합니다. 스테이지에 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 API Gateway 개발자 안내서의 HTTP API에 대한 로깅 구성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteAccessLogSettings를 참조하세요.

delete-api-mapping

다음 코드 예시는 delete-api-mapping의 사용 방법을 보여 줍니다.

AWS CLI

API 매핑을 삭제하는 방법

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 API Gateway에서 리전 사용자 지정 도메인 이름 설정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteApiMapping을 참조하세요.

delete-api

다음 코드 예시는 delete-api의 사용 방법을 보여 줍니다.

AWS CLI

API를 삭제하는 방법

다음 delete-api 예제에서는 API를 삭제합니다.

```
aws apigatewayv2 delete-api \
    --api-id a1b2c3d4
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon API Gateway 개발자 안내서의 HTTP API 작업과 WebSocket API 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteApi를 참조하세요.

delete-authorizer

다음 코드 예시는 delete-authorizer의 사용 방법을 보여 줍니다.

AWS CLI

권한 부여자를 삭제하는 방법

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 사용하여 HTTP API에 대한 액세스 제어를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteAuthorizer를 참조하세요.

delete-cors-configuration

다음 코드 예시는 `delete-cors-configuration`의 사용 방법을 보여 줍니다.

AWS CLI

HTTP API에 대한 CORS 구성을 삭제하는 방법

다음 `delete-cors-configuration` 예제에서는 HTTP API의 CORS 구성을 삭제하여 해당 HTTP API에 대한 CORS를 비활성화합니다.

```
aws apigatewayv2 delete-cors-configuration \
    --api-id a1b2c3d4
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon API Gateway 개발자 안내서의 HTTP API에 대한 CORS 구성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteCorsConfiguration을 참조하세요.

delete-deployment

다음 코드 예시는 `delete-deployment`의 사용 방법을 보여 줍니다.

AWS CLI

배포를 삭제하는 방법

다음 `delete-deployment` 예제에서는 API의 배포를 삭제합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteDeployment를 참조하세요.

delete-domain-name

다음 코드 예시는 `delete-domain-name`의 사용 방법을 보여 줍니다.

AWS CLI

사용자 지정 도메인 이름을 삭제하는 방법

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 API Gateway에서 리전 사용자 지정 도메인 이름 설정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteDomainName을 참조하세요.

delete-integration

다음 코드 예시는 `delete-integration`의 사용 방법을 보여 줍니다.

AWS CLI

통합을 삭제하는 방법

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 HTTP API에 대한 통합 구성과 WebSocket API 통합 설정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteIntegration을 참조하세요.

`delete-route-settings`

다음 코드 예시는 `delete-route-settings`의 사용 방법을 보여 줍니다.

AWS CLI

경로 설정을 삭제하는 방법

다음 `delete-route-settings` 예제는 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 API Gateway 개발자 안내서의 HTTP API에 대한 라우팅 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteRouteSettings를 참조하세요.

`delete-route`

다음 코드 예시는 `delete-route`의 사용 방법을 보여 줍니다.

AWS CLI

경로를 삭제하는 방법

다음 `delete-route` 예제에서는 API 경로를 삭제합니다.

```
aws apigatewayv2 delete-route \
    --api-id a1b2c3d4 \
    --route-id a1b2c3
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon API Gateway 개발자 안내서의 HTTP API에 대한 라우팅 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteRoute를 참조하세요.

delete-routing-rule

다음 코드 예시는 delete-routing-rule의 사용 방법을 보여 줍니다.

AWS CLI

라우팅 규칙을 삭제하려면

다음 delete-routing-rule 예제에서는 사용자 지정 도메인 이름에 대한 라우팅 규칙을 삭제합니다.

```
aws apigatewayv2 delete-routing-rule \
    --domain-name 'regional.example.com' \
    --routing-rule-id aaa111
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon API Gateway 개발자 안내서의 Routing rules to connect API stages to a custom domain name for REST APIs를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteRoutingRule 섹션을 참조하세요.

delete-stage

다음 코드 예시는 delete-stage의 사용 방법을 보여 줍니다.

AWS CLI

스테이지를 삭제하는 방법

다음 delete-stage 예제에서는 API의 test 스테이지를 삭제합니다.

```
aws apigatewayv2 delete-stage \
    --api-id a1b2c3d4 \
    --stage-name test
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon API Gateway 개발자 안내서의 HTTP API에 대한 스테이지 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteStage를 참조하세요.

delete-vpc-link

다음 코드 예시는 `delete-vpc-link`의 사용 방법을 보여 줍니다.

AWS CLI

HTTP API에 대한 VPC 링크를 삭제하는 방법

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 API Gateway 개발자 안내서의 HTTP API에 대한 VPC 링크 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteVpcLink를 참조하세요.

export-api

다음 코드 예시는 `export-api`의 사용 방법을 보여 줍니다.

AWS CLI

HTTP API의 OpenAPI 정의를 내보내는 방법

다음 `export-api` 예제에서는 `prod`라는 이름의 API 스테이지의 OpenAPI 3.0 정의를 `stage-definition.yaml`이라는 이름의 YAML 파일로 내보냅니다. 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 API Gateway에서 HTTP API 내보내기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ExportApi를 참조하세요.

get-api-mapping

다음 코드 예시는 get-api-mapping의 사용 방법을 보여 줍니다.

AWS CLI

사용자 지정 도메인 이름의 API 매핑에 대한 정보를 가져오는 방법

다음 get-api-mapping 예제에서는 api.example.com 사용자 지정 도메인 이름의 API 매핑에 대한 정보를 표시합니다.

```
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 API Gateway에서 리전 사용자 지정 도메인 이름 설정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetApiMapping을 참조하세요.

`get-api-mappings`

다음 코드 예시는 `get-api-mappings`의 사용 방법을 보여 줍니다.

AWS CLI

사용자 지정 도메인 이름의 API 매핑들을 가져오는 방법

다음 `get-api-mappings` 예제에서는 `api.example.com` 사용자 지정 도메인 이름에 대한 모든 API 매핑의 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 [API Gateway에서 리전 사용자 지정 도메인 이름 설정](#)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [GetApiMappings](#)를 참조하세요.

`get-api`

다음 코드 예시는 `get-api`의 사용 방법을 보여 줍니다.

AWS CLI

API에 대한 정보를 검색하는 방법

다음 `get-api` 예제에서는 API에 대한 정보를 표시합니다.

```
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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetApi를 참조하세요.

get-apis

다음 코드 예시는 `get-apis`의 사용 방법을 보여 줍니다.

AWS CLI

API 목록을 검색하는 방법

다음 `get-apis` 예제에서는 현재 사용자에 대한 모든 API를 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 HTTP API 작업과 WebSocket API 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetApis를 참조하세요.

get-authorizer

다음 코드 예시는 `get-authorizer`의 사용 방법을 보여 줍니다.

AWS CLI

권한 부여자에 대한 정보를 검색하는 방법

다음 `get-authorizer` 예제에서는 권한 부여자에 대한 정보를 표시합니다.

```
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 사용하여 HTTP API에 대한 액세스 제어를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetAuthorizer를 참조하세요.

get-authorizers

다음 코드 예시는 get-authorizers의 사용 방법을 보여 줍니다.

AWS CLI

API에 대한 권한 부여자의 목록을 검색하는 방법

다음 get-authorizers 예제에서는 API에 대한 모든 권한 부여자의 목록을 표시합니다.

```
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 사용하여 HTTP API에 대한 액세스 제어를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetAuthorizers를 참조하세요.

get-deployment

다음 코드 예시는 get-deployment의 사용 방법을 보여 줍니다.

AWS CLI

배포에 대한 정보를 검색하는 방법

다음 get-deployment 예제에서는 배포에 대한 정보를 표시합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 GetDeployment를 참조하세요.

get-deployments

다음 코드 예시는 `get-deployments`의 사용 방법을 보여 줍니다.

AWS CLI

배포 목록을 검색하는 방법

다음 `get-deployments` 예제에서는 API의 모든 배포에 대한 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.

- API 세부 정보는 AWS CLI 명령 참조의 GetDeployments를 참조하세요.

get-domain-name

다음 코드 예시는 get-domain-name의 사용 방법을 보여 줍니다.

AWS CLI

사용자 지정 도메인 이름에 대한 정보를 검색하는 방법

다음 get-domain-name 예제에서는 사용자 지정 도메인 이름에 대한 정보를 표시합니다.

```
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 API Gateway에서 리전 사용자 지정 도메인 이름 설정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetDomainName을 참조하세요.

get-domain-names

다음 코드 예시는 get-domain-names의 사용 방법을 보여 줍니다.

AWS CLI

사용자 지정 도메인 이름의 목록을 검색하는 방법

다음 get-domain-names 예제에서는 현재 사용자에 대한 모든 사용자 지정 도메인 이름의 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 API Gateway에서 리전 사용자 지정 도메인 이름 설정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetDomainNames를 참조하세요.

get-integration

다음 코드 예시는 get-integration의 사용 방법을 보여 줍니다.

AWS CLI

통합에 대한 정보를 검색하는 방법

다음 get-integration 예제에서는 통합에 대한 정보를 표시합니다.

```
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 HTTP API에 대한 통합 구성과 WebSocket API 통합 설정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetIntegration을 참조하세요.

`get-integrations`

다음 코드 예시는 `get-integrations`의 사용 방법을 보여 줍니다.

AWS CLI

통합의 목록을 검색하는 방법

다음 `get-integrations` 예제에서는 API의 모든 통합에 대한 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 HTTP API에 대한 통합 구성과 WebSocket API 통합 설정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetIntegrations를 참조하세요.

get-route

다음 코드 예시는 get-route의 사용 방법을 보여 줍니다.

AWS CLI

경로에 대한 정보를 검색하는 방법

다음 get-route 예제에서는 경로에 대한 정보를 표시합니다.

```
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 안내서의 HTTP API에 대한 라우팅 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetRoute를 참조하세요.

get-routes

다음 코드 예시는 `get-routes`의 사용 방법을 보여 줍니다.

AWS CLI

경로 목록을 검색하는 방법

다음 `get-routes` 예제에서는 API의 모든 경로에 대한 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 안내서의 HTTP API에 대한 라우팅 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetRoutes를 참조하세요.

get-routing-rule

다음 코드 예시는 get-routing-rule의 사용 방법을 보여 줍니다.

AWS CLI

라우팅 규칙을 가져오려면

다음 get-routing-rule 예제에서는 도메인 이름에 대한 라우팅 규칙을 가져옵니다.

```
aws apigatewayv2 get-routing-rule \
    --domain-name 'regional.example.com' \
    --routing-rule-id aaa111
```

출력:

```
{
    "Actions": [
        {
            "InvokeApi": {
                "ApiId": "abcd1234",
                "Stage": "prod",
                "StripBasePath": false
            }
        }
    ],
    "Conditions": [
        {
            "MatchBasePaths": {
                "AnyOf": [
                    "PetStoreShopper"
                ]
            }
        }
    ],
    "Priority": 50,
    "RoutingRuleArn": "arn:aws:apigateway:us-east-2:123456789012:/domainnames/
regional.example.com/routingrules/aaa111",
```

```
    "RoutingRuleId": "aaa111"
}
```

자세한 내용은 Amazon API Gateway 개발자 안내서의 Routing rules to connect API stages to a custom domain name for REST APIs를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetRoutingRule 섹션을 참조하세요.

get-stage

다음 코드 예시는 get-stage의 사용 방법을 보여 줍니다.

AWS CLI

스테이지에 대한 정보를 검색하는 방법

다음 get-stage 예제에서는 API의 prod 스테이지에 대한 정보를 표시합니다.

```
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 API Gateway 개발자 안내서의 HTTP API에 대한 스테이지 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetStage를 참조하세요.

get-stages

다음 코드 예시는 get-stages의 사용 방법을 보여 줍니다.

AWS CLI

스테이지 목록을 검색하는 방법

다음 get-stages 예제에서는 API의 모든 스테이지를 나열합니다.

```
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 API Gateway 개발자 안내서의 HTTP API에 대한 스테이지 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetStages를 참조하세요.

get-tags

다음 코드 예시는 get-tags의 사용 방법을 보여 줍니다.

AWS CLI

리소스에 대한 태그의 목록을 검색하는 방법

다음 get-tags 예제에서는 API의 모든 태그를 나열합니다.

```
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 API Gateway 리소스 태그 지정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetTags를 참조하세요.

get-vpc-link

다음 코드 예시는 `get-vpc-link`의 사용 방법을 보여 줍니다.

AWS CLI

VPC 링크에 대한 정보를 검색하는 방법

다음 `get-vpc-link` 예제에서는 VPC 링크에 대한 정보를 표시합니다.

```
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 API Gateway 개발자 안내서의 HTTP API에 대한 VPC 링크 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetVpcLink를 참조하세요.

get-vpc-links

다음 코드 예시는 get-vpc-links의 사용 방법을 보여 줍니다.

AWS CLI

VPC 링크 목록을 검색하는 방법

다음 get-vpc-links 예제에서는 현재 사용자에 대한 모든 VPC 링크의 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 API Gateway 개발자 안내서의 HTTP API에 대한 VPC 링크 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetVpcLinks를 참조하세요.

import-api

다음 코드 예시는 import-api의 사용 방법을 보여 줍니다.

AWS CLI

HTTP API를 가져오는 방법

다음 import-api 예제에서는 api-definition.yaml이라는 이름의 OpenAPI 3.0 정의 파일에서 HTTP API를 생성합니다.

```
aws apigatewayv2 import-api \
    --body file://api-definition.yaml
```

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-2:123456789012: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 API Gateway 개발자 안내서의 HTTP API에 대한 OpenAPI 정의 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ImportApi를 참조하세요.

`list-routing-rules`

다음 코드 예시는 `list-routing-rules`의 사용 방법을 보여 줍니다.

AWS CLI

라우팅 규칙을 나열하려면

다음 `list-routing-rules` 예제에서는 도메인 이름에 대한 라우팅 규칙을 나열합니다.

```
aws apigatewayv2 list-routing-rules \
    --domain-name 'regional.example.com'
```

출력:

```
{
    "RoutingRules": [
        {
            "Actions": [
                {
                    "InvokeApi": {
                        "ApiId": "abcd1234",
                        "Stage": "prod",
                        "StripBasePath": false
                    }
                }
            ],
            "Conditions": [
                {
                    "MatchBasePaths": {
                        "AnyOf": [
                            "PetStoreShopper"
                        ]
                    }
                }
            ],
            "Priority": 150,
            "RoutingRuleArn": "arn:aws:apigateway:us-east-1:123456789012:/
domainnames/regional.example.com/routingrules/aaa111",
            "RoutingRuleId": "aaa111"
        }
    ]
}
```

자세한 내용은 Amazon API Gateway 개발자 안내서의 Routing rules to connect API stages to a custom domain name for REST APIs를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListRoutingRules 섹션을 참조하세요.

put-routing-rule

다음 코드 예시는 `put-routing-rule`의 사용 방법을 보여 줍니다.

AWS CLI

라우팅 규칙을 업데이트하려면

다음 put-routing-rule 예제에서는 라우팅 규칙의 우선 순위를 업데이트합니다.

```
aws apigatewayv2 put-routing-rule \
    --domain-name 'regional.example.com' \
    --priority 150 \
    --conditions '[ \
        { \
            "MatchBasePaths": { \
                "AnyOf": [ \
                    "PetStoreShopper" \
                ] \
            } \
        } \
    ]' \
    --actions '[ \
        { \
            "InvokeApi": { \
                "ApiId": "abcd1234", \
                "Stage": "prod" \
            } \
        } \
    ]'
```

출력:

```
{
    "Actions": [
        {
            "InvokeApi": {
                "ApiId": "abcd1234",
                "Stage": "prod",
                "StripBasePath": false
            }
        }
    ],
    "Conditions": [
        {
            "MatchBasePaths": {
                "AnyOf": [
                    "PetStoreShopper"
                ]
            }
        }
```

```
    ],
    "Priority": 150,
    "RoutingRuleArn": "arn:aws:apigateway:us-east-2:123456789012:/domainnames/
regional.example.com/routingrules/aaa111",
    "RoutingRuleId": "aaa111"
}
```

자세한 내용은 Amazon API Gateway 개발자 안내서의 Routing rules to connect API stages to a custom domain name for REST APIs를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutRoutingRule 섹션을 참조하세요.

reimport-api

다음 코드 예시는 `reimport-api`의 사용 방법을 보여 줍니다.

AWS CLI

HTTP API를 다시 가져오는 방법

다음 `reimport-api` 예제에서는 기존 HTTP API를 업데이트하여 `api-definition.yaml`에서 지정된 OpenAPI 3.0 정의를 사용하도록 합니다.

```
aws apigatewayv2 reimport-api \
    --body file://api-definition.yaml \
    --api-id a1b2c3d4
```

`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-2:12356789012: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 API Gateway 개발자 안내서의 HTTP API에 대한 OpenAPI 정의 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ReimportApi를 참조하세요.

tag-resource

다음 코드 예시는 tag-resource의 사용 방법을 보여 줍니다.

AWS CLI

리소스에 태그를 지정하는 방법

다음 tag-resource 예제에서는 지정된 API에 키 이름 Department 및 Accounting 값을 갖는 태그를 추가합니다.

```
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 API Gateway 리소스 태그 지정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

`untag-resource`

다음 코드 예시에서는 `untag-resource` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

리소스에서 태그를 제거하는 방법

다음 `untag-resource` 예제에서는 키 이름 `Project` 및 `Owner`가 있는 태그를 지정된 API에서 제거합니다.

```
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 [API Gateway 리소스 태그 지정]을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [UntagResource]를 참조하세요.

`update-api-mapping`

다음 코드 예시는 `update-api-mapping`의 사용 방법을 보여 줍니다.

AWS CLI

API 매핑을 업데이트하는 방법

다음 `update-api-mapping` 예제에서는 사용자 지정 도메인 이름에 대한 API 매핑을 변경합니다. 결과적으로, 지정된 API 및 스테이지에 대한 사용자 지정 도메인 이름을 사용하는 기본 URL은 `https://api.example.com/dev`가 됩니다.

```
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 API Gateway에서 리전 사용자 지정 도메인 이름 설정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateApiMapping을 참조하세요.

update-api

다음 코드 예시는 update-api의 사용 방법을 보여 줍니다.

AWS CLI

HTTP API에 대해 CORS를 활성화하는 방법

다음 update-api 예제에서는 지정된 API의 CORS 구성을 업데이트하여 https://www.example.com으로부터의 요청을 허용하도록 합니다.

```
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 API Gateway 개발자 안내서의 HTTP API에 대한 CORS 구성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateApi를 참조하세요.

update-authorizer

다음 코드 예시는 update-authorizer의 사용 방법을 보여 줍니다.

AWS CLI

권한 부여자를 업데이트하는 방법

다음 update-authorizer 예제에서는 JWT 권한 부여자의 자격 증명 소스를 Authorization이라는 이름의 헤더로 변경합니다.

```
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 사용하여 HTTP API에 대한 액세스 제어](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [UpdateAuthorizer](#)를 참조하세요.

update-deployment

다음 코드 예시는 `update-deployment`의 사용 방법을 보여 줍니다.

AWS CLI

배포의 설명을 변경하는 방법

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 안내서의 [API Gateway에서 HTTP API 개발](#)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [UpdateDeployment](#)를 참조하세요.

update-domain-name

다음 코드 예시는 update-domain-name의 사용 방법을 보여 줍니다.

AWS CLI

사용자 지정 도메인 이름을 업데이트하는 방법

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 API Gateway에서 리전 사용자 지정 도메인 이름 설정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateDomainName을 참조하세요.

update-integration

다음 코드 예시는 update-integration의 사용 방법을 보여 줍니다.

AWS CLI

Lambda 통합을 업데이트하는 방법

다음 update-integration 예제에서는 지정된 Lambda 함수를 사용하도록 기존 AWS 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 HTTP API에 대한 통합 구성과 WebSocket API 통합 설정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateIntegration을 참조하세요.

update-route

다음 코드 예시는 update-route의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 경로의 통합을 업데이트하는 방법

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 추가하는 방법

다음 `update-route` 예제에서는 지정된 경로를 업데이트하여 JWT 권한 부여자를 사용하도록 합니다.

```
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 사용하여 HTTP API에 대한 액세스 제어를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateRoute를 참조하세요.

update-stage

다음 코드 예시는 update-stage의 사용 방법을 보여 줍니다.

AWS CLI

사용자 지정 스로틀링을 구성하는 방법

다음 update-stage 예제에서는 API의 지정된 스테이지 및 경로에 대해 사용자 지정 스로틀링을 구성합니다.

```
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 안내서의 HTTP API 보호를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateStage를 참조하세요.

update-vpc-link

다음 코드 예시는 update-vpc-link의 사용 방법을 보여 줍니다.

AWS CLI

VPC 링크를 업데이트하는 방법

다음 update-vpc-link 예제에서는 VPC 링크의 이름을 업데이트합니다. VPC 링크를 생성한 후에는 해당 보안 그룹이나 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 API Gateway 개발자 안내서의 HTTP API에 대한 VPC 링크 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateVpcLink를 참조하세요.

AWS CLI를 사용하는 API Gateway Management API 예제

다음 코드 예제에서는 API Gateway Management API와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`delete-connection`

다음 코드 예시는 `delete-connection`의 사용 방법을 보여 줍니다.

AWS CLI

WebSocket 연결을 삭제하는 방법

다음 `delete-connection` 예제에서는 지정된 WebSocket API에서 클라이언트의 연결을 해제합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteConnection을 참조하세요.

get-connection

다음 코드 예시는 get-connection의 사용 방법을 보여 줍니다.

AWS CLI

WebSocket 연결에 대한 정보를 가져오는 방법

다음 get-connection 예제에서는 지정된 WebSocket API에 대한 연결을 설명합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 GetConnection을 참조하세요.

post-to-connection

다음 코드 예시는 post-to-connection의 사용 방법을 보여 줍니다.

AWS CLI

WebSocket 연결로 데이터를 전송하는 방법

다음 post-to-connection 예시에서는 지정된 WebSocket API에 연결된 클라이언트에 메시지를 보냅니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 PostToConnection을 참조하세요.

AWS CLI를 사용하는 App Mesh 예제

다음 코드 예제에서는 App Mesh와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

작업

`create-mesh`

다음 코드 예시는 `create-mesh`의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 새 서비스 메시를 생성하는 방법

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 포함된 새 서비스 메시를 생성하는 방법

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 서비스 메시를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateMesh를 참조하세요.

create-route

다음 코드 예시는 create-route의 사용 방법을 보여 줍니다.

AWS CLI

새 gRPC 경로를 생성하는 방법

다음 create-route 예제에서는 JSON 입력 파일을 사용하여 gRPC 경로를 생성합니다. 메타데이터가 123으로 시작하는 GRPC 트래픽이 serviceBgrpc라는 이름의 가상 노드로 라우팅됩니다. 이 경로의 대상과 통신을 시도할 때 특정 gRPC, HTTP 또는 TCP 실패가 발생하는 경우 경로가 3회 재시도됩니다. 각 재시도 사이에 15초의 지연이 발생합니다.

```
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 생성하는 방법

다음 `create-route` 예제에서는 JSON 입력 파일을 사용하여 HTTP/2 경로를 생성합니다. HTTP 경로를 생성하려면 http2Route를 사양에 따라 httpRoute로 바꿉니다. 헤더 값이 123으로 시작하는 URL 접두사로 주소 지정된 모든 HTTP/2 트래픽이 serviceBhttp2라는 이름의 가상 노드로 라우팅됩니다. 이 경로의 대상과 통신을 시도할 때 특정 HTTP 또는 TCP 실패가 발생하는 경우 경로가 3회 재시도됩니다. 각 재시도 사이에 15초의 지연이 발생합니다.

```
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 생성하는 방법

다음 `create-route` 예제에서는 JSON 입력 파일을 사용하여 TCP 경로를 생성합니다. 트래픽의 75%는 serviceBtcp라는 이름의 가상 노드로 라우팅되고 트래픽의 25%는 serviceBv2tcp라는 이름의 가상 노드로 라우팅됩니다. 다양한 대상에 대해 다양한 가중치를 지정하는 것은 애플리케이션의 새 버전을 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 콘텐츠:

```
{
    "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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateRoute를 참조하세요.

`create-virtual-gateway`

다음 코드 예시는 `create-virtual-gateway`의 사용 방법을 보여 줍니다.

AWS CLI

새 가상 게이트웨이를 생성하는 방법

다음 `create-virtual-gateway` 예제에서는 JSON 입력 파일을 사용하여, 포트 9080을 사용하는 HTTP에 대한 리스너가 있는 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.json`의 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 가상 게이트웨이를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateVirtualGateway를 참조하세요.

`create-virtual-node`

다음 코드 예시는 `create-virtual-node`의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 검색에 DNS를 사용하는 새 가상 노드를 생성하는 방법

다음 `create-virtual-node` 예제에서는 JSON 입력 파일을 사용하여, 서비스 검색에 DNS를 사용하는 가상 노드를 생성합니다.

```
aws appmesh create-virtual-node \
    --cli-input-json file://create-virtual-node-dns.json
```

`create-virtual-node-dns.json`의 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 노드를 생성하는 방법

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 노드]를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateVirtualNode를 참조하세요.

`create-virtual-router`

다음 코드 예시는 `create-virtual-router`의 사용 방법을 보여 줍니다.

AWS CLI

새 가상 라우터를 생성하는 방법

다음 `create-virtual-router` 예제에서는 JSON 입력 파일을 사용하여, 포트 80을 사용하는 HTTP에 대한 리스너가 있는 가상 라우터를 생성합니다.

```
aws appmesh create-virtual-router \
    --cli-input-json file://create-virtual-router.json
```

`create-virtual-router.json`의 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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateVirtualRouter를 참조하세요.

`create-virtual-service`

다음 코드 예시는 `create-virtual-service`의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 가상 노드 공급자가 있는 새 가상 서비스를 생성하는 방법

다음 `create-virtual-service` 예제에서는 JSON 입력 파일을 사용하여 가상 노드 공급자가 있는 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 공급자가 있는 새 가상 서비스를 생성하는 방법

다음 `create-virtual-service` 예제에서는 JSON 입력 파일을 사용하여 가상 라우터 공급자가 있는 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 가상 서비스<https://docs.aws.amazon.com/app-mesh/latest/userguide/virtual_services.html>를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateVirtualService를 참조하세요.

delete-mesh

다음 코드 예시는 delete-mesh의 사용 방법을 보여 줍니다.

AWS CLI

서비스 메시를 삭제하는 방법

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 서비스 메시를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteMesh를 참조하세요.

delete-route

다음 코드 예시는 delete-route의 사용 방법을 보여 줍니다.

AWS CLI

경로를 삭제하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteRoute를 참조하세요.

delete-virtual-node

다음 코드 예시는 delete-virtual-node의 사용 방법을 보여 줍니다.

AWS CLI

가상 노드를 삭제하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteVirtualNode를 참조하세요.

`delete-virtual-router`

다음 코드 예시는 `delete-virtual-router`의 사용 방법을 보여 줍니다.

AWS CLI

가상 라우터를 삭제하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteVirtualRouter를 참조하세요.

`delete-virtual-service`

다음 코드 예시는 `delete-virtual-service`의 사용 방법을 보여 줍니다.

AWS CLI

가상 서비스를 삭제하는 방법

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 가상 서비스를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteVirtualService를 참조하세요.

describe-mesh

다음 코드 예시는 describe-mesh의 사용 방법을 보여 줍니다.

AWS CLI

서비스 메시를 설명하는 방법

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 서비스 메시를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeMesh를 참조하세요.

describe-route

다음 코드 예시는 describe-route의 사용 방법을 보여 줍니다.

AWS CLI

경로를 설명하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeRoute를 참조하세요.

describe-virtual-node

다음 코드 예시는 `describe-virtual-node`의 사용 방법을 보여 줍니다.

AWS CLI

가상 노드를 설명하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeVirtualNode를 참조하세요.

describe-virtual-router

다음 코드 예시는 describe-virtual-router의 사용 방법을 보여 줍니다.

AWS CLI

가상 라우터를 설명하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeVirtualRouter를 참조하세요.

describe-virtual-service

다음 코드 예시는 describe-virtual-service의 사용 방법을 보여 줍니다.

AWS CLI

가상 서비스를 설명하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeVirtualService를 참조하세요.

`list-meshes`

다음 코드 예시는 `list-meshes`의 사용 방법을 보여 줍니다.

AWS CLI

서비스 메시를 나열하는 방법

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 서비스 메시를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListMeshes를 참조하세요.

list-routes

다음 코드 예시는 list-routes의 사용 방법을 보여 줍니다.

AWS CLI

경로를 나열하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 ListRoutes를 참조하세요.

list-tags-for-resource

다음 코드 예시는 `list-tags-for-resource`의 사용 방법을 보여 줍니다.

AWS CLI

리소스에 대한 태그를 나열하는 방법

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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource 섹션을 참조하세요.

list-virtual-nodes

다음 코드 예시는 `list-virtual-nodes`의 사용 방법을 보여 줍니다.

AWS CLI

가상 노드를 나열하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 ListVirtualNodes를 참조하세요.

`list-virtual-routers`

다음 코드 예시는 `list-virtual-routers`의 사용 방법을 보여 줍니다.

AWS CLI

가상 라우터를 나열하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 ListVirtualRouters를 참조하세요.

list-virtual-services

다음 코드 예시는 list-virtual-services의 사용 방법을 보여 줍니다.

AWS CLI

가상 서비스를 나열하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 ListVirtualServices를 참조하세요.

tag-resource

다음 코드 예시는 tag-resource의 사용 방법을 보여 줍니다.

AWS CLI

리소스에 태그를 지정하는 방법

다음 tag-resource 예제에서는 value1 값을 갖는 태그 key1을 지정된 리소스에 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.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

untag-resource

다음 코드 예시에서는 untag-resource 코드를 사용하는 방법을 보여줍니다.

AWS CLI

리소스에서 태그를 제거하는 방법

다음 untag-resource 예제에서는 지정된 리소스에서 key1 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.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

update-mesh

다음 코드 예시는 update-mesh의 사용 방법을 보여 줍니다.

AWS CLI

서비스 메시를 업데이트하는 방법

다음 update-mesh 예제에서는 JSON 입력 파일을 사용하여, 모든 외부 송신 트래픽이 Envoy 프록시를 통해 변경 없이 그대로 전달되도록 서비스 메시를 업데이트합니다.

```
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 서비스 메시를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateMesh를 참조하세요.

update-route

다음 코드 예시는 update-route의 사용 방법을 보여 줍니다.

AWS CLI

경로를 업데이트하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateRoute를 참조하세요.

update-virtual-node

다음 코드 예시는 update-virtual-node의 사용 방법을 보여 줍니다.

AWS CLI

가상 노드를 업데이트하는 방법

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 노드](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [UpdateVirtualNode](#)를 참조하세요.

update-virtual-router

다음 코드 예시는 update-virtual-router의 사용 방법을 보여 줍니다.

AWS CLI

가상 라우터를 업데이트하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 [UpdateVirtualRouter](https://docs.aws.amazon.com/goto/aws-cli/appmesh-2019-01-25/UpdateVirtualRouter)을 참조하세요.

update-virtual-service

다음 코드 예시는 update-virtual-service의 사용 방법을 보여 줍니다.

AWS CLI

가상 서비스를 업데이트하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateVirtualService를 참조하세요.

AWS CLI를 사용하는 App Runner 예제

다음 코드 예제에서는 App Runner와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`associate-custom-domain`

다음 코드 예시는 `associate-custom-domain`의 사용 방법을 보여 줍니다.

AWS CLI

도메인 이름과 www 하위 도메인을 서비스에 연결하는 방법

다음 `associate-custom-domain` 예제에서는 사용자가 제어하는 사용자 지정 도메인 이름을 App Runner 서비스와 연결합니다. 도메인 이름은 특수 사례 하위 도메인 `www.example.com`을 포함하는 루트 도메인 `example.com`입니다.

```
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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 AssociateCustomDomain을 참조하세요.

`create-auto-scaling-configuration`

다음 코드 예시는 `create-auto-scaling-configuration`의 사용 방법을 보여 줍니다.

AWS CLI

고가용성 오토 스케일링 구성을 생성하는 방법

다음 `create-auto-scaling-configuration` 예제에서는 `MinSize`를 5로 설정하여 고가용성에 최적화된 오토 스케일링 구성을 생성합니다. 이 구성을 사용하면 App Runner는 AWS 리전에 따라 최대 다섯 개까지 가장 가능한 가용 영역에 서비스 인스턴스를 분산하려고 시도합니다.

이 호출은 `AutoScalingConfiguration` 객체를 반환하며 다른 설정들은 그 각자의 기본값으로 설정됩니다. 이 예제에서는 이 호출이 `high-availability`라는 이름의 구성을 생성하기 위한 첫 번째 호출입니다. 개정은 1로 설정되며 이는 최신 개정입니다.

```
aws apprunner create-auto-scaling-configuration \
    --cli-input-json file://input.json
```

`input.json`의 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
        }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateAutoScalingConfiguration을 참조하세요.

`create-connection`

다음 코드 예시는 `create-connection`의 사용 방법을 보여 줍니다.

AWS CLI

GitHub 연결을 생성하는 방법

다음 `create-connection` 예제에서는 프라이빗 GitHub 코드 리포지토리에 대한 연결을 생성합니다. 성공적인 호출 후 연결 상태는 `PENDING_HANDSHAKE`입니다. 이는 공급자와의 인증 핸드셰이크가 아직 발생하지 않았기 때문입니다. App Runner 콘솔을 사용하여 핸드셰이크를 완료합니다.

```
aws apprunner create-connection \
    --cli-input-json file://input.json
```

`input.json`의 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 AWS App Runner 개발자 안내서의 App Runner 연결 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateConnection을 참조하세요.

create-service

다음 코드 예시는 create-service의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 소스 코드 리포지토리 서비스를 생성하는 방법

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 서비스를 생성하는 방법

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 서비스를 생성하는 방법

다음 `create-service` 예제에서는 Elastic Container Registry(ECR)에 저장된 이미지를 기반으로 App Runner 서비스를 생성합니다.

```
aws apprunner create-service \
    --cli-input-json file://input.json
```

input.json의 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"
        }
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateService를 참조하세요.

`delete-auto-scaling-configuration`

다음 코드 예시는 `delete-auto-scaling-configuration`의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 오토 스케일링 구성의 최신 활성 개정을 삭제하는 방법

다음 `delete-auto-scaling-configuration` 예제에서는 App Runner 오토 스케일링 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 `"Latest": false`가 표시됩니다. 삭제된 후에는 더 이상 최신 활성 개정이 아니기 때문입니다.

```
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"
}
```

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: 오토 스케일링 구성의 특정 개정을 삭제하는 방법

다음 `delete-auto-scaling-configuration` 예제에서는 App Runner 오토 스케일링 구성의 특정 개정을 삭제합니다. 특정 개정을 삭제하려면 개정 번호가 포함된 ARN을 지정합니다.

이 예제에서는 이 작업 앞에 여러 개정이 있습니다. 이 작업은 개정 `1`을 삭제합니다.

```
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
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteAutoScalingConfiguration을 참조하세요.

`delete-connection`

다음 코드 예시는 `delete-connection`의 사용 방법을 보여 줍니다.

AWS CLI

연결을 삭제하는 방법

다음 `delete-connection` 예제에서는 App Runner 연결을 삭제합니다. 성공적인 호출 후 연결 상태는 `DELETED`입니다. 이는 연결을 더 이상 사용할 수 없기 때문입니다.

```
aws apprunner delete-connection \
    --cli-input-json file://input.json
```

`input.json`의 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteConnection 섹션을 참조하세요.

delete-service

다음 코드 예시는 delete-service의 사용 방법을 보여 줍니다.

AWS CLI

서비스 삭제

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"
        }
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteService를 참조하세요.

describe-auto-scaling-configuration

다음 코드 예시는 describe-auto-scaling-configuration의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 오토 스케일링 구성의 최신 활성 개정을 설명하는 방법

다음 describe-auto-scaling-configuration 예제에서는 App Runner 오토 스케일링 구성의 최신 활성 개정의 설명을 가져옵니다. 최신 활성 개정을 설명하려면 개정 구성 요소 없이 구성 이름으로 끝나는 ARN을 지정합니다.

이 예제에서는 두 가지 개정이 있습니다. 따라서 개정 2(최신)가 설명됩니다. 결과 객체는 "Latest": true를 표시합니다.

```
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: 오토 스케일링 구성의 특정 개정을 설명하는 방법

다음 `describe-auto-scaling-configuration` 예제에서는 App Runner 오토 스케일링 구성의 특정 개정의 설명을 가져옵니다. 특정 개정을 설명하려면 개정 번호가 포함된 ARN을 지정합니다.

이 예제에서는 여러 개정이 존재하고 개정 `1`이 쿼리됩니다. 결과 객체는 `"Latest": false`를 표시합니다.

```
aws apprunner describe-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",
        "Latest": false,
        "Status": "ACTIVE",
        "MaxConcurrency": 100,
        "MaxSize": 50,
```

```
        "MinSize": 5
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeAutoScalingConfiguration을 참조하세요.

describe-custom-domains

다음 코드 예시는 describe-custom-domains의 사용 방법을 보여 줍니다.

AWS CLI

서비스와 연결된 사용자 지정 도메인 이름의 설명을 가져오는 방법

다음 describe-custom-domains 예제에서는 App Runner 서비스와 연결된 사용자 지정 도메인 이름의 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeCustomDomains](#)를 참조하세요.

describe-service

다음 코드 예시는 `describe-service`의 사용 방법을 보여 줍니다.

AWS CLI

서비스를 설명하는 방법

다음 `describe-service` 예제에서는 App Runner 서비스의 설명을 가져옵니다.

```
aws apprunner describe-service \
    --cli-input-json file://input.json
```

input.json의 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"
        }
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeService를 참조하세요.

`disassociate-custom-domain`

다음 코드 예시는 `disassociate-custom-domain`의 사용 방법을 보여 줍니다.

AWS CLI

서비스에서 도메인 이름의 연결을 해제하는 방법

다음 `disassociate-custom-domain` 예제에서는 App Runner 서비스에서 도메인 `example.com`의 연결을 해제합니다. 이 호출은 루트 도메인과 함께 연결된 하위 도메인 `www.example.com`의 연결도 해제합니다.

```
aws apprunner disassociate-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"
}
```

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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DisassociateCustomDomain을 참조하세요.

list-auto-scaling-configurations

다음 코드 예시는 `list-auto-scaling-configurations`의 사용 방법을 보여 줍니다.

AWS CLI

App Runner 오토 스케일링 구성의 페이지 매긴 목록을 가져오는 방법

다음 `list-auto-scaling-configurations` 예제에서는 AWS 계정에서 모든 App Runner 오토 스케일링 구성을 나열합니다. 각 응답에서 오토 스케일링 구성이 최대 다섯 개까지 나열됩니다. `AutoScalingConfigurationName` 및 `LatestOnly`는 지정되지 않습니다. 그 기본값들로 인해 모든 활성 구성의 최신 개정이 나열됩니다.

이 예제에서는 응답에 두 개의 결과가 포함되며 추가 결과가 없으므로 NextToken이 반환되지 않습니다.

```
aws apprunner list-auto-scaling-configurations \
    --cli-input-json file://input.json
```

input.json의 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
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListAutoScalingConfigurations를 참조하세요.

list-connections

다음 코드 예시는 list-connections의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 모든 연결을 나열하는 방법

다음 `list-connections` 예제에서는 AWS 계정에서 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: 연결을 이름별로 나열하는 방법

다음 `list-connections` 예제에서는 연결을 이름별로 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListConnections를 참조하세요.

`list-operations`

다음 코드 예시는 `list-operations`의 사용 방법을 보여 줍니다.

AWS CLI

서비스에서 발생한 작업을 나열하는 방법

다음 `list-operations` 예제에서는 지금까지 App Runner 서비스에서 발생한 모든 작업을 나열합니다. 이 예제에서는 서비스가 새 서비스이며 단일 작업 유형인 CREATE_SERVICE만 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
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [ListOperations](#) 섹션을 참조하세요.

`list-services`

다음 코드 예시는 `list-services`의 사용 방법을 보여 줍니다.

AWS CLI

App Runner 서비스의 페이지 매긴 목록을 가져오는 방법

다음 `list-services` 예제에서는 AWS 계정에서 모든 App Runner 서비스를 나열합니다. 각 응답에 서비스가 최대 두 개까지 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.json`의 콘텐츠:

```
{
    "MaxResults": 2
}
```

출력:

```
{
```

```
    "NextToken":
 "eyJDdXN0b21lckFjY291bnRJZCI6IjI3MDIwNTQwMjg0NSIsIlNlcnZpY2VTdGF0dXNDb2RlIjoiUFJPVklTSU9OSU
    "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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListServices를 참조하세요.

list-tags-for-resource

다음 코드 예시는 `list-tags-for-resource`의 사용 방법을 보여 줍니다.

AWS CLI

App Runner 서비스와 연결된 태그를 나열하는 방법

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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource 섹션을 참조하세요.

pause-service

다음 코드 예시는 pause-service의 사용 방법을 보여 줍니다.

AWS CLI

서비스를 일시 중지하는 방법

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"
        }
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 PauseService를 참조하세요.

resume-service

다음 코드 예시는 resume-service의 사용 방법을 보여 줍니다.

AWS CLI

서비스를 재개하는 방법

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"
        }
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ResumeService를 참조하세요.

start-deployment

다음 코드 예시는 `start-deployment`의 사용 방법을 보여 줍니다.

AWS CLI

수동 배포를 시작하는 방법

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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 StartDeployment를 참조하세요.

`tag-resource`

다음 코드 예시는 `tag-resource`의 사용 방법을 보여 줍니다.

AWS CLI

App Runner 서비스에 태그를 추가하는 방법

다음 `tag-resource` 예제에서는 App Runner 서비스에 두 개의 태그를 추가합니다.

```
aws apprunner tag-resource \
    --cli-input-json file://input.json
```

`input.json`의 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.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

untag-resource

다음 코드 예시는 untag-resource의 사용 방법을 보여 줍니다.

AWS CLI

App Runner 서비스에서 태그를 제거하는 방법

다음 untag-resource 예제에서는 App Runner 서비스에서 두 개의 태그를 제거합니다.

```
aws apprunner untag-resource \
    --cli-input-json file://input.json
```

input.json의 콘텐츠:

```
{
    "ResourceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa",
    "TagKeys": [
        "Department",
        "CustomerId"
    ]
}
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

`update-service`

다음 코드 예시는 `update-service`의 사용 방법을 보여 줍니다.

AWS CLI

메모리 크기를 업데이트하는 방법

다음 `update-service` 예제에서는 App Runner 서비스 인스턴스(스케일링 단위)의 메모리 크기를 2048MiB 로 업데이트합니다.

호출이 성공하면 App Runner는 비동기 업데이트 프로세스를 시작합니다. 호출에 의해 반환되는 `Service` 구조는 이 호출이 적용하는 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"
        }
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateService를 참조하세요.

AWS CLI를 사용한 AWS AppConfig 예시

다음 코드 예시에서는 AWS AppConfig에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

작업

`create-application`

다음 코드 예시는 `create-application`의 사용 방법을 보여 줍니다.

AWS CLI

애플리케이션을 생성하는 방법

다음 `create-application` 예제에서는 AWS AppConfig에서 애플리케이션을 생성합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateApplication을 참조하세요.

`create-configuration-profile`

다음 코드 예시는 `create-configuration-profile`의 사용 방법을 보여 줍니다.

AWS CLI

구성 프로필을 생성하는 방법

다음 `create-configuration-profile` 예제에서는 Systems Manager의 기능인 Parameter Store에 저장된 구성을 사용하여 구성 프로필을 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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateConfigurationProfile을 참조하세요.

`create-environment`

다음 코드 예시는 `create-environment`의 사용 방법을 보여 줍니다.

AWS CLI

환경을 생성하는 방법

다음 `create-environment` 예제에서는 create-application을 사용하여 생성한 애플리케이션을 사용해 Example-Environment라는 이름의 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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateEnvironment를 참조하세요.

create-extension-association

다음 코드 예시는 create-extension-association의 사용 방법을 보여 줍니다.

AWS CLI

확장 연결을 생성하는 방법

다음 create-extension-association 예제에서는 AWS AppConfig에서 새 확장 연결을 생성합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateExtensionAssociation을 참조하세요.

create-extension

다음 코드 예시는 `create-extension`의 사용 방법을 보여 줍니다.

AWS CLI

확장을 생성하는 방법

다음 `create-extension` 예제에서는 AWS AppConfig에서 새 확장을 생성합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateExtension을 참조하세요.

create-hosted-configuration-version

다음 코드 예시는 create-hosted-configuration-version의 사용 방법을 보여 줍니다.

AWS CLI

호스팅 구성 버전을 생성하는 방법

다음 create-hosted-configuration-version 예제에서는 AWS AppConfig 호스팅 구성 스토어에서 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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateHostedConfigurationVersion을 참조하세요.

delete-application

다음 코드 예시는 delete-application의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteApplication을 참조하세요.

delete-configuration-profile

다음 코드 예시는 delete-configuration-profile의 사용 방법을 보여 줍니다.

AWS CLI

구성 프로필을 삭제하는 방법

다음 delete-configuration-profile 예제에서는 지정된 구성 프로필을 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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteConfigurationProfile을 참조하세요.

delete-deployment-strategy

다음 코드 예시는 delete-deployment-strategy의 사용 방법을 보여 줍니다.

AWS CLI

배포 전략을 삭제하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteDeploymentStrategy를 참조하세요.

delete-environment

다음 코드 예시는 delete-environment의 사용 방법을 보여 줍니다.

AWS CLI

환경을 삭제하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteEnvironment를 참조하세요.

delete-extension-association

다음 코드 예시는 delete-extension-association의 사용 방법을 보여 줍니다.

AWS CLI

확장 연결을 삭제하는 방법

다음 delete-extension-association 예제에서는 AWS AppConfig에서 확장 연결을 삭제합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteExtensionAssociation을 참조하세요.

delete-extension

다음 코드 예시는 delete-extension의 사용 방법을 보여 줍니다.

AWS CLI

확장을 삭제하는 방법

다음 delete-extension 예제에서는 AWS AppConfig 에서 확장을 삭제합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteExtension을 참조하세요.

delete-hosted-configuration-version

다음 코드 예시는 delete-hosted-configuration-version의 사용 방법을 보여 줍니다.

AWS CLI

호스팅 구성 버전을 삭제하는 방법

다음 delete-hosted-configuration-version 예제에서는 AWS AppConfig 호스팅 구성 스토어에서 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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteHostedConfigurationVersion을 참조하세요.

get-application

다음 코드 예시는 get-application의 사용 방법을 보여 줍니다.

AWS CLI

애플리케이션의 세부 정보를 나열하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 GetApplication을 참조하세요.

get-configuration-profile

다음 코드 예시는 get-configuration-profile의 사용 방법을 보여 줍니다.

AWS CLI

구성 프로필 세부 정보를 검색하는 방법

다음 get-configuration-profile 예제에서는 지정된 구성 프로필의 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.

- API 세부 정보는 AWS CLI 명령 참조의 GetConfigurationProfile을 참조하세요.

get-configuration

다음 코드 예시는 get-configuration의 사용 방법을 보여 줍니다.

AWS CLI

구성 세부 정보를 검색하는 방법

다음 get-configuration 예제에서는 예제 애플리케이션의 구성 세부 정보를 반환합니다. 이후의 get-configuration에 대한 호출에서는 버전이 변경된 경우에만 애플리케이션의 구성을 업데이트

하도록 `client-configuration-version` 파라미터를 사용합니다. 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-file`의 콘텐츠:

```
{ "Name": "ExampleApplication", "Id": ExampleID, "Rank": 7 }
```

출력:

```
{
    "ConfigurationVersion": "1",
    "ContentType": "application/json"
}
```

자세한 내용은 AWS AppConfig 사용 설명서의 6단계: 구성 수신을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetConfiguration을 참조하세요.

`get-deployment-strategy`

다음 코드 예시는 `get-deployment-strategy`의 사용 방법을 보여 줍니다.

AWS CLI

배포 전략의 세부 정보를 검색하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 GetDeploymentStrategy를 참조하세요.

get-deployment

다음 코드 예시는 get-deployment의 사용 방법을 보여 줍니다.

AWS CLI

배포 세부 정보를 검색하는 방법

다음 get-deployment 예제에서는 지정된 환경 및 배포의 애플리케이션에 대한 배포의 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.

- API 세부 정보는 AWS CLI 명령 참조의 GetDeployment를 참조하세요.

get-environment

다음 코드 예시는 get-environment의 사용 방법을 보여 줍니다.

AWS CLI

환경 세부 정보를 검색하는 방법

다음 get-environment 예제에서는 지정된 환경의 세부 정보 및 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.

- API 세부 정보는 AWS CLI 명령 참조의 GetEnvironment를 참조하세요.

get-extension-association

다음 코드 예시는 get-extension-association의 사용 방법을 보여 줍니다.

AWS CLI

확장 연결 세부 정보를 가져오는 방법

다음 `get-extension-association` 예제에서는 확장 연결에 대한 정보를 표시합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 GetExtensionAssociation을 참조하세요.

`get-extension`

다음 코드 예시는 `get-extension`의 사용 방법을 보여 줍니다.

AWS CLI

확장 세부 정보를 가져오는 방법

다음 `get-extension` 예제에서는 확장에 대한 정보를 표시합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 GetExtension을 참조하세요.

get-hosted-configuration-version

다음 코드 예시는 get-hosted-configuration-version의 사용 방법을 보여 줍니다.

AWS CLI

호스팅 구성 세부 정보를 검색하는 방법

다음 get-hosted-configuration-version 예제에서는 AWS AppConfig 호스팅 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.

- API 세부 정보는 AWS CLI 명령 참조의 GetHostedConfigurationVersion을 참조하세요.

list-applications

다음 코드 예시는 list-applications의 사용 방법을 보여 줍니다.

AWS CLI

사용 가능한 애플리케이션을 나열하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 ListApplications를 참조하세요.

list-configuration-profiles

다음 코드 예시는 list-configuration-profiles의 사용 방법을 보여 줍니다.

AWS CLI

사용 가능한 구성 프로필을 나열하는 방법

다음 list-configuration-profiles 예제에서는 지정된 애플리케이션에 대해 사용 가능한 구성 프로필을 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListConfigurationProfiles를 참조하세요.

`list-deployment-strategies`

다음 코드 예시는 `list-deployment-strategies`의 사용 방법을 보여 줍니다.

AWS CLI

사용 가능한 배포 전략을 나열하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 ListDeploymentStrategies를 참조하세요.

`list-deployments`

다음 코드 예시는 `list-deployments`의 사용 방법을 보여 줍니다.

AWS CLI

사용 가능한 배포를 나열하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 ListDeployments를 참조하세요.

list-environments

다음 코드 예시는 `list-environments`의 사용 방법을 보여 줍니다.

AWS CLI

사용 가능한 환경을 나열하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 ListEnvironments를 참조하세요.

list-extension-associations

다음 코드 예시는 `list-extension-associations`의 사용 방법을 보여 줍니다.

AWS CLI

AWS 리전에 대한 AWS 계정 내 모든 AWS AppConfig 확장 연결을 나열하는 방법

다음 `list-extension-associations` 예제에서는 특정 AWS 리전에서 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListExtensionAssociations를 참조하세요.

list-extensions

다음 코드 예시는 `list-extensions`의 사용 방법을 보여 줍니다.

AWS CLI

AWS 리전에 대한 AWS 계정 내 모든 AWS AppConfig 확장을 나열하는 방법

다음 `list-extensions` 예제에서는 특정 AWS 리전에서 현재 AWS 계정에 대한 모든 AWS AppConfig 확장을 나열합니다. 이 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListExtensions를 참조하세요.

`list-hosted-configuration-versions`

다음 코드 예시는 `list-hosted-configuration-versions`의 사용 방법을 보여 줍니다.

AWS CLI

사용 가능한 호스팅 구성 버전을 나열하는 방법

다음 `list-hosted-configuration-versions` 예제에서는 지정된 애플리케이션 및 구성 프로필에 대해 AWS AppConfig 호스팅 구성 스토어에서 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListHostedConfigurationVersions를 참조하세요.

`list-tags-for-resource`

다음 코드 예시는 `list-tags-for-resource`의 사용 방법을 보여 줍니다.

AWS CLI

애플리케이션의 태그를 나열하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource를 참조하세요.

start-deployment

다음 코드 예시는 `start-deployment`의 사용 방법을 보여 줍니다.

AWS CLI

구성 배포를 시작하는 방법

다음 `start-deployment` 예제에서는 지정된 환경, 배포 전략 및 구성 프로필을 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.

- API 세부 정보는 AWS CLI 명령 참조의 StartDeployment를 참조하세요.

stop-deployment

다음 코드 예시는 `stop-deployment`의 사용 방법을 보여 줍니다.

AWS CLI

구성 배포를 중지하는 방법

다음 `stop-deployment` 예제에서는 지정된 환경에 대한 애플리케이션 구성 배포를 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.

- API 세부 정보는 AWS CLI 명령 참조의 StopDeployment를 참조하세요.

tag-resource

다음 코드 예시는 tag-resource의 사용 방법을 보여 줍니다.

AWS CLI

애플리케이션에 태그를 지정하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

untag-resource

다음 코드 예시는 untag-resource의 사용 방법을 보여 줍니다.

AWS CLI

애플리케이션에서 태그를 제거하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

update-application

다음 코드 예시는 update-application의 사용 방법을 보여 줍니다.

AWS CLI

애플리케이션을 업데이트하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateApplication을 참조하세요.

update-configuration-profile

다음 코드 예시는 update-configuration-profile의 사용 방법을 보여 줍니다.

AWS CLI

구성 프로필을 업데이트하는 방법

다음 update-configuration-profile 예제에서는 지정된 구성 프로필의 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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateConfigurationProfile을 참조하세요.

update-deployment-strategy

다음 코드 예시는 `update-deployment-strategy`의 사용 방법을 보여 줍니다.

AWS CLI

배포 전략을 업데이트하는 방법

다음 `update-deployment-strategy` 예제에서는 지정된 배포 전략에서 최종 베이크 소요 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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateDeploymentStrategy를 참조하세요.

update-environment

다음 코드 예시는 update-environment의 사용 방법을 보여 줍니다.

AWS CLI

환경을 업데이트하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateEnvironment를 참조하세요.

update-extension-association

다음 코드 예시는 update-extension-association의 사용 방법을 보여 줍니다.

AWS CLI

AWS AppConfig 확장 연결을 업데이트하는 방법

다음 update-extension-association 예제에서는 AWS AppConfig에서 확장 연결에 새 파라미터 값을 추가합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateExtensionAssociation을 참조하세요.

`update-extension`

다음 코드 예시는 `update-extension`의 사용 방법을 보여 줍니다.

AWS CLI

AWS AppConfig 확장을 업데이트하는 방법

다음 `update-extension` 예제에서는 AWS AppConfig에서 확장에 추가 파라미터 키를 추가합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateExtension을 참조하세요.

validate-configuration

다음 코드 예시는 validate-configuration의 사용 방법을 보여 줍니다.

AWS CLI

구성을 검증하는 방법

다음 validate-configuration 예제에서는 구성 프로필에서 유효성 검사기를 사용하여 구성을 검증합니다.

```
aws appconfig validate-configuration \
    --application-id abc1234 \
    --configuration-profile-id ur8hx2f \
    --configuration-version 1
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS AppConfig 사용 설명서의 3단계: 구성 및 구성 프로필 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ValidateConfiguration을 참조하세요.

AWS CLI를 사용하는 Application Auto Scaling 예제

다음 코드 예제에서는 Application Auto Scaling과 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

delete-scaling-policy

다음 코드 예시는 `delete-scaling-policy`의 사용 방법을 보여 줍니다.

AWS CLI

조정 정책을 삭제하는 방법

이 예제에서는 기본 클러스터에서 실행되는 Amazon ECS 서비스 웹앱에 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
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteScalingPolicy를 참조하세요.

`delete-scheduled-action`

다음 코드 예시는 `delete-scheduled-action`의 사용 방법을 보여 줍니다.

AWS CLI

예약된 작업을 삭제하는 방법

다음 `delete-scheduled-action` 예제에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteScheduledAction을 참조하세요.

`deregister-scalable-target`

다음 코드 예시는 `deregister-scalable-target`의 사용 방법을 보여 줍니다.

AWS CLI

확장 가능 대상의 등록을 취소하는 방법

이 예제에서는 기본 클러스터에서 실행 중인 웹앱이라고 하는 Amazon ECS 서비스에 대한 확장 가능 대상의 등록을 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 리소스 ID를 식별하는 문자열이 포함되어 있으며 이는 사용자 지정 리소스의 경우 Amazon API Gateway 엔드포인트를 통하는 사용자 지정 리소스에 대한 경로입니다.

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 콘텐츠:

```
https://example.execute-api.us-west-2.amazonaws.com/prod/
scalableTargetDimensions/1-23456789
```

- API 세부 정보는 AWS CLI 명령 참조의 DeregisterScalableTarget을 참조하세요.

describe-scalable-targets

다음 코드 예시는 describe-scalable-targets의 사용 방법을 보여 줍니다.

AWS CLI

확장 가능 대상을 설명하는 방법

다음 describe-scalable-targets 예제에서는 ecs 서비스 네임스페이스에 대한 확장 가능 대상을 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 Application Auto Scaling 사용 설명서의 Application Auto Scaling과 함께 사용할 수 있는 AWS 서비스를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeScalableTargets를 참조하세요.

describe-scaling-activities

다음 코드 예시는 describe-scaling-activities의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 지정된 Amazon ECS 서비스에 대한 스케일링 활동을 설명하는 방법

다음 describe-scaling-activities 예제에서는 default 클러스터에서 실행 중인 web-app이라고 하는 Amazon ECS 서비스에 대한 스케일링 활동을 설명합니다. 출력은 조정 정책에 의해 시작되는 스케일링 활동을 표시합니다.

```
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 스케일링 활동을 참조하세요.

예제 2: 지정된 DynamoDB 테이블에 대한 스케일링 활동을 설명하는 방법

다음 `describe-scaling-activities` 예제에서는 `TestTable`이라고 하는 DynamoDB에 대한 스케일링 활동을 설명합니다. 출력은 두 가지 다른 예약된 작업에 의해 시작되는 스케일링 활동을 표시합니다.

```
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 스케일링 활동을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeScalingActivities를 참조하세요.

describe-scaling-policies

다음 코드 예시는 describe-scaling-policies의 사용 방법을 보여 줍니다.

AWS CLI

조정 정책을 설명하는 방법

이 예제 명령은 ECS 서비스 네임스페이스에 대한 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeScalingPolicies를 참조하세요.

describe-scheduled-actions

다음 코드 예시는 describe-scheduled-actions의 사용 방법을 보여 줍니다.

AWS CLI

예약된 작업을 설명하는 방법

다음 describe-scheduled-actions 예제에서는 지정된 서비스 네임스페이스에 대한 예약된 작업의 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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeScheduledActions 섹션을 참조하세요.

list-tags-for-resource

다음 코드 예시는 list-tags-for-resource의 사용 방법을 보여 줍니다.

AWS CLI

확장 가능 대상에 대한 태그를 나열하는 방법

다음 list-tags-for-resource 예제에서는 ARN에 의해 지정된 확장 가능 대상에 연결된 태그 키 이름과 값을 나열합니다.

```
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 태그 지정 지원을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource를 참조하세요.

put-scaling-policy

다음 코드 예시는 put-scaling-policy의 사용 방법을 보여 줍니다.

AWS CLI

예 1: 사전 정의된 지표 사양을 사용하여 대상 추적 조정 정책 적용

다음 `put-scaling-policy` 예제에서는 기본 클러스터에서 미리 정의된 지표 사양이 있는 대상 추적 조정 정책을 웹앱이라고 하는 Amazon ECS 서비스에 적용합니다. 이 정책은 60초의 스케일 아웃 및 스케일 인 휴지 기간을 사용하여 서비스의 평균 CPU 사용률을 75%로 유지합니다. 출력에는 ARN과 사용자를 대신하여 생성된 두 개의 CloudWatch 경보 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 콘텐츠가 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 기본 클러스터에서 사용자 지정 지표 사양이 있는 대상 추적 조정 정책을 웹앱이라고 하는 Amazon ECS 서비스에 적용합니다. 이 정책은 60초의 스케일 아웃 및 스케일 인 휴지 기간을 사용하여 서비스의 평균 사용률을 75%로 유지합니다. 출력에는 ARN과 사용자를 대신하여 생성된 두 개의 CloudWatch 경보 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 콘텐츠가 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 기본 클러스터에서 대상 추적 조정 정책을 web-app이라고 하는 Amazon ECS 서비스에 적용합니다. 정책은 Application Load Balancer의 `RequestCountPerTarget` 지표가 임계값을 초과할 때 ECS 서비스를 스케일 아웃하는 데 사용됩니다. 출력에는 ARN과 사용자를 대신하여 생성된 CloudWatch 경보 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 AWS Application Auto Scaling 사용 설명서의 Application Auto Scaling에 대한 대상 추적 조정 정책을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutScalingPolicy를 참조하세요.

put-scheduled-action

다음 코드 예시는 `put-scheduled-action`의 사용 방법을 보여 줍니다.

AWS CLI

DynamoDB 테이블에 예약된 작업을 추가하는 방법

이 예제에서는 TestTable이라고 하는 DynamoDB 테이블에 예약된 작업을 추가하여 반복 일정으로 스케일 아웃합니다. 지정된 일정(UTC 기준 매일 오후 12시 15분)에서 현재 용량이 MinCapacity에

대해 지정된 값보다 작은 경우 Application Auto Scaling은 MinCapacity에 의해 지정된 값으로 스케일 아웃됩니다.

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.

- API 세부 정보는 AWS CLI 명령 참조의 PutScheduledAction을 참조하세요.

register-scalable-target

다음 코드 예시는 `register-scalable-target`의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: ECS 서비스를 확장 가능 대상으로 등록하는 방법

다음 `register-scalable-target` 예제에서는 Application Auto Scaling에 Amazon ECS 서비스를 등록합니다. 또한 키 이름 `environment` 및 `production` 값을 갖는 태그를 확장 가능 대상에 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 Application Auto Scaling 사용 설명서의 Application Auto Scaling과 함께 사용할 수 있는 AWS 서비스 주제를 참조하세요.

예제 2: 확장 가능 대상에 대한 스케일링 활동을 중단하는 방법

다음 `register-scalable-target` 예제에서는 기존 확장 가능 대상에 대한 조정 활동을 중단합니다.

```
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 스케일링 중단 및 재개](#)를 참조하세요.

예제 3: 확장 가능 대상에 대한 스케일링 활동을 재개하는 방법

다음 `register-scalable-target` 예제에서는 기존 확장 가능 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 스케일링 중단 및 재개를 참조하세요.

- 자세한 내용은 AWS CLI 명령 참조의 RegisterScalableTarget을 참조하세요.

`tag-resource`

다음 코드 예시는 `tag-resource`의 사용 방법을 보여 줍니다.

AWS CLI

확장 가능 대상에 태그를 추가하는 방법

다음 `tag-resource` 예제에서는 키 이름 `environment` 및 `production` 값을 갖는 태그를 ARN에 의해 지정된 확장 가능 대상에 추가합니다.

```
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 태그 지정 지원을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

`untag-resource`

다음 코드 예시는 `untag-resource`의 사용 방법을 보여 줍니다.

AWS CLI

확장 가능 대상에서 태그를 제거하는 방법

다음 `untag-resource` 예제에서는 ARN에 의해 지정된 확장 가능 대상에서 키 이름 `environment`를 갖는 태그 페어를 제거합니다.

```
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 태그 지정 지원을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

AWS CLI를 사용하는 Application Discovery Service 예제

다음 코드 예제에서는 Application Discovery Service와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

작업

describe-agents

다음 코드 예시는 describe-agents의 사용 방법을 보여 줍니다.

AWS CLI

지정된 collectionStatus 상태가 있는 에이전트 설명

이 예제 명령은 수집 상태가 “STARTED” 또는 “STOPPED”인 수집 에이전트를 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeAgents를 참조하세요.

describe-configurations

다음 코드 예시는 describe-configurations의 사용 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"
          }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeConfigurations를 참조하세요.

`list-configurations`

다음 코드 예시는 `list-configurations`의 사용 방법을 보여 줍니다.

AWS CLI

필터 조건 집합을 충족하는 검색된 모든 서버를 나열하는 방법

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"
            }
        ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListConfigurations를 참조하세요.

AWS CLI를 사용한 AppRegistry 예시

다음 코드 예시는 AppRegistry와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

associate-attribute-group

다음 코드 예시에서는 `associate-attribute-group`의 사용 방법을 보여줍니다.

AWS CLI

속성 그룹 연결

다음 `associate-attribute-group` 예시에서는 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 Service Catalog AppRegistry 관리자 안내서의 속성 그룹 연결 및 연결 해제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AssociateAttributeGroup을 참조하세요.

`create-application`

다음 코드 예시에서는 `create-application`의 사용 방법을 보여줍니다.

AWS CLI

애플리케이션 생성

다음 `create-application` 예시에서는 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 Service Catalog AppRegistry 관리자 안내서의 애플리케이션 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateApplication을 참조하세요.

`create-attribute-group`

다음 코드 예시에서는 `create-attribute-group`의 사용 방법을 보여줍니다.

AWS CLI

속성 그룹 생성

다음 `create-attribute-group` 예시에서는 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 Service Catalog AppRegistry 관리자 안내서의 속성 그룹 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [CreateAttributeGroup](https://docs.aws.amazon.com/goto/aws-cli/servicecatalog-appregistry-2020-06-24/CreateAttributeGroup)을 참조하세요.

delete-application

다음 코드 예시에서는 `delete-application`의 사용 방법을 보여줍니다.

AWS CLI

애플리케이션 삭제

다음 `delete-application` 예시에서는 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 Service Catalog AppRegistry 관리자 안내서의 [애플리케이션 관리](https://docs.aws.amazon.com/servicecatalog/latest/arguide/apps.html)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteApplication](https://docs.aws.amazon.com/goto/aws-cli/servicecatalog-appregistry-2020-06-24/DeleteApplication)을 참조하세요.

delete-attribute-group

다음 코드 예시에서는 `delete-attribute-group`의 사용 방법을 보여줍니다.

AWS CLI

예시 8: 속성 그룹 삭제

다음 `delete-attribute-group` 예시에서는 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 Service Catalog AppRegistry 관리자 안내서의 속성 그룹 삭제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteAttributeGroup을 참조하세요.

get-application

다음 코드 예시에서는 `get-application`의 사용 방법을 보여줍니다.

AWS CLI

애플리케이션 가져오기

다음 `get-application` 예시에서는 AWS 계정의 특정 애플리케이션에 대한 메타데이터 정보를 가져옵니다.

```
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 Service Catalog AppRegistry 관리자 안내서의 애플리케이션 세부 정보 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetApplication을 참조하세요.

get-attribute-group

다음 코드 예시에서는 `get-attribute-group`의 사용 방법을 보여줍니다.

AWS CLI

속성 그룹 가져오기

다음 `get-attribute-group` 예시에서는 AWS 계정의 특정 속성 그룹을 가져옵니다.

```
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 Service Catalog AppRegistry 관리자 안내서의 [속성 그룹의 메타데이터 관리]를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [GetAttributeGroup]을 참조하세요.

list-applications

다음 코드 예시에서는 `list-applications`의 사용 방법을 보여줍니다.

AWS CLI

애플리케이션 나열

다음 `list-applications` 예시에서는 AWS 계정의 모든 애플리케이션 목록을 가져옵니다.

```
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 Service Catalog AppRegistry 관리자 안내서의 애플리케이션 세부 정보 보기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListApplications를 참조하세요.

list-associated-attribute-groups

다음 코드 예시에서는 `list-associated-attribute-groups`의 사용 방법을 보여줍니다.

AWS CLI

연결된 속성 그룹 나열

다음 `list-associated-attribute-groups` 예시에서는 AWS 계정의 특정 애플리케이션과 연결된 AWS 계정의 모든 속성 그룹 목록을 가져옵니다.

```
aws servicecatalog-appregistry list-associated-attribute-groups \
    --application "ExampleApplication"
```

출력:

```
{
    "attributeGroups": [
        "01sj5xdwhbw54kejwnt09fnpcl"
    ]
```

```
}
```

자세한 내용은 AWS Service Catalog AppRegistry 관리자 안내서의 [속성 그룹 연결 및 연결 해제]를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ListAssociatedAttributeGroups]를 참조하세요.

list-attribute-groups-for-application

다음 코드 예시에서는 `list-attribute-groups-for-application`의 사용 방법을 보여줍니다.

AWS CLI

애플리케이션의 속성 그룹 나열

다음 `list-attribute-groups-for-application` 예시에서는 AWS 계정의 특정 애플리케이션에 연결된 AWS 계정의 모든 속성 그룹에 대한 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 보기]를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ListAttributeGroupsForApplication]을 참조하세요.

list-attribute-groups

다음 코드 예시에서는 `list-attribute-groups`의 사용 방법을 보여줍니다.

AWS CLI

속성 그룹 나열

다음 `list-attribute-groups` 예시에서는 AWS 계정의 모든 속성 그룹 목록을 가져옵니다.

```
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 보기](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListAttributeGroups를 참조하세요.

update-application

다음 코드 예시에서는 update-application의 사용 방법을 보여줍니다.

AWS CLI

애플리케이션 업데이트

다음 update-application 예시에서는 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 Service Catalog AppRegistry 관리자 안내서의 애플리케이션 편집을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateApplication을 참조하세요.

update-attribute-group

다음 코드 예시에서는 update-attribute-group의 사용 방법을 보여줍니다.

AWS CLI

속성 그룹 업데이트

다음 update-attribute-group 예시에서는 AWS 계정의 특정 속성 그룹을 업데이트하여 설명을 포함합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateAttributeGroup을 참조하세요.

AWS CLI를 사용하는 Athena 예제

다음 코드 예제는 Athena와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

batch-get-named-query

다음 코드 예시는 batch-get-named-query의 사용 방법을 보여 줍니다.

AWS CLI

둘 이상의 쿼리에 대한 정보를 반환하는 방법

다음 batch-get-named-query 예제에서는 지정된 ID가 있는 명명된 쿼리들에 대한 정보를 반환합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 BatchGetNamedQuery를 참조하세요.

`batch-get-query-execution`

다음 코드 예시는 `batch-get-query-execution`의 사용 방법을 보여 줍니다.

AWS CLI

하나 이상의 쿼리 실행에 대한 정보를 반환하는 방법

다음 `batch-get-query-execution` 예제에서는 지정된 쿼리 ID가 있는 쿼리들에 대한 쿼리 실행 정보를 반환합니다.

```
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://amzn-s3-demo-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://amzn-s3-demo-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.

- API 세부 정보는 AWS CLI 명령 참조의 BatchGetQueryExecution을 참조하세요.

`create-data-catalog`

다음 코드 예시는 `create-data-catalog`의 사용 방법을 보여 줍니다.

AWS CLI

데이터 카탈로그를 생성하는 방법

다음 `dynamo_db_catalog` 예제에서는 `create-data-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 `aws athena get-data-catalog --name dynamo_db_catalog`를 사용합니다.

자세한 내용은 Amazon Athena 사용 설명서의 카탈로그 등록: create-data-catalog를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateDataCatalog를 참조하세요.

`create-named-query`

다음 코드 예시는 `create-named-query`의 사용 방법을 보여 줍니다.

AWS CLI

명명된 쿼리를 생성하는 방법

다음 `create-named-query` 예제에서는 출발 및 도착이 모두 10분 넘게 지연된 2016년 1월 시애틀 출발 뉴욕 도착 항공편에 대해 `flights_parquet` 테이블에 쿼리하는 `AthenaAdmin` 작업 그룹의 저장된 쿼리를 생성합니다. 테이블의 공항 코드 값은 큰따옴표가 포함된 문자열(예: “SEA”)이므로 이러한 값은 백슬래시로 이스케이프되고 작은따옴표로 묶입니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateNamedQuery를 참조하세요.

`create-work-group`

다음 코드 예시는 `create-work-group`의 사용 방법을 보여 줍니다.

AWS CLI

작업 그룹을 생성하는 방법

다음 `create-work-group` 예제에서는 쿼리 결과 출력 위치 `s3://amzn-s3-demo-bucket`이 있는 `Data_Analyst_Group`이라는 이름의 작업 그룹을 생성합니다. 이 명령은 클라이언트 구성 설정을 재정의하는 작업 그룹을 생성하며 여기에 쿼리 결과 출력 위치가 포함됩니다. 또한 이 명령은 CloudWatch 지표를 활성화하고 세 개의 키-값 태그 페어를 작업 그룹에 추가하여 다른 작업 그

룹과 구분합니다. 참고로, --configuration 인수에는 옵션을 구분하는 쉼표 앞에 공백이 없습니다.

```
aws athena create-work-group \
    --name Data_Analyst_Group \
    --configuration ResultConfiguration={OutputLocation="s3://amzn-s3-demo-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 aws athena get-work-group --work-group Data_Analyst_Group을 사용합니다.

자세한 내용은 Amazon Athena 사용 설명서의 작업 그룹 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateWorkGroup을 참조하세요.

delete-data-catalog

다음 코드 예시는 delete-data-catalog의 사용 방법을 보여 줍니다.

AWS CLI

데이터 카탈로그를 삭제하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteDataCatalog를 참조하세요.

delete-named-query

다음 코드 예시는 delete-named-query의 사용 방법을 보여 줍니다.

AWS CLI

명명된 쿼리를 삭제하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteNamedQuery를 참조하세요.

delete-work-group

다음 코드 예시는 delete-work-group의 사용 방법을 보여 줍니다.

AWS CLI

작업 그룹을 삭제하는 방법

다음 delete-work-group 예제에서는 TeamB 작업 그룹을 삭제합니다.

```
aws athena delete-work-group \
    --work-group TeamB
```

이 명령은 출력을 생성하지 않습니다. 삭제를 확인하려면 aws athena list-work-groups를 사용합니다.

자세한 내용은 Amazon Athena 사용 설명서의 작업 그룹 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteWorkGroup을 참조하세요.

get-data-catalog

다음 코드 예시는 get-data-catalog의 사용 방법을 보여 줍니다.

AWS CLI

데이터 카탈로그에 대한 정보를 반환하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 GetDataCatalog를 참조하세요.

get-database

다음 코드 예시는 get-database의 사용 방법을 보여 줍니다.

AWS CLI

데이터 카탈로그에서 데이터베이스에 대한 정보를 반환하는 방법

다음 get-database 예제에서는 AwsDataCatalog 데이터 카탈로그에서 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 표시: get-database]를 참조하세요.

- API에 대한 세부 정보는 AWS CLI 명령 참조의 [GetDatabase]를 참조하세요.

get-named-query

다음 코드 예시는 get-named-query의 사용 방법을 보여 줍니다.

AWS CLI

명명된 쿼리를 반환하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 GetNamedQuery를 참조하세요.

get-query-execution

다음 코드 예시는 `get-query-execution`의 사용 방법을 보여 줍니다.

AWS CLI

쿼리 실행에 대한 정보를 반환하는 방법

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://amzn-s3-demo-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.

- API 세부 정보는 AWS CLI 명령 참조의 GetQueryExecution을 참조하세요.

get-query-results

다음 코드 예시는 `get-query-results`의 사용 방법을 보여 줍니다.

AWS CLI

쿼리 결과를 반환하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 GetQueryResults를 참조하세요.

get-table-metadata

다음 코드 예시는 `get-table-metadata`의 사용 방법을 보여 줍니다.

AWS CLI

테이블에 대한 메타데이터 정보를 반환하는 방법

다음 `get-table-metadata` 예제에서는 열 이름 및 해당 데이터 유형을 포함하는 `counties` 테이블에 대한 메타데이터 정보를 `AwsDataCatalog` 데이터 카탈로그의 `sampledb` 데이터베이스에서 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://amzn-s3-demo-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.

- API 세부 정보는 AWS CLI 명령 참조의 GetTableMetadata를 참조하세요.

get-work-group

다음 코드 예시는 `get-work-group`의 사용 방법을 보여 줍니다.

AWS CLI

작업 그룹에 대한 정보를 반환하는 방법

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://amzn-s3-demo-bucket/"
            },
            "EnforceWorkGroupConfiguration": false,
            "PublishCloudWatchMetricsEnabled": true,
            "RequesterPaysEnabled": false
        },
        "Description": "Workgroup for Athena administrators",
        "CreationTime": 1573677174.105
    }
}
```

자세한 내용은 Amazon Athena 사용 설명서의 [작업 그룹 관리](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [GetWorkGroup](#)을 참조하세요.

`list-data-catalogs`

다음 코드 예시는 `list-data-catalogs`의 사용 방법을 보여 줍니다.

AWS CLI

Athena에 등록된 데이터 카탈로그를 나열하는 방법

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-catalogs를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListDataCatalogs를 참조하세요.

list-databases

다음 코드 예시는 `list-databases`의 사용 방법을 보여 줍니다.

AWS CLI

데이터 카탈로그의 데이터베이스를 나열하는 방법

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 카탈로그의 데이터베이스 나열: list-databases를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListDatabases를 참조하세요.

list-named-queries

다음 코드 예시는 `list-named-queries`의 사용 방법을 보여 줍니다.

AWS CLI

작업 그룹에 대한 명명된 쿼리를 나열하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 ListNamedQueries를 참조하세요.

`list-query-executions`

다음 코드 예시는 `list-query-executions`의 사용 방법을 보여 줍니다.

AWS CLI

지정된 작업 그룹에서 쿼리의 쿼리 ID를 나열하는 방법

다음 `list-query-executions` 예제에서는 `AthenaAdmin` 작업 그룹의 쿼리 ID를 최대 열 개까지 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 작업]을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ListQueryExecutions]를 참조하세요.

`list-table-metadata`

다음 코드 예시는 `list-table-metadata`의 사용 방법을 보여 줍니다.

AWS CLI

데이터 카탈로그의 지정된 데이터베이스에서 테이블에 대한 메타데이터를 나열하는 방법

다음 `list-table-metadata` 예제에서는 `AwsDataCatalog` 데이터 카탈로그의 `geography` 데이터베이스에서 최대 두 개의 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://amzn-s3-demo-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://amzn-s3-demo-bucket/csv/CountyPopulation",
                "outputformat":
 "org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat",
                "serde.param.field.delim": ",",
                "serde.serialization.lib":
 "org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe",
                "skip.header.line.count": "1",
                "typeOfData": "file"
            }
        }
    ],
```

```
    "NextToken": "eyJ0ZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAyfQ=="
}
```

자세한 내용은 Amazon Athena 사용 설명서의 데이터베이스의 모든 테이블에 대한 메타데이터 표시: list-table-metadata를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTableMetadata를 참조하세요.

list-tags-for-resource

다음 코드 예시는 list-tags-for-resource의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 작업 그룹에 대한 태그를 나열하는 방법

다음 list-tags-for-resource 예제에서는 Data_Analyst_Group 작업 그룹에 대한 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 카탈로그에 대한 태그를 나열하는 방법

다음 `list-tags-for-resource` 예제에서는 `dynamo_db_catalog` 데이터 카탈로그에 대한 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 리소스에 대한 태그 나열: list-tags-for-resource를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource를 참조하세요.

`list-work-groups`

다음 코드 예시는 `list-work-groups`의 사용 방법을 보여 줍니다.

AWS CLI

작업 그룹을 나열하는 방법

다음 `list-work-groups` 예제에서는 현재 계정에서 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListWorkGroups를 참조하세요.

`start-query-execution`

다음 코드 예시는 `start-query-execution`의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 지정된 데이터베이스 및 데이터 카탈로그 내 지정된 테이블에 있는 작업 그룹에서 쿼리를 실행하는 방법

다음 `start-query-execution` 예제에서는 `AthenaAdmin` 작업 그룹을 사용하여 `AwsDataCatalog` 데이터 카탈로그 내 `cflogsdatabase`의 `cloudfront_logs` 테이블에서 쿼리를 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 실행]을 참조하세요.

예제 2: 지정된 작업 그룹을 사용하여 지정된 데이터 카탈로그에서 데이터베이스를 생성하는 쿼리를 실행하는 방법

다음 `start-query-execution` 예제에서는 `AthenaAdmin` 작업 그룹을 사용하여 기본 데이터 카탈로그 `AwsDataCatalog`에서 newdb 데이터베이스를 생성합니다.

```
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.

예제 3: 지정된 데이터베이스 및 데이터 카탈로그 내 테이블에서 뷰를 생성하는 쿼리를 실행하는 방법

다음 `start-query-execution` 예제에서는 `SELECT` 문을 사용하여 `cflogsdatabase` 내 `cloudfront_logs` 테이블에서 `cf10` 뷰를 생성합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 StartQueryExecution을 참조하세요.

`stop-query-execution`

다음 코드 예시는 `stop-query-execution`의 사용 방법을 보여 줍니다.

AWS CLI

실행 중인 쿼리를 중지하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 [StopQueryExecution]을 참조하세요.

tag-resource

다음 코드 예시는 tag-resource의 사용 방법을 보여 줍니다.

AWS CLI

리소스에 태그를 추가하는 방법

다음 tag-resource 예제에서는 dynamo_db_catalog 데이터 카탈로그에 태그 세 개를 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 aws athena list-tags-for-resource --resource-arn arn:aws:athena:us-west-2:111122223333:datacatalog/dynamo_db_catalog를 사용합니다.

자세한 내용은 Amazon Athena 사용 설명서의 [리소스에 태그 추가: tag-resource]를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [TagResource]를 참조하세요.

untag-resource

다음 코드 예시에서는 untag-resource의 사용 방법을 보여줍니다.

AWS CLI

리소스에서 태그를 제거하는 방법

다음 untag-resource 예제에서는 Specialization 및 Focus 키와 해당 관련 값들을 dynamo_db_catalog 데이터 카탈로그 리소스에서 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.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

`update-data-catalog`

다음 코드 예시는 `update-data-catalog`의 사용 방법을 보여 줍니다.

AWS CLI

데이터 카탈로그를 업데이트하는 방법

다음 `update-data-catalog` 예제에서는 `cw_logs_catalog` 데이터 카탈로그의 Lambda 함수와 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 `aws athena get-data-catalog --name cw_logs_catalog`를 사용합니다.

자세한 내용은 Amazon Athena 사용 설명서의 카탈로그 업데이트: update-data-catalog를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateDataCatalog를 참조하세요.

`update-work-group`

다음 코드 예시는 `update-work-group`의 사용 방법을 보여 줍니다.

AWS CLI

작업 그룹을 업데이트하는 방법

다음 update-work-group 예제에서는 Data_Analyst_Group 작업 그룹을 비활성화합니다. 사용자는 비활성화된 작업 그룹에서 쿼리를 실행하거나 생성할 수 없지만 여전히 지표, 데이터 사용량 제한 제어, 작업 그룹 설정, 쿼리 기록 및 저장된 쿼리를 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_Group을 사용하고 출력에서 State 속성을 확인합니다.

자세한 내용은 Amazon Athena 사용 설명서의 작업 그룹 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateWorkGroup을 참조하세요.

AWS CLI를 사용하는 Auto Scaling 예제

다음 코드 예제는 Auto Scaling과 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

attach-instances

다음 코드 예시는 attach-instances의 사용 방법을 보여 줍니다.

AWS CLI

Auto Scaling 그룹에 인스턴스를 연결하는 방법

이 예시에서는 지정된 Auto Scaling 그룹에 지정된 인스턴스를 연결합니다.

```
aws autoscaling attach-instances \
    --instance-ids i-061c63c5eb45f0416 \
    --auto-scaling-group-name my-asg
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 AttachInstances를 참조하세요.

attach-load-balancer-target-groups

다음 코드 예시는 `attach-load-balancer-target-groups`의 사용 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 EC2 Auto Scaling을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AttachLoadBalancerTargetGroups를 참조하세요.

attach-load-balancers

다음 코드 예시는 `attach-load-balancers`의 사용 방법을 보여 줍니다.

AWS CLI

Auto Scaling 그룹에 Classic Load Balancer를 연결하는 방법

이 예제에서는 지정된 Auto Scaling 그룹에 지정된 Classic Load Balancer를 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 EC2 Auto Scaling을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AttachLoadBalancers를 참조하세요.

cancel-instance-refresh

다음 코드 예시는 `cancel-instance-refresh`의 사용 방법을 보여 줍니다.

AWS CLI

인스턴스 새로 고침을 취소하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 CancelInstanceRefresh를 참조하세요.

complete-lifecycle-action

다음 코드 예시는 `complete-lifecycle-action`의 사용 방법을 보여 줍니다.

AWS CLI

수명 주기 작업을 완료하는 방법

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 설명은 Amazon EC2 Auto Scaling 사용자 가이드의 Amazon EC2 Auto Scaling 수명 주기 후크를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CompleteLifecycleAction을 참조하세요.

`create-auto-scaling-group`

다음 코드 예시는 `create-auto-scaling-group`의 사용 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 EC2 Auto Scaling 사용 설명서의 오토 스케일링을 참조하세요.

예 2: Application Load Balancer, Network Load Balancer 또는 Gateway Load Balancer를 연결하는 방법

이 예시에서는 예상 트래픽을 지원하는 로드 밸런서의 대상 그룹 ARN을 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 EC2 Auto Scaling을 참조하세요.

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 내용을 알아보려면 Amazon EC2 - Linux 인스턴스용 사용 설명서의 배치 그룹을 참조하세요.

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. 또한 이 명령은 기존 ENI의 ID가 지정된 시작 템플릿의 v1을 지정합니다. eth0의 기존 ENI를 지정하는 시작 템플릿을 사용하는 경우 요청에서 서브넷 ID는 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 EC2 Auto Scaling 사용 설명서의 오토 스케일링을 참조하세요.

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 내용을 알아보려면 Amazon EC2 Auto Scaling 사용 설명서의 Amazon EC2 Auto Scaling 종료 정책 사용을 참조하세요.

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 설명은 Amazon EC2 Auto Scaling 사용자 가이드의 Amazon EC2 Auto Scaling 수명 주기 후크를 참조하세요.

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 설명은 Amazon EC2 Auto Scaling 사용자 가이드의 Amazon EC2 Auto Scaling 수명 주기 후크를 참조하세요.

예 8: 사용자 지정 종료 정책을 지정하는 방법

이 예시에서는 스케일 인할 때 안전하게 종료할 수 있는 인스턴스를 Amazon EC2 Auto Scaling에 알려주는 사용자 지정 Lambda 함수 종료 정책을 지정하는 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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateAutoScalingGroup을 참조하세요.

create-launch-configuration

다음 코드 예시는 `create-launch-configuration`의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 시작 구성을 생성하는 방법

이 예시에서는 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, 부트스트래핑 스크립트를 사용하여 시작 구성을 생성하는 방법

이 예시에서는 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 생성하는 방법

이 예제에서는 IAM 역할의 인스턴스 프로필 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 내용을 알아보려면 Amazon EC2 Auto Scaling 사용 설명서의 Amazon EC2 인스턴스에서 실행되는 애플리케이션의 IAM 역할을 참조하세요.

예제 4: 세부 모니터링이 활성화된 시작 구성을 생성하는 방법

이 예시에서는 EC2 세부 모니터링이 활성화된 시작 구성을 생성하여 EC2 지표를 1분 내에 CloudWatch로 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 EC2 Auto Scaling 사용 설명서의 Auto Scaling 인스턴스에 대한 모니터링 구성을 참조하세요.

예제 5: 스팟 인스턴스를 시작하는 시작 구성을 생성하는 방법

이 예시에서는 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 생성하는 방법

이 예시에서는 기존 인스턴스의 속성을 기반으로 시작 구성을 생성합니다. 배치 테넌시와 `--placement-tenancy` 및 `--no-associate-public-ip-address` 옵션을 포함하여 퍼블릭 IP 주소의 설정 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 생성하는 방법

이 예시에서는 디바이스 이름과 gp3 볼륨 크기가 20인 Amazon EBS `/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 EC2 Auto Scaling API 참조의 EBS 섹션을 참조하세요.

JSON 형식의 파라미터 값을 인용하는 구문에 대한 자세한 내용은 AWS 명령줄 인터페이스 사용 설명서의 AWS CLI에서 문자열에 따옴표 사용을 참조하세요.

예제 8: 인스턴스 저장소 볼륨에 대한 블록 디바이스 매핑을 사용하여 시작 구성을 생성하는 방법

이 예시에서는 를 디바이스 이름 의 인스턴스 스토어 볼륨`ephemeral1`으로 사용하여 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 EC2 Auto Scaling API 참조의 BlockDeviceMapping 섹션을 참조하세요.

JSON 형식의 파라미터 값을 인용하는 구문에 대한 자세한 내용은 AWS 명령줄 인터페이스 사용 설명서의 AWS CLI에서 문자열에 따옴표 사용을 참조하세요.

예제 9: 시작 구성을 생성하고 시작 시 블록 디바이스의 연결을 억제하는 방법

이 예시에서는 AMI의 블록 디바이스 매핑(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 EC2 Auto Scaling API 참조의 BlockDeviceMapping 섹션을 참조하세요.

JSON 형식의 파라미터 값을 인용하는 구문에 대한 자세한 내용은 AWS 명령줄 인터페이스 사용 설명서의 AWS CLI에서 문자열에 따옴표 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateLaunchConfiguration을 참조하세요.

`create-or-update-tags`

다음 코드 예시는 `create-or-update-tags`의 사용 방법을 보여 줍니다.

AWS CLI

Auto Scaling 그룹의 태그를 생성하거나 업데이트하려면

이 예시에서는 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 EC2 Auto Scaling 사용 설명서의 Auto Scaling 그룹 및 인스턴스 태그 지정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의CreateOrUpdateTags를 참조하세요.

`delete-auto-scaling-group`

다음 코드 예시는 `delete-auto-scaling-group`의 사용 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 EC2 Auto Scaling 사용 설명서의 Auto Scaling 인프라 삭제를 참조하세요.

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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteAutoScalingGroup을 참조하세요.

delete-launch-configuration

다음 코드 예시는 delete-launch-configuration의 사용 방법을 보여 줍니다.

AWS CLI

시작 구성을 삭제하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteLaunchConfiguration을 참조하세요.

delete-lifecycle-hook

다음 코드 예시는 delete-lifecycle-hook의 사용 방법을 보여 줍니다.

AWS CLI

수명 주기 후크를 삭제하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteLifecycleHook을 참조하세요.

delete-notification-configuration

다음 코드 예시는 delete-notification-configuration의 사용 방법을 보여 줍니다.

AWS CLI

Auto Scaling 알림을 삭제하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteNotificationConfiguration을 참조하세요.

delete-policy

다음 코드 예시는 `delete-policy`의 사용 방법을 보여 줍니다.

AWS CLI

조정 정책을 삭제하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 DeletePolicy를 참조하세요.

delete-scheduled-action

다음 코드 예시는 `delete-scheduled-action`의 사용 방법을 보여 줍니다.

AWS CLI

Auto Scaling 그룹에서 예약된 작업을 삭제하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteScheduledAction 섹션을 참조하세요.

delete-tags

다음 코드 예시는 `delete-tags`의 사용 방법을 보여 줍니다.

AWS CLI

Auto Scaling 그룹에서 태그를 삭제하는 방법

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 EC2 Auto Scaling 사용 설명서의 Auto Scaling 그룹 및 인스턴스 태그 지정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteTags 섹션을 참조하세요.

delete-warm-pool

다음 코드 예시는 `delete-warm-pool`의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 웜 풀을 삭제하는 방법

다음 예제에서는 지정된 Auto Scaling 그룹에 대한 웜 풀을 삭제합니다.

```
aws autoscaling delete-warm-pool \
    --auto-scaling-group-name my-asg
```

이 명령은 출력을 생성하지 않습니다.

자세한 설명은 Amazon EC2 Auto Scaling 사용자 가이드의 Amazon EC2 Auto Scaling을 위한 웜 풀을 참조하세요.

예제 2: 웜 풀을 강제로 삭제하는 방법

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 설명은 Amazon EC2 Auto Scaling 사용자 가이드의 Amazon EC2 Auto Scaling을 위한 웜 풀을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteWarmPool을 참조하세요.

`describe-account-limits`

다음 코드 예시는 `describe-account-limits`의 사용 방법을 보여 줍니다.

AWS CLI

Amazon EC2 Auto Scaling 계정 제한을 설명하려면

이 예시에서는 AWS 계정의 Amazon EC2 Auto Scaling 제한을 설명합니다.

```
aws autoscaling describe-account-limits
```

출력:

```
{
    "NumberOfLaunchConfigurations": 5,
    "MaxNumberOfLaunchConfigurations": 100,
    "NumberOfAutoScalingGroups": 3,
    "MaxNumberOfAutoScalingGroups": 20
}
```

자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 Amazon EC2 Auto Scaling 서비스 할당량을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeAccountLimits 섹션을 참조하세요.

describe-adjustment-types

다음 코드 예시는 `describe-adjustment-types`의 사용 방법을 보여 줍니다.

AWS CLI

사용 가능한 조정 유형을 설명하려면

이 예시에서는 사용 가능한 조정 유형을 설명합니다.

```
aws autoscaling describe-adjustment-types
```

출력:

```
{
    "AdjustmentTypes": [
        {
            "AdjustmentType": "ChangeInCapacity"
        },
        {
            "AdjustmentType": "ExactCapacity"
        },
        {
            "AdjustmentType": "PercentChangeInCapacity"
        }
    ]
}
```

자세한 내용을 알아보려면 Amazon EC2 Auto Scaling 사용 설명서의 조정 조절 유형을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeAdjustmentTypes를 참조하세요.

describe-auto-scaling-groups

다음 코드 예시는 `describe-auto-scaling-groups`의 사용 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-names 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-names "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 그룹을 설명하는 방법

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 사용자 인터페이스 사용 설명서의 AWS CLI 출력 필터링을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeAutoScalingGroups를 참조하세요.

describe-auto-scaling-instances

다음 코드 예시는 describe-auto-scaling-instances의 사용 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 인스턴스를 설명하는 방법

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 Command Line Interface 사용 설명서의 Filter AWS CLI output 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeAutoScalingInstances를 참조하세요.

describe-auto-scaling-notification-types

다음 코드 예시는 describe-auto-scaling-notification-types의 사용 방법을 보여 줍니다.

AWS CLI

사용 가능한 알림 유형을 설명하는 방법

이 예시에서는 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 정보는 Amazon EC2 Auto Scaling 사용 설명서에서 Auto Scaling 그룹 조정 시 Amazon SNS 알림 수신을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeAutoScalingNotificationTypes를 참조하세요.

describe-instance-refreshes

다음 코드 예시는 describe-instance-refreshes의 사용 방법을 보여 줍니다.

AWS CLI

인스턴스 새로 고침을 설명하는 방법

다음 describe-instance-refreshes 예제에서는 지정된 Auto Scaling 그룹에 대해 상태 메시지 및 상태 이유(사용 가능한 경우)를 포함하는 모든 인스턴스 새로 고침 요청의 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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeInstanceRefreshes를 참조하세요.

describe-launch-configurations

다음 코드 예시는 describe-launch-configurations의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 지정된 시작 구성을 설명하는 방법

이 예시에서는 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 설명하는 방법

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 직접 호출에서 이 필드의 값을 `--starting-token` 옵션과 함께 사용하세요.

```
aws autoscaling describe-launch-configurations \
    --starting-token Z3M3LMPEXAMPLE
```

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeLaunchConfigurations]를 참조하세요.

`describe-lifecycle-hook-types`

다음 코드 예시는 `describe-lifecycle-hook-types`의 사용 방법을 보여 줍니다.

AWS CLI

사용 가능한 수명 주기 후크 유형을 설명하는 방법

이 예시에서는 사용 가능한 수명 주기 후크의 유형을 설명합니다.

```
aws autoscaling describe-lifecycle-hook-types
```

출력:

```
{
    "LifecycleHookTypes": [
        "autoscaling:EC2_INSTANCE_LAUNCHING",
        "autoscaling:EC2_INSTANCE_TERMINATING"
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeLifecycleHookTypes]를 참조하세요.

describe-lifecycle-hooks

다음 코드 예시는 describe-lifecycle-hooks의 사용 방법을 보여 줍니다.

AWS CLI

수명 주기 후크를 설명하는 방법

이 예시에서는 지정된 Auto Scaling 그룹에 대한 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeLifecycleHooks를 참조하세요.

describe-load-balancer-target-groups

다음 코드 예시는 describe-load-balancer-target-groups의 사용 방법을 보여 줍니다.

AWS CLI

Auto Scaling 그룹의 로드 밸런서 대상 그룹을 설명하는 방법

이 예제에서는 지정된 Auto Scaling 그룹에 연결된 로드 밸런서 대상 그룹을 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeLoadBalancerTargetGroups를 참조하세요.

`describe-load-balancers`

다음 코드 예시는 `describe-load-balancers`의 사용 방법을 보여 줍니다.

AWS CLI

Auto Scaling 그룹의 Classic Load Balancer를 설명하는 방법

이 예시에서는 지정된 Auto Scaling 그룹에 대한 Classic Load Balancer를 설명합니다.

```
aws autoscaling describe-load-balancers \
    --auto-scaling-group-name my-asg
```

출력:

```
{
    "LoadBalancers": [
        {
            "State": "Added",
```

```
            "LoadBalancerName": "my-load-balancer"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeLoadBalancers를 참조하세요.

describe-metric-collection-types

다음 코드 예시는 describe-metric-collection-types의 사용 방법을 보여 줍니다.

AWS CLI

사용 가능한 지표 컬렉션 유형을 설명하는 방법

이 예시에서는 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 EC2 Auto Scaling 사용 설명서의 Auto Scaling 그룹 지표를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeMetricCollectionTypes를 참조하세요.

describe-notification-configurations

다음 코드 예시는 `describe-notification-configurations`의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 지정된 그룹의 알림 구성을 설명하는 방법

이 예시에서는 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 정보는 Amazon EC2 Auto Scaling 사용 설명서에서 Auto Scaling 그룹 조정 시 Amazon SNS 알림 수신을 참조하세요.

예제 1: 지정된 수의 알림 구성을 설명하는 방법

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 직접 호출에서 이 필드의 값을 `starting-token` 파라미터와 함께 사용하세요.

```
aws autoscaling describe-notification-configurations \
    --auto-scaling-group-name my-asg \
    --starting-token Z3M3LMPEXAMPLE
```

자세한 정보는 Amazon EC2 Auto Scaling 사용 설명서에서 [Auto Scaling 그룹 조정 시 Amazon SNS 알림 수신](#)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeNotificationConfigurations](#)를 참조하세요.

`describe-policies`

다음 코드 예시는 `describe-policies`의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 지정된 그룹의 스케일링 정책 설명

이 예시에서는 지정된 Auto Scaling 그룹의 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 EC2 Auto Scaling 사용 설명서의 동적 조정을 참조하세요.

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 EC2 Auto Scaling 사용 설명서의 동적 조정을 참조하세요.

예제 3: 여러 조정 정책을 설명하는 방법

특정 개수의 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 EC2 Auto Scaling 사용 설명서의 동적 조정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribePolicies를 참조하세요.

describe-scaling-activities

다음 코드 예시는 describe-scaling-activities의 사용 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 내용을 알아보려면 Amazon EC2 Auto Scaling 사용 설명서의 [Auto Scaling 그룹에 대한 크기 조정 활동 확인](#)을 참조하세요.

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 설명은 Amazon EC2 Auto Scaling 사용 설명서의 Amazon EC2 Auto Scaling 문제 해결을 참조하세요.

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 내용을 알아보려면 Amazon EC2 Auto Scaling 사용 설명서의 Auto Scaling 그룹에 대한 크기 조정 활동 확인을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeScalingActivities를 참조하세요.

describe-scaling-process-types

다음 코드 예시는 describe-scaling-process-types의 사용 방법을 보여 줍니다.

AWS CLI

사용 가능한 프로세스 유형을 설명하는 방법

이 예시에서는 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 중단 및 재개 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeScalingProcessTypes를 참조하세요.

describe-scheduled-actions

다음 코드 예시는 describe-scheduled-actions의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 예약된 모든 작업을 설명하는 방법

이 예시에서는 예약된 모든 작업을 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 설명은 Amazon EC2 Auto Scaling 사용자 가이드의 예약 조정을 참조하세요.

예제 2: 지정된 그룹에 예정된 작업을 설명하는 방법

특정 Auto Scaling 그룹에 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 설명은 Amazon EC2 Auto Scaling 사용자 가이드의 예약 조정을 참조하세요.

예제 3: 지정된 예약 작업을 설명하는 방법

특정 예약 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 설명은 Amazon EC2 Auto Scaling 사용자 가이드의 예약 조정을 참조하세요.

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 설명은 [Amazon EC2 Auto Scaling 사용자 가이드]의 예약 조정을 참조하세요.

예제 5: 지정된 시간에 종료되는 예약 작업을 설명하는 방법

특정 시간에 종료되는 예약 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 설명은 Amazon EC2 Auto Scaling 사용자 가이드의 예약 조정을 참조하세요.

예제 6: 지정된 수의 예약 작업을 설명하는 방법

특정 개수의 예약 작업을 반환하려면 `--max-items` 옵션을 사용하세요.

```
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 예정 작업이 더 많습니다. 추가 예정 작업을 가져오려면 다음과 같이 후속 직접 호출에서 이 필드의 값을 `--starting-token` 옵션과 함께 사용하세요.

```
aws autoscaling describe-scheduled-actions \
    --auto-scaling-group-name my-asg \
    --starting-token Z3M3LMPEXAMPLE
```

자세한 설명은 Amazon EC2 Auto Scaling 사용자 가이드의 예약 조정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeScheduledActions 섹션을 참조하세요.

`describe-tags`

다음 코드 예시는 `describe-tags`의 사용 방법을 보여 줍니다.

AWS CLI

모든 태그를 설명하는 방법

이 예시에서는 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 EC2 Auto Scaling 사용 설명서의 Auto Scaling 그룹 및 인스턴스 태그 지정을 참조하세요.

예제 2: 지정된 그룹에 대한 태그를 설명하는 방법

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 EC2 Auto Scaling 사용 설명서의 Auto Scaling 그룹 및 인스턴스 태그 지정을 참조하세요.

예제 3: 지정된 개수의 태그를 설명하는 방법

특정 개수의 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 직접 호출에서 이 필드의 값을 `--starting-token` 옵션과 함께 사용하세요.

```
aws autoscaling describe-tags \
    --filters Name=auto-scaling-group,Values=my-asg \
    --starting-token Z3M3LMPEXAMPLE
```

자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 Auto Scaling 그룹 및 인스턴스 태그 지정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeTags 섹션을 참조하세요.

`describe-termination-policy-types`

다음 코드 예시는 `describe-termination-policy-types`의 사용 방법을 보여 줍니다.

AWS CLI

사용 가능한 종료 정책 유형을 설명하는 방법

이 예시에서는 사용 가능한 종료 정책 유형을 설명합니다.

```
aws autoscaling describe-termination-policy-types
```

출력:

```
{
    "TerminationPolicyTypes": [
        "AllocationStrategy",
        "ClosestToNextInstanceHour",
        "Default",
```

```
            "NewestInstance",
            "OldestInstance",
            "OldestLaunchConfiguration",
            "OldestLaunchTemplate"
        ]
}
```

자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 스케일 인 중 Auto Scaling 인스턴스 종료 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeTerminationPolicyTypes를 참조하세요.

describe-warm-pool

다음 코드 예시는 describe-warm-pool의 사용 방법을 보여 줍니다.

AWS CLI

웜 풀을 설명하는 방법

이 예제에서는 지정된 Auto Scaling 그룹에 대한 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 설명은 Amazon EC2 Auto Scaling 사용자 가이드의 Amazon EC2 Auto Scaling을 위한 웜 풀을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeWarmPool을 참조하세요.

detach-instances

다음 코드 예시는 detach-instances의 사용 방법을 보여 줍니다.

AWS CLI

Auto Scaling 그룹에서 인스턴스를 분리하려면

이 예시에서는 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\"}"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DetachInstances를 참조하세요.

detach-load-balancer-target-groups

다음 코드 예시는 detach-load-balancer-target-groups의 사용 방법을 보여 줍니다.

AWS CLI

Auto Scaling 그룹에서 로드 밸런서 대상 그룹을 분리하는 방법

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 EC2 Auto Scaling 사용 설명서에서 로드 밸런서를 오토 스케일링 그룹에 연결을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DetachLoadBalancerTargetGroups를 참조하세요.

detach-load-balancers

다음 코드 예시는 detach-load-balancers의 사용 방법을 보여 줍니다.

AWS CLI

Auto Scaling 그룹에서 Classic Load Balancer를 분리하려면

이 예시에서는 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 EC2 Auto Scaling 사용 설명서에서 로드 밸런서를 오토 스케일링 그룹에 연결을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DetachLoadBalancers를 참조하세요.

`disable-metrics-collection`

다음 코드 예시는 `disable-metrics-collection`의 사용 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 EC2 Auto Scaling 사용 설명서의 Auto Scaling 그룹 및 인스턴스에 대한 CloudWatch 지표 모니터링을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisableMetricsCollection을 참조하세요.

`enable-metrics-collection`

다음 코드 예시는 `enable-metrics-collection`의 사용 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 EC2 Auto Scaling 사용 설명서의 Auto Scaling 그룹 및 인스턴스에 대한 CloudWatch 지표 모니터링을 참조하세요.

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 EC2 Auto Scaling 사용 설명서의 Auto Scaling 그룹 및 인스턴스에 대한 CloudWatch 지표 모니터링을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 EnableMetricsCollection을 참조하세요.

`enter-standby`

다음 코드 예시는 `enter-standby`의 사용 방법을 보여 줍니다.

AWS CLI

인스턴스를 대기 모드로 이동하려면

이 예시에서는 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 EC2 Auto Scaling instance lifecycle 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 EnterStandby를 참조하세요.

execute-policy

다음 코드 예시는 `execute-policy`의 사용 방법을 보여 줍니다.

AWS CLI

조정 정책을 실행하려면

이 예제에서는 지정된 Auto Scaling 그룹에 `my-step-scale-out-policy`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 조정 및 단순 조정 정책을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ExecutePolicy를 참조하세요.

exit-standby

다음 코드 예시는 exit-standby의 사용 방법을 보여 줍니다.

AWS CLI

인스턴스를 대기 모드에서 해제하려면

이 예제에서는 지정된 인스턴스를 대기 모드에서 해제합니다.

```
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 https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-enter-exit-standby.htmlAmazon EC2 Auto Scaling 사용 설명서의 Auto Scaling 그룹에서 일시적으로 인스턴스 제거를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ExitStandby를 참조하세요.

put-lifecycle-hook

다음 코드 예시는 put-lifecycle-hook의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 수명 주기 후크를 생성하는 방법

이 예제에서는 4,800초의 제한 시간으로 새로 시작된 인스턴스에서 호출할 수명 주기 후크를 생성합니다. 이는 사용자 데이터 스크립트가 완료될 때까지 인스턴스를 대기 상태로 유지하거나 EventBridge를 사용하여 AWS Lambda 함수를 호출하는 데 유용합니다.

```
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 설명은 Amazon EC2 Auto Scaling 사용자 가이드의 Amazon EC2 Auto Scaling 수명 주기 후크를 참조하세요.

예제 2: 인스턴스의 상태 전환을 알리기 위해 Amazon SNS 이메일 메시지를 전송하는 방법

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 설명은 Amazon EC2 Auto Scaling 사용자 가이드의 Amazon EC2 Auto Scaling 수명 주기 후크를 참조하세요.

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 설명은 Amazon EC2 Auto Scaling 사용자 가이드의 Amazon EC2 Auto Scaling 수명 주기 후크를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutLifecycleHook을 참조하세요.

put-notification-configuration

다음 코드 예시는 `put-notification-configuration`의 사용 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 정보는 Amazon EC2 Auto Scaling 사용 설명서에서 Auto Scaling 그룹 조정 시 Amazon SNS 알림 수신을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutNotificationConfiguration을 참조하세요.

put-scaling-policy

다음 코드 예시는 put-scaling-policy의 사용 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 ARN과 사용자를 대신하여 생성된 두 개의 CloudWatch 경보 이름이 포함됩니다. 같은 이름의 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 예시를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutScalingPolicy를 참조하세요.

put-scheduled-update-group-action

다음 코드 예시는 put-scheduled-update-group-action의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: Auto Scaling 그룹에 예약된 작업을 추가하는 방법

이 예제에서는 지정된 Auto Scaling 그룹에 예약 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 예약 작업으로 덮어씁니다.

자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 예약 조정을 참조하세요.

예제 2: 반복 일정을 지정하는 방법

이 예제에서는 매년 1월, 6월, 12월 1일 00:30시에 실행되도록 예약된 작업을 생성하여 반복 일정에 따라 조정합니다.

```
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 예약 작업으로 덮어씁니다.

자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 예약 조정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutScheduledUpdateGroupAction을 참조하세요.

put-warm-pool

다음 코드 예시는 put-warm-pool의 사용 방법을 보여 줍니다.

AWS CLI

웜 풀을 생성하는 방법

다음 예제에서는 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 해당 웜 풀이 업데이트됩니다.

자세한 설명은 Amazon EC2 Auto Scaling 사용자 가이드의 Amazon EC2 Auto Scaling을 위한 웜 풀을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutWarmPool을 참조하세요.

record-lifecycle-action-heartbeat

다음 코드 예시는 record-lifecycle-action-heartbeat의 사용 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 설명은 Amazon EC2 Auto Scaling 사용자 가이드의 Amazon EC2 Auto Scaling 수명 주기 후크를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RecordLifecycleActionHeartbeat를 참조하세요.

resume-processes

다음 코드 예시는 resume-processes의 사용 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 중단 및 재개 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ResumeProcesses를 참조하세요.

rollback-instance-refresh

다음 코드 예시는 rollback-instance-refresh의 사용 방법을 보여 줍니다.

AWS CLI

인스턴스 새로 고침을 롤백하는 방법

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 설명은 Amazon EC2 Auto Scaling 사용자 가이드의 롤백으로 변경 취소하기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RollbackInstanceRefresh를 참조하세요.

set-desired-capacity

다음 코드 예시는 set-desired-capacity의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 SetDesiredCapacity를 참조하세요.

set-instance-health

다음 코드 예시는 `set-instance-health`의 사용 방법을 보여 줍니다.

AWS CLI

인스턴스 상태를 설정하려면

이 예제에서는 지정된 인스턴스의 상태를 `Unhealthy`로 설정합니다.

```
aws autoscaling set-instance-health \
    --instance-id i-061c63c5eb45f0416 \
    --health-status Unhealthy
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 SetInstanceHealth를 참조하세요.

set-instance-protection

다음 코드 예시는 `set-instance-protection`의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 인스턴스의 인스턴스 보호 설정을 활성화하는 방법

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: 인스턴스의 인스턴스 보호 설정을 비활성화하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 SetInstanceProtection을 참조하세요.

`start-instance-refresh`

다음 코드 예시는 `start-instance-refresh`의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 명령줄 파라미터를 사용하여 인스턴스 새로 고침을 시작하는 방법

다음 `start-instance-refresh` 예제에서는 명령줄 인수를 사용하여 인스턴스 새로 고침을 시작합니다. 선택적 `preferences` 파라미터는 `60`초의 `InstanceWarmup`과 `50`%의 `MinHealthyPercentage`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 시작하는 방법

다음 `start-instance-refresh` 예제에서는 JSON 파일을 사용하여 인스턴스 새로 고침을 시작합니다. 다음 예제와 같이 Auto Scaling 그룹을 지정하고 JSON 파일에서 원하는 구성 및 기본 설정을 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.

• API 세부 정보는 AWS CLI 명령 참조의 StartInstanceRefresh를 참조하세요.

suspend-processes

다음 코드 예시는 suspend-processes의 사용 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 중단 및 재개 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 SuspendProcesses를 참조하세요.

terminate-instance-in-auto-scaling-group

다음 코드 예시는 `terminate-instance-in-auto-scaling-group`의 사용 방법을 보여 줍니다.

AWS CLI

Auto Scaling 그룹에서 인스턴스를 종료하는 방법

이 예시에서는 그룹 크기를 업데이트하지 않고 지정된 Auto Scaling 그룹에서 지정된 인스턴스를 종료합니다. 지정된 인스턴스가 종료된 후에 Amazon EC2 Auto Scaling은 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\"}"
        }
    ]
```

```
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [TerminateInstanceInAutoScalingGroup]을 참조하세요.

`update-auto-scaling-group`

다음 코드 예시는 `update-auto-scaling-group`의 사용 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 설명은 Amazon EC2 Auto Scaling 사용 설명서의 [Auto Scaling 그룹에 대한 스케일링 제한 설정]을 참조하세요.

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 여러 가용 영역의 서브넷 ID 목록을 사용하여 `--vpc-zone-identifier`의 값을 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 EC2 Auto Scaling]을 참조하세요.

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 EC2 Auto Scaling 사용 설명서의 오토 스케일링을 참조하세요.

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 설명서에서 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 설명서에서 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 설명은 Amazon EC2 Auto Scaling 사용자 가이드의 여러 인스턴스 유형 및 구매 옵션이 포함된 Auto Scaling 그룹을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateAutoScalingGroup을 참조하세요.

AWS CLI를 사용하는 Auto Scaling Plans 예제

다음 코드 예제는 Auto Scaling Plans와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

작업

`create-scaling-plan`

다음 코드 예시는 `create-scaling-plan`의 사용 방법을 보여 줍니다.

AWS CLI

조정 계획을 생성하는 방법

다음 `create-scaling-plan` 예제에서는 이미 생성된 JSON 파일(이름이 config.json인 파일)을 사용하여 `my-scaling-plan`이라는 이름의 조정 계획을 생성합니다. 조정 계획의 구조에는 `my-asg`라는 이름의 Auto Scaling 그룹에 대한 스케일링 지침이 포함되어 있습니다. 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 설명서를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateScalingPlan을 참조하세요.

delete-scaling-plan

다음 코드 예시는 delete-scaling-plan의 사용 방법을 보여 줍니다.

AWS CLI

조정 계획을 삭제하는 방법

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 설명서를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteScalingPlan을 참조하세요.

describe-scaling-plan-resources

다음 코드 예시는 describe-scaling-plan-resources의 사용 방법을 보여 줍니다.

AWS CLI

조정 계획을 위한 확장 가능 리소스를 설명하는 방법

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 AWS Auto Scaling이란?을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeScalingPlanResources를 참조하세요.

describe-scaling-plans

다음 코드 예시는 describe-scaling-plans의 사용 방법을 보여 줍니다.

AWS CLI

조정 계획을 설명하는 방법

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 AWS Auto Scaling이란?을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeScalingPlans를 참조하세요.

`get-scaling-plan-resource-forecast-data`

다음 코드 예시는 `get-scaling-plan-resource-forecast-data`의 사용 방법을 보여 줍니다.

AWS CLI

로드 예측 데이터를 검색하는 방법

이 예제에서는 지정된 조정 계획과 연결된 확장 가능 리소스(Auto Scaling 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 설명서의 AWS Auto Scaling이란을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetScalingPlanResourceForecastData를 참조하세요.

`update-scaling-plan`

다음 코드 예시는 `update-scaling-plan`의 사용 방법을 보여 줍니다.

AWS CLI

조정 계획을 업데이트하는 방법

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 AWS Auto Scaling이란?을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateScalingPlan을 참조하세요.

AWS CLI를 사용한 AWS Backup 예시

다음 코드 예시에서는 AWS Backup에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

작업

`create-backup-plan`

다음 코드 예시는 `create-backup-plan`의 사용 방법을 보여 줍니다.

AWS CLI

백업 계획을 생성하는 방법

다음 `create-backup-plan` 예제에서는 35일 보존을 사용하는 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 AWS 백업 개발자 안내서의 백업 계획 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateBackupPlan을 참조하세요.

`create-backup-vault`

다음 코드 예시는 `create-backup-vault`의 사용 방법을 보여 줍니다.

AWS CLI

백업 볼트를 생성하는 방법

다음 `create-backup-vault` 예제에서는 지정된 이름의 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 AWS 백업 개발자 안내서의 백업 볼트 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateBackupVault를 참조하세요.

get-backup-plan-from-template

다음 코드 예시는 `get-backup-plan-from-template`의 사용 방법을 보여 줍니다.

AWS CLI

템플릿에서 기존 백업 계획을 가져오는 방법

다음 `get-backup-plan-from-template` 예제에서는 35일 보존을 사용하는 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 AWS 백업 개발자 안내서의 백업 계획 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetBackupPlanFromTemplate을 참조하세요.

get-backup-plan

다음 코드 예시는 get-backup-plan의 사용 방법을 보여 줍니다.

AWS CLI

백업 계획의 세부 정보를 가져오는 방법

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 AWS 백업 개발자 안내서의 백업 계획 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetBackupPlan을 참조하세요.

list-backup-jobs

다음 코드 예시는 list-backup-jobs의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 모든 백업 작업을 나열하는 방법

다음 list-backup-jobs 예제에서는 AWS 계정에서 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 AWS 백업 개발자 안내서의 백업 생성을 참조하세요.

예제 2: 완료된 백업 작업을 나열하는 방법

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 AWS 백업 개발자 안내서의 백업 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListBackupJobs를 참조하세요.

AWS CLI를 사용한 AWS Batch 예시

다음 코드 예시에서는 AWS Batch에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`cancel-job`

다음 코드 예시는 `cancel-job`의 사용 방법을 보여 줍니다.

AWS CLI

작업을 취소하는 방법

이 예제에서는 지정된 작업 ID가 있는 작업을 취소합니다.

명령:

```
aws batch cancel-job --job-id bcf0b186-a532-4122-842e-2ccab8d54efb --
reason "Cancelling job."
```

- API 세부 정보는 AWS CLI 명령 참조의 CancelJob을 참조하세요.

create-compute-environment

다음 코드 예시는 `create-compute-environment`의 사용 방법을 보여 줍니다.

AWS CLI

온디맨드 인스턴스를 사용하여 관리형 컴퓨팅 환경을 생성하는 방법

이 예제에서는 온디맨드 방식으로 시작되는 특정 C4 인스턴스 유형을 사용하여 관리형 컴퓨팅 환경을 생성합니다. 이 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 생성하는 방법

이 예제에서는 스팟 입찰 가격이 인스턴스 유형에 대한 온디맨드 가격의 20% 이하일 때 시작되는 M4 인스턴스 유형을 사용하여 관리형 컴퓨팅 환경을 생성합니다. 이 컴퓨팅 환경을 M4Spot이라고 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateComputeEnvironment를 참조하세요.

`create-job-queue`

다음 코드 예시는 `create-job-queue`의 사용 방법을 보여 줍니다.

AWS CLI

단일 컴퓨팅 환경을 사용하여 낮은 우선순위 작업 대기열을 생성하는 방법

이 예제에서는 M4Spot 컴퓨팅 환경을 사용하는 LowPriority라고 하는 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 개의 컴퓨팅 환경을 사용하여 높은 우선순위 작업 대기열을 생성하는 방법

이 예제에서는 순서가 1인 C4OnDemand 컴퓨팅 환경과 순서가 2인 M4Spot 컴퓨팅 환경을 사용하는 HighPriority라고 하는 작업 대기열을 생성합니다. 스케줄러는 먼저 C4OnDemand 컴퓨팅 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateJobQueue를 참조하세요.

delete-compute-environment

다음 코드 예시는 `delete-compute-environment`의 사용 방법을 보여 줍니다.

AWS CLI

컴퓨팅 환경을 삭제하는 방법

이 예제에서는 P2OnDemand 컴퓨팅 환경을 삭제합니다.

명령:

```
aws batch delete-compute-environment --compute-environment P2OnDemand
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteComputeEnvironment를 참조하세요.

delete-job-queue

다음 코드 예시는 delete-job-queue의 사용 방법을 보여 줍니다.

AWS CLI

대기열을 삭제하는 방법

이 예제에서는 GPGPU 작업 대기열을 삭제합니다.

명령:

```
aws batch delete-job-queue --job-queue GPGPU
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteJobQueue를 참조하세요.

deregister-job-definition

다음 코드 예시는 deregister-job-definition의 사용 방법을 보여 줍니다.

AWS CLI

작업 정의의 등록을 취소하는 방법

이 예제에서는 sleep10이라고 하는 작업 정의의 등록을 취소합니다.

명령:

```
aws batch deregister-job-definition --job-definition sleep10
```

- API 세부 정보는 AWS CLI 명령 참조의 DeregisterJobDefinition을 참조하세요.

describe-compute-environments

다음 코드 예시는 describe-compute-environments의 사용 방법을 보여 줍니다.

AWS CLI

컴퓨팅 환경을 설명하는 방법

이 예시에서는 P2Ondemand 컴퓨팅 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"
```

```
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeComputeEnvironments를 참조하세요.

describe-job-definitions

다음 코드 예시는 describe-job-definitions의 사용 방법을 보여 줍니다.

AWS CLI

활성 작업 정의를 설명하는 방법

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
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeJobDefinitions를 참조하세요.

describe-job-queues

다음 코드 예시는 `describe-job-queues`의 사용 방법을 보여 줍니다.

AWS CLI

작업 대기열을 설명하는 방법

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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeJobQueues를 참조하세요.

describe-jobs

다음 코드 예시는 describe-jobs의 사용 방법을 보여 줍니다.

AWS CLI

작업을 설명하는 방법

다음 describe-jobs 예제에서는 지정된 작업 ID가 있는 작업을 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
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeJobs를 참조하세요.

`list-jobs`

다음 코드 예시는 `list-jobs`의 사용 방법을 보여 줍니다.

AWS CLI

실행 중인 작업을 나열하는 방법

이 예제에서는 HighPriority 작업 대기열에서 실행 중인 작업을 나열합니다.

명령:

```
aws batch list-jobs --job-queue HighPriority
```

출력:

```
{
    "jobSummaryList": [
        {
            "jobName": "example",
            "jobId": "e66ff5fd-a1ff-4640-b1a2-0b0a142f49bb"
        }
    ]
}
```

제출된 작업을 나열하는 방법

이 예제에서는 HighPriority 작업 대기열에서 SUBMITTED 작업 상태인 작업을 나열합니다.

명령:

```
aws batch list-jobs --job-queue HighPriority --job-status SUBMITTED
```

출력:

```
{
    "jobSummaryList": [
        {
            "jobName": "example",
            "jobId": "68f0c163-fbd4-44e6-9fd1-25b14a434786"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListJobs를 참조하세요.

`register-job-definition`

다음 코드 예시는 `register-job-definition`의 사용 방법을 보여 줍니다.

AWS CLI

작업 정의를 등록하는 방법

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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 RegisterJobDefinition을 참조하세요.

`submit-job`

다음 코드 예시는 `submit-job`의 사용 방법을 보여 줍니다.

AWS CLI

작업을 제출하는 방법

이 예제에서는 예제라고 하는 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 SubmitJob을 참조하세요.

`terminate-job`

다음 코드 예시는 `terminate-job`의 사용 방법을 보여 줍니다.

AWS CLI

작업을 종료하는 방법

이 예제에서는 지정된 작업 ID가 있는 작업을 종료합니다.

명령:

```
aws batch terminate-job --job-id 61e743ed-35e4-48da-b2de-5c8333821c84 --
reason "Terminating job."
```

- API 세부 정보는 AWS CLI 명령 참조의 TerminateJob을 참조하세요.

`update-compute-environment`

다음 코드 예시는 `update-compute-environment`의 사용 방법을 보여 줍니다.

AWS CLI

컴퓨팅 환경을 업데이트하는 방법

이 예제에서는 P2OnDemand 컴퓨팅 환경이 삭제될 수 있도록 이 환경을 비활성화합니다.

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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateComputeEnvironment를 참조하세요.

`update-job-queue`

다음 코드 예시는 `update-job-queue`의 사용 방법을 보여 줍니다.

AWS CLI

작업 대기열을 업데이트하는 방법

이 예제에서는 작업 대기열이 삭제될 수 있도록 이 작업 대기열을 비활성화합니다.

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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateJobQueue를 참조하세요.

AWS CLI를 사용한 AWS Budgets 예시

다음 코드 예시에서는 AWS Budgets에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

작업

`create-budget`

다음 코드 예시는 `create-budget`의 사용 방법을 보여 줍니다.

AWS CLI

비용 및 사용량 예산을 생성하는 방법

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.json`의 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.json`의 콘텐츠:

```
[
    {
        "Notification": {
            "ComparisonOperator": "GREATER_THAN",
            "NotificationType": "ACTUAL",
            "Threshold": 80,
            "ThresholdType": "PERCENTAGE"
        },
        "Subscribers": [
```

```
            {
                "Address": "example@example.com",
                "SubscriptionType": "EMAIL"
            }
        ]
    }
]
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateBudget을 참조하세요.

`create-notification`

다음 코드 예시는 `create-notification`의 사용 방법을 보여 줍니다.

AWS CLI

지정된 비용 및 사용량 예산에 대한 알림을 생성하는 방법

이 예제에서는 지정된 비용 및 사용량 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
```

- API 세부 정보는 AWS CLI 명령 참조의CreateNotification을 참조하세요.

`create-subscriber`

다음 코드 예시는 `create-subscriber`의 사용 방법을 보여 줍니다.

AWS CLI

비용 및 사용량 예산과 연결된 알림에 대한 구독자를 생성하는 방법

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
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateSubscriber를 참조하세요.

`delete-budget`

다음 코드 예시는 `delete-budget`의 사용 방법을 보여 줍니다.

AWS CLI

비용 및 사용량 예산을 삭제하는 방법

이 예제에서는 지정된 비용 및 사용량 예산을 삭제합니다.

명령:

```
aws budgets delete-budget --account-id 111122223333 --budget-name "Example Budget"
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteBudget을 참조하세요.

`delete-notification`

다음 코드 예시는 `delete-notification`의 사용 방법을 보여 줍니다.

AWS CLI

예산에서 알림을 삭제하는 방법

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
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteNotification을 참조하세요.

`delete-subscriber`

다음 코드 예시는 `delete-subscriber`의 사용 방법을 보여 줍니다.

AWS CLI

알림에서 구독자를 삭제하는 방법

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
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteSubscriber를 참조하세요.

describe-budget

다음 코드 예시는 describe-budget의 사용 방법을 보여 줍니다.

AWS CLI

계정과 연결된 예산을 검색하는 방법

이 예제에서는 지정된 비용 및 사용량 예산을 검색합니다.

명령:

```
aws budgets describe-budget --account-id 111122223333 --budget-name "Example Budget"
```

출력:

```
{
   "Budget": {
       "CalculatedSpend": {
           "ForecastedSpend": {
               "Amount": "2641.54800000000022919266484677791595458984375",
               "Unit": "USD"
           },
           "ActualSpend": {
               "Amount": "604.4560000000000172803993336856365203857421875",
               "Unit": "USD"
```

```
            }
        },
        "BudgetType": "COST",
        "BudgetLimit": {
            "Amount": "100",
            "Unit": "USD"
        },
        "BudgetName": "Example Budget",
        "CostTypes": {
            "IncludeOtherSubscription": true,
            "IncludeUpfront": true,
            "IncludeRefund": true,
            "UseBlended": false,
            "IncludeDiscount": true,
            "UseAmortized": false,
            "IncludeTax": true,
            "IncludeCredit": true,
            "IncludeSupport": true,
            "IncludeRecurring": true,
            "IncludeSubscription": true
        },
        "TimeUnit": "MONTHLY",
        "TimePeriod": {
            "Start": 1477958399.0,
            "End": 3706473600.0
        },
        "CostFilters": {
            "AZ": [
                "us-east-1"
            ]
        }
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeBudget을 참조하세요.

describe-budgets

다음 코드 예시는 `describe-budgets`의 사용 방법을 보여 줍니다.

AWS CLI

계정과 연결된 예산들을 검색하는 방법

이 예제에서는 계정에 대한 비용 및 사용량 예산들을 검색합니다.

명령:

```
aws budgets describe-budgets --account-id 111122223333 --max-results 20
```

출력:

```
{
   "Budgets": [
       {
           "CalculatedSpend": {
               "ForecastedSpend": {
                   "Amount": "2641.54800000000022919266484677791595458984375",
                   "Unit": "USD"
               },
               "ActualSpend": {
                   "Amount": "604.4560000000000172803993336856365203857421875",
                   "Unit": "USD"
               }
           },
           "BudgetType": "COST",
           "BudgetLimit": {
               "Amount": "100",
               "Unit": "USD"
           },
           "BudgetName": "Example Budget",
           "CostTypes": {
               "IncludeOtherSubscription": true,
               "IncludeUpfront": true,
               "IncludeRefund": true,
               "UseBlended": false,
               "IncludeDiscount": true,
               "UseAmortized": false,
               "IncludeTax": true,
               "IncludeCredit": true,
               "IncludeSupport": true,
               "IncludeRecurring": true,
               "IncludeSubscription": true
           },
           "TimeUnit": "MONTHLY",
           "TimePeriod": {
               "Start": 1477958399.0,
```

```
                "End": 3706473600.0
            },
            "CostFilters": {
                "AZ": [
                    "us-east-1"
                ]
            }
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeBudgets를 참조하세요.

describe-notifications-for-budget

다음 코드 예시는 describe-notifications-for-budget의 사용 방법을 보여 줍니다.

AWS CLI

예산에 대한 알림을 검색하는 방법

이 예제에서는 비용 및 사용량 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeNotificationsForBudget을 참조하세요.

describe-subscribers-for-notification

다음 코드 예시는 describe-subscribers-for-notification의 사용 방법을 보여 줍니다.

AWS CLI

예산 알림에 대한 구독자를 검색하는 방법

이 예제에서는 비용 및 사용량 예산 알림에 대한 구독자를 검색합니다.

명령:

```
aws budgets describe-subscribers-for-notification --
account-id 111122223333 --budget-name "Example Budget" --
notification NotificationType=ACTUAL,ComparisonOperator=GREATER_THAN,Threshold=80,ThresholdT
 --max-results 5
```

출력:

```
{
   "Subscribers": [
       {
           "SubscriptionType": "EMAIL",
           "Address": "example2@example.com"
       },
       {
           "SubscriptionType": "EMAIL",
           "Address": "example@example.com"
       }
   ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeSubscribersForNotification을 참조하세요.

update-budget

다음 코드 예시는 update-budget의 사용 방법을 보여 줍니다.

AWS CLI

비용 및 사용량 예산에 대한 예산을 교체하는 방법

이 예제에서는 비용 및 사용량 예산을 새 예산으로 교체합니다.

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"
 }
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateBudget을 참조하세요.

update-notification

다음 코드 예시는 update-notification의 사용 방법을 보여 줍니다.

AWS CLI

비용 및 사용량 예산에 대한 알림을 교체하는 방법

이 예제에서는 비용 및 사용량 예산에 대한 80% 알림을 90% 알림으로 교체합니다.

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
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateNotification을 참조하세요.

update-subscriber

다음 코드 예시는 `update-subscriber`의 사용 방법을 보여 줍니다.

AWS CLI

비용 및 사용량 예산에 대한 구독자를 교체하는 방법

이 예제에서는 비용 및 사용량 예산에 대한 구독자를 교체합니다.

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
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateSubscriber를 참조하세요.

AWS CLI를 사용한 Amazon Chime 예시

다음 코드 예시에서는 Amazon Chime에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`associate-phone-number-with-user`

다음 코드 예시는 `associate-phone-number-with-user`의 사용 방법을 보여 줍니다.

AWS CLI

전화번호를 사용자와 연결하는 방법

다음 `associate-phone-number-with-user` 예시에서는 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 Managing User Phone Numbers 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AssociatePhoneNumberWithUser를 참조하세요.

`associate-signin-delegate-groups-with-account`

다음 코드 예시는 `associate-signin-delegate-groups-with-account`의 사용 방법을 보여 줍니다.

AWS CLI

로그인 위임 그룹을 연결하는 방법

다음 `associate-signin-delegate-groups-with-account` 예시에서는 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 Managing User Access and Permissions' 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AssociateSigninDelegateGroupsWithAccount를 참조하세요.

`batch-create-room-membership`

다음 코드 예시는 `batch-create-room-membership`의 사용 방법을 보여 줍니다.

AWS CLI

여러 룸 멤버십을 생성하는 방법

다음 `batch-create-room-membership` 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 BatchCreateRoomMembership을 참조하세요.

batch-delete-phone-number

다음 코드 예시는 batch-delete-phone-number의 사용 방법을 보여 줍니다.

AWS CLI

여러 전화번호를 삭제하는 방법

다음 batch-delete-phone-number 예시에서는 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 Working with Phone Numbers 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 BatchDeletePhoneNumber를 참조하세요.

batch-suspend-user

다음 코드 예시는 batch-suspend-user의 사용 방법을 보여 줍니다.

AWS CLI

여러 사용자를 일시 중지하는 방법

다음 `batch-suspend-user` 예시에서는 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": []
}
```

- API 세부 정보는 AWS CLI 명령 참조의 BatchSuspendUser를 참조하세요.

`batch-unsuspend-user`

다음 코드 예시는 `batch-unsuspend-user`의 사용 방법을 보여 줍니다.

AWS CLI

여러 사용자의 일시 중지를 취소하는 방법

다음 `batch-unsuspend-user` 예시에서는 지정된 Amazon Chime 계정의 나열된 사용자에 대한 모든 이전의 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": []
```

```
}
```

- API 세부 정보는 AWS CLI 명령 참조의 BatchUnsuspendUser를 참조하세요.

batch-update-phone-number

다음 코드 예시는 batch-update-phone-number의 사용 방법을 보여 줍니다.

AWS CLI

여러 전화번호 제품 유형을 동시에 업데이트하는 방법

다음 batch-update-phone-number 예시에서는 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 호출 이름을 동시에 업데이트하는 방법

다음 batch-update-phone-number 예시에서는 지정된 모든 전화번호에 대한 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 Working with Phone Numbers 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 BatchUpdatePhoneNumber를 참조하세요.

batch-update-user

다음 코드 예시는 batch-update-user의 사용 방법을 보여 줍니다.

AWS CLI

단일 명령으로 여러 사용자를 업데이트하는 방법

다음 batch-update-user 예시에서는 지정된 Amazon Chime 계정에서 나열된 각 사용자에 대해 LicenseType을 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": []
}
```

- API 세부 정보는 AWS CLI 명령 참조의 BatchUpdateUser를 참조하세요.

create-account

다음 코드 예시는 create-account의 사용 방법을 보여 줍니다.

AWS CLI

계정을 생성하려면

다음 create-account 예시에서는 관리자의 AWS 계정에서 Amazon Chime 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 Getting Started 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateAccount를 참조하세요.

create-bot

다음 코드 예시는 `create-bot`의 사용 방법을 보여 줍니다.

AWS CLI

Amazon Chime 봇을 생성하는 방법

다음 `create-bot` 예시에서는 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 Integrate a Chat Bot with Amazon Chime 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateBot을 참조하세요.

`create-phone-number-order`

다음 코드 예시는 `create-phone-number-order`의 사용 방법을 보여 줍니다.

AWS CLI

전화번호 명령을 생성하는 방법

다음 `create-phone-number-order` 예시에서는 지정된 전화번호에 대한 전화번호 명령을 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 Working with Phone Numbers 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreatePhoneNumberOrder를 참조하세요.

`create-room-membership`

다음 코드 예시는 `create-room-membership`의 사용 방법을 보여 줍니다.

AWS CLI

룸 멤버십을 생성하는 방법

다음 `create-room-membership` 예시에서는 지정된 사용자를 채팅룸에 채팅룸 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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateRoomMembership을 참조하세요.

`create-room`

다음 코드 예시는 `create-room`의 사용 방법을 보여 줍니다.

AWS CLI

채팅룸을 만들려면

다음 `create-room` 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateRoom을 참조하세요.

create-user

다음 코드 예시는 create-user의 사용 방법을 보여 줍니다.

AWS CLI

공유 디바이스에 대한 사용자 프로파일을 생성하는 방법

다음 create-user 예시에서는 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 Preparing for Setup 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateUser를 참조하세요.

delete-account

다음 코드 예시는 delete-account의 사용 방법을 보여 줍니다.

AWS CLI

계정을 삭제하는 방법

다음 delete-account 예시에서는 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 Deleting Your Account 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteAccount를 참조하세요.

delete-phone-number

다음 코드 예시는 delete-phone-number의 사용 방법을 보여 줍니다.

AWS CLI

전화번호를 삭제하는 방법

다음 delete-phone-number 예시에서는 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 Working with Phone Numbers 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeletePhoneNumber를 참조하세요.

delete-room-membership

다음 코드 예시는 delete-room-membership의 사용 방법을 보여 줍니다.

AWS CLI

채팅룸의 멤버인 사용자를 제거하는 방법

다음 delete-room-membership 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteRoomMembership을 참조하세요.

delete-room

다음 코드 예시는 delete-room의 사용 방법을 보여 줍니다.

AWS CLI

채팅방을 삭제하는 방법

다음 delete-room 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteRoom을 참조하세요.

disassociate-phone-number-from-user

다음 코드 예시는 disassociate-phone-number-from-user의 사용 방법을 보여 줍니다.

AWS CLI

사용자와 전화번호의 연결을 해제하는 방법

다음 disassociate-phone-number-from-user 예시에서는 지정된 사용자에서 전화번호의 연결을 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 Managing User Phone Numbers 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisassociatePhoneNumberFromUser를 참조하세요.

disassociate-signin-delegate-groups-from-account

다음 코드 예시는 disassociate-signin-delegate-groups-from-account의 사용 방법을 보여 줍니다.

AWS CLI

로그인 위임 그룹의 연결을 해제하는 방법

다음 disassociate-signin-delegate-groups-from-account 예시에서는 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 Managing User Access and Permissions' 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisassociateSigninDelegateGroupsFromAccount를 참조하세요.

get-account-settings

다음 코드 예시는 get-account-settings의 사용 방법을 보여 줍니다.

AWS CLI

계정의 설정을 검색하는 방법

다음 get-account-settings 예시에서는 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 Managing Your Amazon Chime Accounts 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetAccountSettings를 참조하세요.

get-account

다음 코드 예시는 get-account의 사용 방법을 보여 줍니다.

AWS CLI

계정의 세부 정보를 검색하는 방법

다음 get-account 예시에서는 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 Managing Your Amazon Chime Accounts 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetAccount를 참조하세요.

get-bot

다음 코드 예시는 get-bot의 사용 방법을 보여 줍니다.

AWS CLI

봇에 대한 세부 정보를 검색하는 방법

다음 get-bot 예시에서는 지정된 봇의 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 Update Chat Bots 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetBot을 참조하세요.

get-global-settings

다음 코드 예시는 `get-global-settings`의 사용 방법을 보여 줍니다.

AWS CLI

전역 설정을 가져오는 방법

다음 `get-global-settings` 예시에서는 관리자의 AWS 계정과 연결된 Amazon Chime Business Calling 및 Amazon Chime Voice Connector에 대한 통화 세부 정보 레코드를 보관하는 데 사용되는 S3 버킷 이름을 검색합니다.

```
aws chime get-global-settings
```

출력:

```
{
    "BusinessCalling": {
        "CdrBucket": "s3bucket"
    },
    "VoiceConnector": {
        "CdrBucket": "s3bucket"
```

```
    }
}
```

자세한 내용은 Amazon Chime 관리 안내서의 Managing Global Settings 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetGlobalSettings를 참조하세요.

get-phone-number-order

다음 코드 예시는 get-phone-number-order의 사용 방법을 보여 줍니다.

AWS CLI

전화번호 명령에 대한 세부 정보를 가져오는 방법

다음 get-phone-number-order 예시에서는 지정된 전화번호 명령의 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 Working with Phone Numbers 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetPhoneNumberOrder를 참조하세요.

get-phone-number-settings

다음 코드 예시는 get-phone-number-settings의 사용 방법을 보여 줍니다.

AWS CLI

발신 호출 이름을 검색하는 방법

다음 get-phone-number-settings 예시에서는 발신 사용자의 AWS 계정에 대한 기본 발신 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 Working with Phone Numbers 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetPhoneNumberSettings를 참조하세요.

get-phone-number

다음 코드 예시는 get-phone-number의 사용 방법을 보여 줍니다.

AWS CLI

전화번호 세부 정보를 가져오는 방법

다음 get-phone-number 예시에서는 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 Working with Phone Numbers 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetPhoneNumber를 참조하세요.

get-room

다음 코드 예시는 get-room의 사용 방법을 보여 줍니다.

AWS CLI

채팅룸에 대한 세부 정보를 가져오는 방법

다음 get-room 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 GetRoom을 참조하세요.

get-user-settings

다음 코드 예시는 get-user-settings의 사용 방법을 보여 줍니다.

AWS CLI

사용자 설정을 검색하는 방법

다음 get-user-settings 예시에서는 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 Managing User Phone Numbers 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetUserSettings를 참조하세요.

get-user

다음 코드 예시는 get-user의 사용 방법을 보여 줍니다.

AWS CLI

사용자에 대한 세부 정보를 가져오는 방법

다음 get-user 예시에서는 지정된 사용자에 대한 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 Managing Users 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetUser를 참조하세요.

invite-users

다음 코드 예시는 invite-users의 사용 방법을 보여 줍니다.

AWS CLI

사용자를 Amazon Chime에 가입하도록 초대하는 방법

다음 invite-users 예시에서는 사용자를 지정된 Amazon Chime 계정에 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 Inviting and Suspending Users 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 InviteUsers를 참조하세요.

list-accounts

다음 코드 예시는 list-accounts의 사용 방법을 보여 줍니다.

AWS CLI

계정 목록을 가져오는 방법

다음 list-accounts 예시에서는 관리자의 AWS 계정에서 Amazon Chime 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 Managing Your Amazon Chime Accounts 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListAccounts를 참조하세요.

list-bots

다음 코드 예시는 `list-bots`의 사용 방법을 보여 줍니다.

AWS CLI

봇 목록을 검색하는 방법

다음 `list-bots` 예시에서는 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 Use Chat Bots with Amazon Chime 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListBots를 참조하세요.

list-phone-number-orders

다음 코드 예시는 `list-phone-number-orders`의 사용 방법을 보여 줍니다.

AWS CLI

전화번호 명령을 나열하는 방법

다음 `list-phone-number-orders` 예시에서는 Amazon Chime 관리자의 계정과 연결된 전화번호 명령을 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 Working with Phone Numbers 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListPhoneNumberOrders를 참조하세요.

list-phone-numbers

다음 코드 예시는 `list-phone-numbers`의 사용 방법을 보여 줍니다.

AWS CLI

Amazon Chime 계정의 전화번호를 나열하는 방법

다음 `list-phone-numbers` 예시에서는 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 Working with Phone Numbers 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForVault를 참조하세요.

`list-room-memberships`

다음 코드 예시는 `list-room-memberships`의 사용 방법을 보여 줍니다.

AWS CLI

룸 멤버십을 나열하는 방법

다음 `list-room-memberships` 예시에서는 지정된 채팅룸에 대한 멤버십 세부 정보의 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListRoomMemberships를 참조하세요.

list-rooms

다음 코드 예시는 `list-rooms`의 사용 방법을 보여 줍니다.

AWS CLI

채팅룸을 나열하는 방법

다음 `list-rooms` 예시에서는 지정된 계정에서 채팅룸의 목록을 표시합니다. 이 목록은 지정된 멤버가 속한 채팅룸만 나열하도록 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListRooms를 참조하세요.

list-users

다음 코드 예시는 `list-users`의 사용 방법을 보여 줍니다.

AWS CLI

계정 내 모든 사용자를 나열하는 방법

다음 `list-users` 예시에서는 지정된 Amazon Chime 계정에 대한 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 Managing Users 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListUsers를 참조하세요.

logout-user

다음 코드 예시는 logout-user의 사용 방법을 보여 줍니다.

AWS CLI

사용자를 로그아웃하는 방법

다음 logout-user 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 LogoutUser를 참조하세요.

regenerate-security-token

다음 코드 예시는 regenerate-security-token의 사용 방법을 보여 줍니다.

AWS CLI

보안 토큰을 재생성하는 방법

다음 regenerate-security-token 예시에서는 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 Authenticate Chat Bot Requests 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RegenerateSecurityToken을 참조하세요.

`reset-personal-pin`

다음 코드 예시는 `reset-personal-pin`의 사용 방법을 보여 줍니다.

AWS CLI

사용자의 개인 회의 PIN을 재설정하는 방법

다음 `reset-personal-pin` 예시에서는 지정된 사용자의 개인 회의 PIN을 재설정합니다.

```
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 Changing Personal Meeting PINs 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ResetPersonalPin을 참조하세요.

restore-phone-number

다음 코드 예시는 `restore-phone-number`의 사용 방법을 보여 줍니다.

AWS CLI

전화번호를 복원하는 방법

다음 `restore-phone-number` 예시에서는 지정된 전화번호를 삭제 대기열에서 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 Working with Phone Numbers 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RestorePhoneNumber를 참조하세요.

`search-available-phone-numbers`

다음 코드 예시는 `search-available-phone-numbers`의 사용 방법을 보여 줍니다.

AWS CLI

사용 가능한 전화번호를 검색하는 방법

다음 `search-available-phone-numbers` 예시에서는 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 Working with Phone Numbers 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 SearchAvailablePhoneNumbers를 참조하세요.

`update-account-settings`

다음 코드 예시는 `update-account-settings`의 사용 방법을 보여 줍니다.

AWS CLI

계정의 설정을 업데이트하는 방법

다음 `update-account-settings` 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateAccountSettings을 참조하세요.

`update-account`

다음 코드 예시는 `update-account`의 사용 방법을 보여 줍니다.

AWS CLI

계정을 업데이트하는 방법

다음 `update-account` 예시에서는 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 Renaming Your Account 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateAccount를 참조하세요.

update-bot

다음 코드 예시는 update-bot의 사용 방법을 보여 줍니다.

AWS CLI

봇을 업데이트하는 방법

다음 update-bot 예시에서는 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 Update Chat Bots 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateBot을 참조하세요.

update-global-settings

다음 코드 예시는 update-global-settings의 사용 방법을 보여 줍니다.

AWS CLI

전역 설정을 업데이트하는 방법

다음 update-global-settings 예시에서는 관리자의 AWS 계정과 연결된 Amazon Chime Business Calling 및 Amazon Chime Voice Connector에 대한 통화 세부 정보 레코드를 보관하는 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 Managing Global Settings 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateGlobalSettings를 참조하세요.

update-phone-number-settings

다음 코드 예시는 update-phone-number-settings의 사용 방법을 보여 줍니다.

AWS CLI

발신 호출 이름을 업데이트하는 방법

다음 update-phone-number-settings 예시에서는 관리자의 AWS 계정에 대한 기본 발신 호출 이름을 업데이트합니다.

```
aws chime update-phone-number-settings \
```

```
    --calling-name "myName"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Chime 관리 안내서의 Working with Phone Numbers 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdatePhoneNumberSettings를 참조하세요.

update-phone-number

다음 코드 예시는 update-phone-number의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 전화번호에 대한 제품 유형 업데이트

다음 update-phone-number 예시에서는 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"
```

```
    }
}
```

예시 2: 전화번호에 대한 발신 호출 이름 업데이트

다음 `update-phone-number` 예시에서는 지정된 전화번호에 대한 발신 호출 이름을 업데이트합니다.

aws chime 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 Working with Phone Numbers 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdatePhoneNumber를 참조하세요.

update-room-membership

다음 코드 예시는 `update-room-membership`의 사용 방법을 보여 줍니다.

AWS CLI

룸 멤버십을 업데이트하는 방법

다음 update-room-membership 예시에서는 지정된 채팅룸 멤버의 역할을 Administrator로 수정합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateRoomMembership을 참조하세요.

update-room

다음 코드 예시는 update-room의 사용 방법을 보여 줍니다.

AWS CLI

채팅룸을 업데이트하는 방법

다음 update-room 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateRoom을 참조하세요.

update-user-settings

다음 코드 예시는 update-user-settings의 사용 방법을 보여 줍니다.

AWS CLI

사용자 설정을 업데이트하는 방법

다음 update-user-settings 예시에서는 지정된 사용자가 수신 및 발신 통화를 수행하고 SMS 메시지를 보내고 받을 수 있도록 합니다.

```
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 Managing User Phone Numbers 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateUserSettings를 참조하세요.

`update-user`

다음 코드 예시는 `update-user`의 사용 방법을 보여 줍니다.

AWS CLI

사용자 세부 정보를 업데이트하는 방법

이 예시에서는 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateUser 섹션을 참조하세요.

AWS CLI를 사용한 Cloud Control API 예제

다음 코드 예제에서는 Cloud Control API에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여 줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`create-resource`

다음 코드 예시는 `create-resource`의 사용 방법을 보여 줍니다.

AWS CLI

리소스를 생성하려면

다음 `create-resource` 예제에서는 보존 기간이 168시간이고 샤드 수가 3인 ResourceExample 라는 이름의 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 Creating a resource를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateResource를 참조하세요.

`delete-resource`

다음 코드 예시는 `delete-resource`의 사용 방법을 보여 줍니다.

AWS CLI

리소스를 삭제하는 방법

다음 `delete-resource` 예제에서는 AWS 계정에서 식별자 ResourceExample이 있는 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 Deleting a resource를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteResource를 참조하세요.

`get-resource-request-status`

다음 코드 예시는 `get-resource-request-status`의 사용 방법을 보여 줍니다.

AWS CLI

리소스 요청의 상태 정보를 가져오려면

다음 `get-resource-request-status` 예제는 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 Managing resource operation requests를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetResourceRequestStatus를 참조하세요.

get-resource

다음 코드 예시는 get-resource의 사용 방법을 보여 줍니다.

AWS CLI

리소스의 현재 상태를 가져오려면

다음 get-resource 예제는 이름이 ResourceExample인 AWS::Kinesis::Stream 리소스의 현재 상태를 반환합니다.

```
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 Reading a resource's current state를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetResources를 참조하세요.

`list-resource-requests`

다음 코드 예시는 `list-resource-requests`의 사용 방법을 보여 줍니다.

AWS CLI

활성 리소스 작업 요청을 나열하려면

다음 `list-resource-requests` 예제에서는 AWS 계정에서 실패한 CREATE 및 UPDATE 작업에 대한 리소스 요청을 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 Managing resource operation requests를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListResourceRequests를 참조하세요.

list-resources

다음 코드 예시는 list-resources의 사용 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 Discovering resources를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListResources를 참조하세요.

update-resource

다음 코드 예시는 update-resource의 사용 방법을 보여 줍니다.

AWS CLI

기존 리소스의 속성을 업데이트하려면

다음 `update-resource` 예제에서는 이름이 ExampleLogGroup인 AWS::Logs::LogGroup 리소스의 보존 정책을 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 Updating a resource를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateResource를 참조하세요.

AWS CLI를 사용한 AWS Cloud Map 예시

다음 코드 예시에서는 AWS Cloud Map에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`create-http-namespace`

다음 코드 예시는 `create-http-namespace`의 사용 방법을 보여 줍니다.

AWS CLI

HTTP 네임스페이스 생성

다음 `create-http-namespace` 예시에서는 HTTP 네임스페이스 `example.com`을 생성합니다.

```
aws servicediscovery create-http-namespace \
    --name example.com \
    --creator-request-id example-request-id
```

출력:

```
{
    "OperationId": "gv4g5meo7ndmeh4fqskygvk23d2fijwa-k9302yzd"
}
```

작업이 성공했는지 확인하려면 `get-operation`을 실행합니다. 자세한 내용은 get-operation을 참조하세요.

네임스페이스 생성에 대한 자세한 내용은 AWS Cloud Map 개발자 안내서의 애플리케이션 서비스를 그룹화하기 위한 AWS Cloud Map 네임스페이스 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateHttpNamespace를 참조하세요.

`create-private-dns-namespace`

다음 코드 예시는 `create-private-dns-namespace`의 사용 방법을 보여 줍니다.

AWS CLI

프라이빗 DNS 네임스페이스 생성

다음 `create-private-dns-namespace` 예시에서는 프라이빗 DNS 네임스페이스를 생성합니다.

```
aws servicediscovery create-private-dns-namespace \
    --name example.com \
    --vpc vpc-1c56417b
```

출력:

```
{
    "OperationId": "gv4g5meo7ndmeh4fqskygvk23d2fijwa-k9302yzd"
}
```

작업이 성공했는지 확인하려면 `get-operation`을 실행합니다. 자세한 내용은 get-operation을 참조하세요.

자세한 내용은 AWS Cloud Map 개발자 안내서의 네임스페이스 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreatePrivateDnsNamespace를 참조하세요.

`create-public-dns-namespace`

다음 코드 예시는 `create-public-dns-namespace`의 사용 방법을 보여 줍니다.

AWS CLI

퍼블릭 DNS 네임스페이스 생성

다음 `create-public-dns-namespace` 예시에서는 퍼블릭 DNS 네임스페이스 `example.com`을 생성합니다.

```
aws servicediscovery create-public-dns-namespace \
    --name example-public-dns.com \
    --creator-request-id example-public-request-id \
    --properties DnsProperties={SOA={TTL=60}}
```

출력:

```
{
    "OperationId": "gv4g5meo7ndmeh4fqskygvk23d2fijwa-k9302yzd"
}
```

작업이 성공했는지 확인하기 위해 를 실행할 수 있습니다`get-operation`.

네임스페이스 생성에 대한 자세한 내용은 AWS Cloud Map 개발자 안내서의 애플리케이션 서비스를 그룹화하기 위한 AWS Cloud Map 네임스페이스 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreatePublicDnsNamespace를 참조하세요.

create-service

다음 코드 예시는 create-service의 사용 방법을 보여 줍니다.

AWS CLI

서비스 생성

다음 create-service 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateService를 참조하세요.

delete-namespace

다음 코드 예시에서는 delete-namespace의 사용 방법을 보여줍니다.

AWS CLI

네임스페이스 삭제

다음 delete-namespace 예시에서는 네임스페이스를 삭제합니다.

```
aws servicediscovery delete-namespace \
    --id ns-ylexjili4cdxy3xm
```

출력:

```
{
    "OperationId": "gv4g5meo7ndmeh4fqskygvk23d2fijwa-k98y6drk"
}
```

작업이 성공했는지 확인하려면 get-operation을 실행합니다. 자세한 내용은 get-operation을 참조하세요.

자세한 내용은 AWS Cloud Map 개발자 안내서의 네임스페이스 삭제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteNamespace를 참조하세요.

delete-service-attributes

다음 코드 예시는 delete-service-attributes의 사용 방법을 보여 줍니다.

AWS CLI

서비스 속성 삭제

다음 delete-service-attributes 예시에서는 지정된 서비스와 연결된 Port 키가 있는 서비스 속성을 삭제합니다.

```
aws servicediscovery delete-service-attributes \
    --service-id srv-e4anhexample0004 \
    --attributes Port
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Cloud Map 개발자 안내서의 네임스페이스 삭제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteServiceAttributes를 참조하세요.

delete-service

다음 코드 예시는 `delete-service`의 사용 방법을 보여 줍니다.

AWS CLI

서비스 삭제

다음 `delete-service` 예시에서는 서비스를 삭제합니다.

```
aws servicediscovery delete-service \
    --id srv-p5zdwlg5uvvzjita
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Cloud Map 개발자 안내서의 서비스 삭제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteService를 참조하세요.

deregister-instance

다음 코드 예시에서는 `deregister-instance`의 사용 방법을 보여줍니다.

AWS CLI

서비스 인스턴스 등록 취소

다음 `deregister-instance` 예시에서는 서비스 인스턴스의 등록을 취소합니다.

```
aws servicediscovery deregister-instance \
    --service-id srv-p5zdwlg5uvvzjita \
    --instance-id myservice-53
```

출력:

```
{
    "OperationId": "4yejorelbukcjzpnr6tlmrghsjwpngf4-k98rnaiq"
}
```

작업이 성공했는지 확인하려면 `get-operation`을 실행합니다. 자세한 내용은 get-operation을 참조하세요.

자세한 내용은 AWS Cloud Map 개발자 안내서의 서비스 인스턴스 등록 취소를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeregisterInstance를 참조하세요.

`discover-instances-revision`

다음 코드 예시는 `discover-instances-revision`의 사용 방법을 보여 줍니다.

AWS CLI

인스턴스 개정 검색

다음 `discover-instances-revision` 예시에서는 인스턴스의 증가 개정을 검색합니다.

```
aws servicediscovery discover-instances-revision \
    --namespace-name example.com \
    --service-name myservice
```

출력:

```
{
    "InstancesRevision": 123456
}
```

자세한 내용은 AWS Cloud Map 개발자 안내서의 AWS Cloud Map 서비스 인스턴스를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DiscoverInstancesRevision을 참조하세요.

`discover-instances`

다음 코드 예시는 `discover-instances`의 사용 방법을 보여 줍니다.

AWS CLI

등록된 인스턴스 검색

다음 discover-instances 예시에서는 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 내용은 AWS Cloud Map 개발자 안내서의 AWS Cloud Map 서비스 인스턴스를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DiscoverInstances를 참조하세요.

get-instance

다음 코드 예시는 get-instance의 사용 방법을 보여 줍니다.

AWS CLI

인스턴스 세부 정보 가져오기

다음 get-instance 예시에서는 서비스의 속성을 가져옵니다.

```
aws servicediscovery get-instance \
    --service-id srv-e4anhexample0004
    --instance-id i-abcd1234
```

출력:

```
{
    "Instances": {
        "Id": "arn:aws:servicediscovery:us-west-2:111122223333;:service/srv-e4anhexample0004",
        "Attributes": {
            "AWS_INSTANCE_IPV4": "192.0.2.44",
            "AWS_INSTANCE_PORT": "80",
            "color": "green",
            "region": "us-west-2",
            "stage": "beta"
        }
    }
}
```

자세한 내용은 AWS Cloud Map 개발자 안내서의 AWS Cloud Map 서비스 인스턴스를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetInstance 섹션을 참조하세요.

get-instances-health-status

다음 코드 예시는 `get-instances-health-status`의 사용 방법을 보여 줍니다.

AWS CLI

서비스와 연결된 인스턴스의 상태 가져오기

다음 `get-instances-health-status` 예시에서는 지정된 서비스와 연결된 인스턴스의 상태를 가져옵니다.

```
aws servicediscovery get-instances-health-status \
    --service-id srv-e4anhexample0004
```

출력:

```
{
```

```
    "Status": {
        "i-abcd1234": "HEALTHY",
        "i-abcd1235": "UNHEALTHY"
    }
}
```

자세한 내용은 AWS Cloud Map 개발자 안내서의 AWS Cloud Map 서비스 인스턴스를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetInstancesHealthStatus를 참조하세요.

get-namespace

다음 코드 예시는 get-namespace의 사용 방법을 보여 줍니다.

AWS CLI

네임스페이스 세부 정보 가져오기

다음 get-namespace 예시에서는 지정된 네임스페이스에 대한 정보를 검색합니다.

```
aws servicediscovery get-namespace \
    --id ns-e4anhexample0004
```

출력:

```
{
    "Namespaces": {
        "Arn": "arn:aws:servicediscovery:us-west-2:123456789012:namespace/ns-
e4anhexample0004",
        "CreateDate": "20181118T211712Z",
        "CreatorRequestId": "example-creator-request-id-0001",
        "Description": "Example.com AWS Cloud Map HTTP Namespace",
        "Id": "ns-e4anhexample0004",
        "Name": "example-http.com",
        "Properties": {
            "DnsProperties": {},
            "HttpProperties": {
                "HttpName": "example-http.com"
            }
        },
        "Type": "HTTP"
    }
```

```
}
```

자세한 내용은 AWS Cloud Map 개발자 안내서의 AWS Cloud Map 네임스페이스를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetNamespace를 참조하세요.

get-operation

다음 코드 예시는 get-operation의 사용 방법을 보여 줍니다.

AWS CLI

작업 결과 가져오기

다음 get-operation 예시에서는 네임스페이스 생성 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 내용은 AWS Cloud Map 개발자 안내서의 애플리케이션 서비스를 그룹화하기 위한 AWS Cloud Map 네임스페이스 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetOperation 섹션을 참조하세요.

get-service-attributes

다음 코드 예시는 get-service-attributes의 사용 방법을 보여 줍니다.

AWS CLI

서비스 속성 가져오기

다음 get-service-attributes 예시에서는 서비스의 속성을 가져옵니다.

```
aws servicediscovery get-service-attributes \
    --service-id srv-e4anhexample0004
```

출력:

```
{
    "ServiceAttributes": {
        "ServiceArn": "arn:aws:servicediscovery:us-west-2:111122223333;:service/srv-e4anhexample0004",
        "Attributes": {
            "Port": "80"
        }
    }
}
```

자세한 내용은 AWS Cloud Map 개발자 안내서의 AWS Cloud Map 서비스를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetServiceAttributes를 참조하세요.

get-service

다음 코드 예시는 get-service의 사용 방법을 보여 줍니다.

AWS CLI

서비스 설정 가져오기

다음 get-service 예시에서는 지정된 서비스의 설정을 가져옵니다.

```
aws servicediscovery get-service \
    --id srv-e4anhexample0004
```

출력:

```
{
    "Service": {
```

```
        "Id": "srv-e4anhexample0004",
        "Arn": "arn:aws:servicediscovery:us-west-2:111122223333:service/srv-
e4anhexample0004",
        "Name": "test-service",
        "NamespaceId": "ns-e4anhexample0004",
        "DnsConfig": {},
        "Type": "HTTP",
        "CreateDate": "2025-02-24T10:59:02.905000-06:00",
        "CreatorRequestId": "3f50f9d9-b14c-482e-a556-d2a22fe6106d"
    }
}
```

자세한 내용은 AWS Cloud Map 개발자 안내서의 AWS Cloud Map 서비스를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetService를 참조하세요.

list-instances

다음 코드 예시는 `list-instances`의 사용 방법을 보여 줍니다.

AWS CLI

서비스 인스턴스 나열

다음 `list-instances` 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListInstances를 참조하세요.

list-namespaces

다음 코드 예시에서는 `list-namespaces`의 사용 방법을 보여줍니다.

AWS CLI

네임스페이스 나열

다음 `list-namespaces` 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListNamespaces를 참조하세요.

list-operations

다음 코드 예시는 list-operations의 사용 방법을 보여 줍니다.

AWS CLI

지정된 기준을 충족하는 작업 나열

다음 list-operations 예시에서는 PENDING 또는 SUCCESS 상태의 작업을 나열합니다.

```
aws servicediscovery list-operations \
    --service-id srv-e4anhexample0004 \
```

```
    --filters Name=STATUS,Condition=IN,Values=PENDING,SUCCESS
```

출력:

```
{
    "Operations": [
        {
            "Id": "76yy8ovhpdz0plmjzbsnqgnrqvpv2qdt-kexample",
            "Status": "SUCCESS"
        },
        {
            "Id": "prysnyzpji3u2ciy45nke83x2zanl7yk-dexample",
            "Status": "SUCCESS"
        },
        {
            "Id": "ko4ekftir7kzlbechsh7xvcdgcpk66gh-7example",
            "Status": "PENDING"
        }
    ]
}
```

자세한 내용은 AWS Cloud Map 개발자 안내서에 있는 AWS Cloud Map이란 무엇인가요?를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListOperations 섹션을 참조하세요.

list-services

다음 코드 예시는 list-services의 사용 방법을 보여 줍니다.

AWS CLI

서비스 나열

다음 list-services 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListServices를 참조하세요.

list-tags-for-resource

다음 코드 예시는 `list-tags-for-resource`의 사용 방법을 보여 줍니다.

AWS CLI

지정된 리소스와 연결된 태그 나열

다음 `list-tags-for-resource` 예시에서는 지정된 리소스의 태그를 나열합니다.

```
aws servicediscovery list-tags-for-resource \
    --resource-arn arn:aws:servicediscovery:us-west-2:123456789012:namespace/ns-
e4anhexample0004
```

출력:

```
{
    "Tags": [
```

```
        {
            "Key": "Project",
            "Value": "Zeta"
        },
        {
            "Key": "Department",
            "Value": "Engineering"
        }
    ]
}
```

자세한 내용은 AWS Cloud Map 개발자 안내서의 AWS Cloud Map 리소스 태그 지정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource를 참조하세요.

register-instance

다음 코드 예시는 register-instance의 사용 방법을 보여 줍니다.

AWS CLI

서비스 인스턴스 등록

다음 register-instance 예시에서는 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 확인하려면 get-operation을 실행합니다. 자세한 내용은 get-operation을 참조하세요.

자세한 내용은 AWS Cloud Map 개발자 안내서의 인스턴스 등록을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [RegisterInstance]를 참조하세요.

`tag-resource`

다음 코드 예시는 `tag-resource`의 사용 방법을 보여 줍니다.

AWS CLI

지정된 리소스와 태그 연결

다음 `tag-resource` 예시에서는 지정된 네임스페이스가 있고 값이 `Engineering`인 `Department` 태그를 연결합니다.

```
aws servicediscovery tag-resource \
    --resource-arn arn:aws:servicediscovery:us-west-2:123456789012:namespace/ns-
e4anhexample0004 \
    --tags Key=Department, Value=Engineering
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Cloud Map 개발자 안내서의 AWS Cloud Map 리소스 태그 지정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

`untag-resource`

다음 코드 예시는 `untag-resource`의 사용 방법을 보여 줍니다.

AWS CLI

지정된 리소스에서 태그 제거

다음 `untag-resource` 예시에서는 지정된 네임스페이스에서 `Department` 태그를 제거합니다.

```
aws servicediscovery untag-resource \
    --resource-arn arn:aws:servicediscovery:us-west-2:123456789012:namespace/ns-
e4anhexample0004 \
    --tags Key=Department, Value=Engineering
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Cloud Map 개발자 안내서의 AWS Cloud Map 리소스 태그 지정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

update-http-namespace

다음 코드 예시는 update-http-namespace의 사용 방법을 보여 줍니다.

AWS CLI

HTTP 네임스페이스 업데이트

다음 update-http-namespace 예시에서는 지정된 HTTP 네임스페이스의 설명을 업데이트합니다.

```
aws servicediscovery update-http-namespace \
    --id ns-vh4nbmEXAMPLE \
    --updater-request-id example-request-id \
    --namespace Description="The updated namespace description."
```

출력:

```
{
    "OperationId": "gv4g5meo7ndmeh4fqskygvk23d2fijwa-k9302yzd"
}
```

작업이 성공했는지 확인하려면 get-operation을 실행합니다. 자세한 내용은 get-operation을 참조하세요.

자세한 내용은 AWS Cloud Map 개발자 안내서의 AWS Cloud Map 네임스페이스를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateHttpNamespace를 참조하세요.

update-instance-custom-health-status

다음 코드 예시는 update-instance-custom-health-status의 사용 방법을 보여 줍니다.

AWS CLI

사용자 지정 상태 확인 업데이트

다음 `update-instance-custom-health-status` 예시에서는 지정된 서비스 및 예시 서비스 인스턴스에 대한 사용자 지정 상태 확인의 상태를 `HEALTHY`로 업데이트합니다.

```
aws servicediscovery update-instance-custom-health-status \
    --service-id srv-e4anhexample0004 \
    --instance-id example \
    --status HEALTHY
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Cloud Map 개발자 안내서의 [AWS Cloud Map 서비스 상태 확인 구성]을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [UpdateInstanceCustomHealthStatus]를 참조하세요.

update-private-dns-namespace

다음 코드 예시는 `update-private-dns-namespace`의 사용 방법을 보여 줍니다.

AWS CLI

프라이빗 DNS 네임스페이스 업데이트

다음 `update-private-dns-namespace` 예시에서는 프라이빗 DNS 네임스페이스의 설명을 업데이트합니다.

```
aws servicediscovery update-private-dns-namespace \
    --id ns-bk3aEXAMPLE \
    --updater-request-id example-private-request-id \
    --namespace Description="The updated namespace description."
```

출력:

```
{
    "OperationId": "gv4g5meo7ndmeh4fqskygvk23d2fijwa-k9302yzd"
}
```

작업이 성공했는지 확인하기 위해 를 실행할 수 있습니다`get-operation`.

자세한 내용은 AWS Cloud Map 개발자 안내서의 [AWS Cloud Map 네임스페이스]를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [UpdatePrivateDnsNamespace]를 참조하세요.

update-public-dns-namespace

다음 코드 예시는 update-public-dns-namespace의 사용 방법을 보여 줍니다.

AWS CLI

퍼블릭 DNS 네임스페이스 업데이트

다음 update-public-dns-namespace 예시에서는 퍼블릭 DNS 네임스페이스의 설명을 업데이트합니다.

```
aws servicediscovery update-public-dns-namespace \
    --id ns-bk3aEXAMPLE \
    --updater-request-id example-public-request-id \
    --namespace Description="The updated namespace description."
```

출력:

```
{
    "OperationId": "gv4g5meo7ndmeh4fqskygvk23d2fijwa-k9302yzd"
}
```

작업이 성공했는지 확인하기 위해 를 실행할 수 있습니다get-operation.

자세한 내용은 AWS Cloud Map 개발자 안내서의 [AWS Cloud Map 네임스페이스](https://docs.aws.amazon.com/cloud-map/latest/dg/working-with-namespaces.html)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdatePublicDnsNamespace를 참조하세요.

update-service-attributes

다음 코드 예시는 update-service-attributes의 사용 방법을 보여 줍니다.

AWS CLI

속성을 추가하도록 서비스 업데이트

다음 update-service-attributes 예시에서는 지정된 서비스를 업데이트하여 Port 키와 80 값이 있는 서비스 속성을 추가합니다.

```
aws servicediscovery update-service-attributes \
    --service-id srv-e4anhexample0004 \
    --attributes Port=80
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Cloud Map 개발자 안내서의 AWS Cloud Map 서비스를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateServiceAttributes를 참조하세요.

`update-service`

다음 코드 예시는 `update-service`의 사용 방법을 보여 줍니다.

AWS CLI

서비스 업데이트

다음 `update-service` 예시에서는 서비스를 업데이트하여 `DnsConfig` 및 `HealthCheckConfig` 설정을 업데이트합니다.

```
aws servicediscovery update-service \
    --id srv-e4anhexample0004 \
    --service
 "DnsConfig={DnsRecords=[{"Type"="A","TTL"=60}]},HealthCheckConfig={"Type"="HTTP","ResourceP
```

출력:

```
{
    "OperationId": "gv4g5meo7ndmeh4fqskygvk23d2fijwa-k9302yzd"
}
```

작업이 성공했는지 확인하기 위해 를 실행할 수 있습니다`get-operation`.

서비스 업데이트에 대한 자세한 내용은 AWS Cloud Map 개발자 안내서의 AWS Cloud Map 서비스 업데이트를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateService를 참조하세요.

AWS CLI를 사용한 AWS Cloud9 예시

다음 코드 예시에서는 AWS Cloud9에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`create-environment-ec2`

다음 코드 예시는 `create-environment-ec2`의 사용 방법을 보여 줍니다.

AWS CLI

AWS Cloud9 EC2 개발 환경을 생성하는 방법

다음 `create-environment-ec2` 예제에서는 지정된 설정을 사용하여 AWS Cloud9 개발 환경을 생성하고, Amazon Elastic Compute Cloud(Amazon EC2) 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 세부 정보는 AWS CLI 명령 참조의 CreateEnvironmentEc2를 참조하세요.

`create-environment-membership`

다음 코드 예시는 `create-environment-membership`의 사용 방법을 보여 줍니다.

AWS CLI

AWS Cloud9 개발 환경에 환경 멤버를 추가하는 방법

이 예제에서는 지정된 환경 멤버를 지정된 AWS Cloud9 개발 환경에 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"
  }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateEnvironmentMembership을 참조하세요.

delete-environment-membership

다음 코드 예시는 delete-environment-membership의 사용 방법을 보여 줍니다.

AWS CLI

AWS Cloud9 개발 환경에서 환경 멤버를 삭제하는 방법

이 예제에서는 지정된 환경 멤버를 지정된 AWS Cloud9 개발 환경에서 삭제합니다.

명령:

```
aws cloud9 delete-environment-membership --environment-
id 8a34f51ce1e04a08882f1e811bd706EX --user-arn arn:aws:iam::123456789012:user/
AnotherDemoUser
```

출력:

```
None.
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteEnvironmentMembership을 참조하세요.

delete-environment

다음 코드 예시는 delete-environment의 사용 방법을 보여 줍니다.

AWS CLI

AWS Cloud9 개발 환경을 삭제하는 방법

이 예제에서는 지정된 AWS Cloud9 개발 환경을 삭제합니다. 환경에 Amazon EC2 인스턴스가 연결된 경우 해당 인스턴스도 종료됩니다.

명령:

```
aws cloud9 delete-environment --environment-id 8a34f51ce1e04a08882f1e811bd706EX
```

출력:

```
None.
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteEnvironment를 참조하세요.

describe-environment-memberships

다음 코드 예시는 describe-environment-memberships의 사용 방법을 보여 줍니다.

AWS CLI

AWS Cloud9 개발 환경의 환경 멤버에 대한 정보를 가져오는 방법

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 가져오는 방법

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 가져오는 방법

이 예제에서는 여러 AWS Cloud9 개발 환경에 대해 지정된 환경 멤버의 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"
    }
  ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeEnvironmentMemberships를 참조하세요.

`describe-environment-status`

다음 코드 예시는 `describe-environment-status`의 사용 방법을 보여 줍니다.

AWS CLI

AWS Cloud9 개발 환경에 대한 상태 정보를 가져오는 방법

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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeEnvironmentStatus를 참조하세요.

describe-environments

다음 코드 예시는 describe-environments의 사용 방법을 보여 줍니다.

AWS CLI

AWS Cloud9 개발 환경에 대한 정보를 가져오는 방법

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"
        }
      }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeEnvironments](https://docs.aws.amazon.com/goto/aws-cli/cloud9-2017-09-23/DescribeEnvironments)를 참조하세요.

`list-environments`

다음 코드 예시는 `list-environments`의 사용 방법을 보여 줍니다.

AWS CLI

사용 가능한 AWS Cloud9 개발 환경 식별자의 목록을 가져오는 방법

이 예제에서는 사용 가능한 AWS Cloud9 개발 환경 식별자의 목록을 가져옵니다.

명령:

```
aws cloud9 list-environments
```

출력:

```
{
  "environmentIds": [
    "685f892f431b45c2b28cb69eadcdb0EX",
    "1980b80e5f584920801c09086667f0EX"
  ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListEnvironments를 참조하세요.

update-environment-membership

다음 코드 예시는 update-environment-membership의 사용 방법을 보여 줍니다.

AWS CLI

AWS Cloud9 개발 환경에 대한 기존 환경 멤버의 설정을 변경하는 방법

이 예제에서는 지정된 AWS Cloud9 개발 환경에 대한 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"
  }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateEnvironmentMembership을 참조하세요.

update-environment

다음 코드 예시는 update-environment의 사용 방법을 보여 줍니다.

AWS CLI

기존 AWS Cloud9 개발 환경의 설정을 변경하는 방법

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.
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateEnvironment를 참조하세요.

AWS CLI를 사용한 AWS CloudFormation 예시

다음 코드 예시에서는 AWS CloudFormation에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`activate-type`

다음 코드 예제에서는 `activate-type`의 사용 방법을 보여줍니다.

AWS CLI

유형을 활성화하려면

다음 `activate-type` 예제에서는 퍼블릭 타사 익스텐션을 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 내용을 알아보려면 AWS CloudFormation 사용자 가이드의 AWS CloudFormation 레지스트리 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ActivateType을 참조하세요.

batch-describe-type-configurations

다음 코드 예제에서는 batch-describe-type-configurations의 사용 방법을 보여줍니다.

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 내용을 알아보려면 AWS CloudFormation 사용자 가이드의 AWS CloudFormation 레지스트리 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 BatchDescribeTypeConfigurations를 참조하세요.

cancel-update-stack

다음 코드 예제에서는 `cancel-update-stack`의 사용 방법을 보여줍니다.

AWS CLI

진행 중인 스택 업데이트를 취소하는 방법

다음 `cancel-update-stack` 명령은 `myteststack` 스택의 스택 업데이트를 취소합니다.

```
aws cloudformation cancel-update-stack --stack-name myteststack
```

- API 세부 정보는 AWS CLI 명령 참조의 CancelUpdateStack 섹션을 참조하세요.

continue-update-rollback

다음 코드 예제에서는 `continue-update-rollback`의 사용 방법을 보여줍니다.

AWS CLI

업데이트 롤백을 재시도하는 방법

다음 `continue-update-rollback` 예제에서는 이전에 실패한 스택 업데이트에서 롤백 작업을 재개합니다.

```
aws cloudformation continue-update-rollback \
```

```
    --stack-name my-stack
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 ContinueUpdateRollback 섹션을 참조하세요.

create-change-set

다음 코드 예제에서는 create-change-set의 사용 방법을 보여줍니다.

AWS CLI

변경 세트를 생성하려면

다음 create-change-set 예제에서는 CAPABILITY_IAM 기능을 사용하여 변경 세트를 생성합니다. template.yaml 파일은 IAM 리소스를 포함하는 스택을 정의하는 현재 폴더의 AWS CloudFormation 템플릿입니다.

```
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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateChangeSet를 참조하세요.

create-stack-instances

다음 코드 예제에서는 create-stack-instances의 사용 방법을 보여줍니다.

AWS CLI

스택 인스턴스를 생성하려면

다음 `create-stack-instances` 예제에서는 두 계정과 네 리전에서 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"
}
```

`create-stack-set` 명령을 사용하여 스택을 생성합니다.

- API 세부 정보는 AWS CLI 명령 참조의 CreateStackInstances를 참조하세요.

`create-stack-set`

다음 코드 예제에서는 `create-stack-set`의 사용 방법을 보여줍니다.

AWS CLI

스택 세트를 생성하려면

다음 `create-stack-set` 예제에서는 지정된 YAML 파일 템플릿을 사용하여 스택 세트를 생성합니다. `template.yaml`은 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 사용하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateStackSet를 참조하세요.

`create-stack`

다음 코드 예제에서는 `create-stack`의 사용 방법을 보여줍니다.

AWS CLI

CloudFormation 스택을 생성하는 방법

다음 `create-stacks` 명령에서는 `sampletemplate.json` 템플릿을 사용하여 이름이 `myteststack`인 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 내용을 알아보려면 AWS CloudFormation 사용 설명서의 스택을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateStack 섹션을 참조하세요.

`deactivate-type`

다음 코드 예제에서는 `deactivate-type`의 사용 방법을 보여줍니다.

AWS CLI

유형을 비활성화하려면

다음 `deactivate-type` 예제에서는 이 계정 및 리전에서 이전에 활성화된 퍼블릭 익스텐션을 비활성화합니다.

```
aws cloudformation deactivate-type \
```

```
    --region us-west-2 \
    --type MODULE \
    --type-name Example::Test::Type::MODULE
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용을 알아보려면 AWS CloudFormation 사용자 가이드의 AWS CloudFormation 레지스트리 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeactivateType을 참조하세요.

delete-change-set

다음 코드 예제에서는 delete-change-set의 사용 방법을 보여줍니다.

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 세트의 전체 ARN을 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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteChangeSet를 참조하세요.

delete-stack-instances

다음 코드 예제에서는 delete-stack-instances의 사용 방법을 보여줍니다.

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 사용하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteStackInstances를 참조하세요.

delete-stack-set

다음 코드 예제에서는 delete-stack-set의 사용 방법을 보여줍니다.

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 사용하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteStackSet를 참조하세요.

delete-stack

다음 코드 예제에서는 delete-stack의 사용 방법을 보여줍니다.

AWS CLI

스택을 삭제하는 방법

다음 delete-stack 예제에서는 지정된 스택을 삭제합니다.

```
aws cloudformation delete-stack \
    --stack-name my-stack
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteStack을 참조하세요.

deploy

다음 코드 예제에서는 deploy의 사용 방법을 보여줍니다.

AWS CLI

다음 명령은 이름이 template.json인 템플릿을 my-new-stack라는 스택에 배포합니다.

```
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 세부 정보는 AWS CLI 명령 참조의 Deploy를 참조하세요.

deregister-type

다음 코드 예제에서는 deregister-type의 사용 방법을 보여줍니다.

AWS CLI

유형 버전 등록을 취소하려면

다음 deregister-type 예제에서는 CloudFormation 레지스트리의 활성 사용에서 지정된 유형 버전을 제거하여 CloudFormation 작업에서 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 내용을 알아보려면 AWS CloudFormation 사용자 가이드의 CloudFormation 레지스트리 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeregisterType을 참조하세요.

describe-account-limits

다음 코드 예제에서는 describe-account-limits의 사용 방법을 보여줍니다.

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
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeAccountLimits 섹션을 참조하세요.

describe-change-set

다음 코드 예제에서는 describe-change-set의 사용 방법을 보여줍니다.

AWS CLI

변경 집합에 대한 정보를 얻으려면

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 변경 세트의 전체 ARN에 지정된 변경 세트의 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeChangeSet를 참조하세요.

describe-publisher

다음 코드 예제에서는 describe-publisher의 사용 방법을 보여줍니다.

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 내용을 알아보려면 AWS CloudFormation 사용자 가이드의 AWS CloudFormation 레지스트리 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribePublisher를 참조하세요.

describe-stack-drift-detection-status

다음 코드 예제에서는 describe-stack-drift-detection-status의 사용 방법을 보여줍니다.

AWS CLI

드리프트 감지 작업의 상태를 확인하려면

다음 describe-stack-drift-detection-status 예제에서는 드리프트 감지 작업의 상태를 표시합니다. ID를 얻으려면 detect-stack-drift 명령을 실행하세요.

```
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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeStackDriftDetectionStatus]를 참조하세요.

describe-stack-events

다음 코드 예제에서는 describe-stack-events의 사용 방법을 보여줍니다.

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=="
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeStackEvents 섹션을 참조하세요.

describe-stack-instance

다음 코드 예제에서는 describe-stack-instance의 사용 방법을 보여줍니다.

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 123456789012의 us-west-2 리전에 있습니다.

```
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=)."
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeStackInstance]를 참조하세요.

describe-stack-resource-drifts

다음 코드 예제에서는 describe-stack-resource-drifts의 사용 방법을 보여줍니다.

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 사용하세요.

```
aws cloudformation describe-stack-resource-drifts \
    --stack-name my-stack
```

출력은 대역 외에서 수정된 AWS Lambda 함수를 보여줍니다.

```
{
    "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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeStackResourceDrifts를 참조하세요.

describe-stack-resource

다음 코드 예제에서는 `describe-stack-resource`의 사용 방법을 보여줍니다.

AWS CLI

스택 리소스에 대한 정보를 가져오려면

다음 `describe-stack-resource` 예제에서는 지정된 스택의 이름이 `MyFunction`인 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"
            }
        }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeStackResource](#) 섹션을 참조하세요.

describe-stack-resources

다음 코드 예제에서는 describe-stack-resources의 사용 방법을 보여줍니다.

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"
            }
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeStackResources 섹션을 참조하세요.

describe-stack-set-operation

다음 코드 예제에서는 `describe-stack-set-operation`의 사용 방법을 보여줍니다.

AWS CLI

스택 세트 작업에 대한 정보를 가져오려면

다음 describe-stack-set-operation 예제에서는 지정된 스택 세트의 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeStackSetOperation을 참조하세요.

`describe-stack-set`

다음 코드 예제에서는 `describe-stack-set`의 사용 방법을 보여줍니다.

AWS CLI

스택 세트에 대한 정보를 가져오려면

다음 describe-stack-set 예제에서는 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeStackSet를 참조하세요.

describe-stacks

다음 코드 예제에서는 describe-stacks의 사용 방법을 보여줍니다.

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/
mytestack/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 내용을 알아보려면 AWS CloudFormation 사용 설명서의 스택을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeStacks](#) 섹션을 참조하세요.

`describe-type-registration`

다음 코드 예제에서는 `describe-type-registration`의 사용 방법을 보여줍니다.

AWS CLI

유형 등록 정보를 표시하려면

다음 `describe-type-registration` 예제에서는 유형의 현재 상태, 유형, 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 내용을 알아보려면 AWS CloudFormation 사용자 가이드의 CloudFormation 레지스트리 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeTypeRegistration을 참조하세요.

describe-type

다음 코드 예제에서는 describe-type의 사용 방법을 보여줍니다.

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 내용을 알아보려면 AWS CloudFormation 사용자 가이드의 CloudFormation 레지스트리 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeType을 참조하세요.

`detect-stack-drift`

다음 코드 예제에서는 `detect-stack-drift`의 사용 방법을 보여줍니다.

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.

- API 세부 정보는 AWS CLI 명령 참조의 DetectStackDrift를 참조하세요.

`detect-stack-resource-drift`

다음 코드 예제에서는 `detect-stack-resource-drift`의 사용 방법을 보여줍니다.

AWS CLI

리소스의 드리프트를 감지하려면

다음 `detect-stack-resource-drift` 예제에서는 드리프트에 대해 `MyStack`라는 스택에 `MyFunction`라는 리소스를 확인합니다.

```
aws cloudformation detect-stack-resource-drift \
    --stack-name MyStack \
    --logical-resource-id MyFunction
```

출력은 대역 외에서 수정된 AWS Lambda 함수를 보여줍니다.

```
{
    "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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DetectStackResourceDrift를 참조하세요.

detect-stack-set-drift

다음 코드 예제에서는 `detect-stack-set-drift`의 사용 방법을 보여줍니다.

AWS CLI

스택 세트 및 모든 관련 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 내용을 알아보려면 AWS CloudFormation 사용자 가이드의 스택 세트에서 관리되지 않는 구성 변경 사항 감지를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DetectStackSetDrift를 참조하세요.

estimate-template-cost

다음 코드 예제에서는 `estimate-template-cost`의 사용 방법을 보여줍니다.

AWS CLI

템플릿 비용을 추정하려면

다음 `estimate-template-cost` 예제에서는 현재 폴더에서 이름이 `template.yaml`인 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 EstimateTemplateCost 섹션을 참조하세요.

execute-change-set

다음 코드 예제에서는 execute-change-set의 사용 방법을 보여줍니다.

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 변경 세트의 전체 ARN에서 지정된 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
```

- API 세부 정보는 AWS CLI 명령 참조의 ExecuteChangeSet를 참조하세요.

get-stack-policy

다음 코드 예제에서는 get-stack-policy의 사용 방법을 보여줍니다.

AWS CLI

스택 정책을 보려면

다음 `get-stack-policy` 예제에서는 지정된 스택에 대한 스택 정책을 표시합니다. 스택에 정책을 연결하려면 `set-stack-policy` 명령을 사용하세요.

```
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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [GetStackPolicy](#)를 참조하세요.

get-template-summary

다음 코드 예제에서는 `get-template-summary`의 사용 방법을 보여줍니다.

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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetTemplateSummary를 참조하세요.

get-template

다음 코드 예제에서는 `get-template`의 사용 방법을 보여줍니다.

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"
                }
            }
        }
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetTemplate을 참조하세요.

list-change-sets

다음 코드 예제에서는 `list-change-sets`의 사용 방법을 보여줍니다.

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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListChangeSets를 참조하세요.

list-exports

다음 코드 예제에서는 `list-exports`의 사용 방법을 보여줍니다.

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"
        }
    ]
```

```
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListExports를 참조하세요.

list-imports

다음 코드 예제에서는 list-imports의 사용 방법을 보여줍니다.

AWS CLI

가져오기를 나열하려면

다음 list-imports 예제에서는 지정된 내보내기를 가져오는 스택을 나열합니다. 사용 가능한 내보내기 목록을 가져오려면 list-exports 명령을 사용하세요.

```
aws cloudformation list-imports \
    --export-name private-vpc-vpcid
```

출력:

```
{
    "Imports": [
        "my-database-stack"
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListImports를 참조하세요.

list-stack-instances

다음 코드 예제에서는 list-stack-instances의 사용 방법을 보여줍니다.

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)."
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListStackInstances를 참조하세요.

`list-stack-resources`

다음 코드 예제에서는 `list-stack-resources`의 사용 방법을 보여줍니다.

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"
            }
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListStackResources 섹션을 참조하세요.

`list-stack-set-operation-results`

다음 코드 예제에서는 `list-stack-set-operation-results`의 사용 방법을 보여줍니다.

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 `AccountGateResult`의 `SKIPPED` 상태는 성공적인 작업에 대해 예상됩니다.

- API 세부 정보는 AWS CLI 명령 참조의 ListStackSetOperationResults를 참조하세요.

`list-stack-set-operations`

다음 코드 예제에서는 `list-stack-set-operations`의 사용 방법을 보여줍니다.

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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListStackSetOperations를 참조하세요.

`list-stack-sets`

다음 코드 예제에서는 `list-stack-sets`의 사용 방법을 보여줍니다.

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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListStackSets를 참조하세요.

`list-stacks`

다음 코드 예제에서는 `list-stacks`의 사용 방법을 보여줍니다.

AWS CLI

AWS CloudFormation 스택을 나열하려면

다음 `list-stacks` 명령에서는 상태가 `CREATE_COMPLETE`인 모든 스택에 대한 요약 내용을 보여줍니다.

```
aws cloudformation list-stacks --stack-status-filter CREATE_COMPLETE
```

출력:

```
[
    {
        "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
mytestsstack/466df9e0-0dff-08e3-8e2f-5088487c4896",
```

```
        "TemplateDescription": "AWS CloudFormation Sample Template S3_Bucket: Sample
 template showing how to create a publicly accessible S3 bucket. **WARNING** This
 template creates an S3 bucket. You will be billed for the AWS resources used if you
 create a stack from this template.",
        "StackStatusReason": null,
        "CreationTime": "2013-08-26T03:27:10.190Z",
        "StackName": "myteststack",
        "StackStatus": "CREATE_COMPLETE"
    }
]
```

- API 세부 정보는 AWS CLI 명령 참조의 ListStacks 섹션을 참조하세요.

`list-type-registrations`

다음 코드 예제에서는 `list-type-registrations`의 사용 방법을 보여줍니다.

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 내용을 알아보려면 AWS CloudFormation 사용자 가이드의 CloudFormation 레지스트리 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTypeRegistrations를 참조하세요.

list-type-versions

다음 코드 예제에서는 list-type-versions의 사용 방법을 보여줍니다.

AWS CLI

익스텐션 버전을 나열하려면

다음 list-type-versions 예제에서는 익스텐션 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 내용을 알아보려면 AWS CloudFormation 사용자 가이드의 AWS CloudFormation 레지스트리 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTypeVersions를 참조하세요.

list-types

다음 코드 예제에서는 list-types의 사용 방법을 보여줍니다.

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 내용을 알아보려면 AWS CloudFormation 사용자 가이드의 CloudFormation 레지스트리 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTypes를 참조하세요.

package

다음 코드 예제에서는 `package`의 사용 방법을 보여줍니다.

AWS CLI

다음 명령은 로컬 아티팩트를 S3 버킷 `bucket-name`에 업로드하여 `template.json`라는 템플릿을 내보내고 내보낸 템플릿을 `packaged-template.json`에 씁니다.

```
aws cloudformation package --template-file /path_to_template/template.json --s3-
bucket bucket-name --output-template-file packaged-template.json --use-json
```

- API 세부 정보는 AWS CLI 명령 참조의 Package를 참조하세요.

publish-type

다음 코드 예제에서는 publish-type의 사용 방법을 보여줍니다.

AWS CLI

익스텐션을 게시하려면

다음 publish-type 예제에서는 지정된 확장을 이 리전의 퍼블릭 익스텐션으로 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 내용을 알아보려면 AWS CloudFormation 사용자 가이드의 AWS CloudFormation 레지스트리 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PublishType을 참조하세요.

register-publisher

다음 코드 예제에서는 register-publisher의 사용 방법을 보여줍니다.

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 내용을 알아보려면 AWS CloudFormation 사용자 가이드의 AWS CloudFormation 레지스트리 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RegisterPublisher를 참조하세요.

register-type

다음 코드 예제에서는 register-type의 사용 방법을 보여줍니다.

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 유형 개발을 위한 CloudFormation 명령줄 인터페이스 사용자 가이드에서 리소스 공급자 등록을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RegisterType을 참조하세요.

`set-stack-policy`

다음 코드 예제에서는 `set-stack-policy`의 사용 방법을 보여줍니다.

AWS CLI

스택 정책을 적용하려면

다음 `set-stack-policy` 예제에서는 지정된 스택의 지정된 리소스에 대한 업데이트를 비활성화합니다. `stack-policy.json`은 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"
    }
  ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 SetStackPolicy를 참조하세요.

`set-type-configuration`

다음 코드 예제에서는 `set-type-configuration`의 사용 방법을 보여줍니다.

AWS CLI

데이터를 구성하려면

다음 set-type-configuration 예제에서는 지정된 계정 및 리전에서 등록된 CloudFormation 익스텐션의 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 내용을 알아보려면 AWS CloudFormation 사용자 가이드의 AWS CloudFormation 레지스트리 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 SetTypeConfiguration을 참조하세요.

set-type-default-version

다음 코드 예제에서는 set-type-default-version의 사용 방법을 보여줍니다.

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 내용을 알아보려면 AWS CloudFormation 사용자 가이드의 CloudFormation 레지스트리 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [SetTypeDefaultVersion](https://docs.aws.amazon.com/goto/aws-cli/cloudformation-2010-05-15/SetTypeDefaultVersion)을 참조하세요.

signal-resource

다음 코드 예제에서는 `signal-resource`의 사용 방법을 보여줍니다.

AWS CLI

리소스에 신호를 보내려면

다음 `signal-resource` 예제는 `my-stack`라는 스택에 `MyWaitCondition`라는 대기 조건을 충족하도록 `success` 신호를 보냅니다.

```
aws cloudformation signal-resource \
    --stack-name my-stack \
    --logical-resource-id MyWaitCondition \
    --unique-id 1234 \
    --status SUCCESS
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 [SignalResource](https://docs.aws.amazon.com/goto/aws-cli/cloudformation-2010-05-15/SignalResource)를 참조하세요.

stop-stack-set-operation

다음 코드 예제에서는 `stop-stack-set-operation`의 사용 방법을 보여줍니다.

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.

- API 세부 정보는 AWS CLI 명령 참조의 [StopStackSetOperation](https://docs.aws.amazon.com/goto/aws-cli/cloudformation-2010-05-15/StopStackSetOperation)을 참조하세요.

test-type

다음 코드 예제에서는 test-type의 사용 방법을 보여줍니다.

AWS CLI

익스텐션을 테스트하려면

다음 test-type 예제에서는 등록된 익스텐션을 테스트하여 CloudFormation 레지스트리에 게시되는 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 내용을 알아보려면 AWS CloudFormation 사용자 가이드의 AWS CloudFormation 레지스트리 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TestType을 참조하세요.

update-stack-instances

다음 코드 예제에서는 update-stack-instances의 사용 방법을 보여줍니다.

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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateStackInstances를 참조하세요.

update-stack-set

다음 코드 예제에서는 update-stack-set의 사용 방법을 보여줍니다.

AWS CLI

스택 세트를 업데이트하려면

다음 update-stack-set 예제에서는 지정된 스택 세트의 스택 인스턴스에 키 이름 Owner와 IT 값이 포함된 태그를 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateStackSet를 참조하세요.

update-stack

다음 코드 예제에서는 update-stack의 사용 방법을 보여줍니다.

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 내용을 알아보려면 AWS CloudFormation 사용 설명서의 AWS CloudFormation 스택 업데이트를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateStack을 참조하세요.

update-termination-protection

다음 코드 예제에서는 update-termination-protection의 사용 방법을 보여줍니다.

AWS CLI

종료 방지 기능을 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateTerminationProtection을 참조하세요.

validate-template

다음 코드 예제에서는 validate-template의 사용 방법을 보여줍니다.

AWS CLI

AWS CloudFormation 템플릿의 유효성을 확인하려면

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 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ValidateTemplate을 참조하세요.

AWS CLI를 사용한 CloudFront 예시

다음 코드 예제는 CloudFront와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

associate-distribution-tenant-web-acl

다음 코드 예시는 associate-distribution-tenant-web-acl의 사용 방법을 보여 줍니다.

AWS CLI

웹 ACL을 CloudFront 배포 테넌트와 연결하려면

다음 associate-distribution-tenant-web-acl 예제에서는 웹 ACL을 ETag가 E13V1IB3VIYABC인 CloudFront 배포와 연결합니다.

```
aws cloudfront associate-distribution-tenant-web-acl \
    --id dt_2wjDZi3hD1ivOXf6rpZJO1AB \
    --if-match E13V1IB3VIYABC \
    --web-acl-arn arn:aws:wafv2:us-east-1:123456789012:global/webacl/web-global-
example/626900da-5f64-418b-ba9b-743f37123ABC
```

출력:

```
{
    "ETag": "E1VC38T7YXBABC",
    "Id": "dt_2wjDZi3hD1ivOXf6rpZJO1AB",
    "WebACLArn": "arn:aws:wafv2:us-east-1:123456789012:global/webacl/web-global-
example/626900da-5f64-418b-ba9b-743f37123ABC"
}
```

자세한 내용은 Amazon CloudFront 개발자 안내서의 Use AWS WAF protections를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AssociateDistributionTenantWebAcl 섹션을 참조하세요.

associate-distribution-web-acl

다음 코드 예시는 associate-distribution-web-acl의 사용 방법을 보여 줍니다.

AWS CLI

웹 ACL을 CloudFront 배포와 연결하려면

다음 associate-distribution-web-acl 예제에서는 웹 ACL을 CloudFront 배포와 연결합니다.

```
aws cloudfront associate-distribution-web-acl \
    --id E1XNX8R2GOAABC \
    --if-match E2YWS1C2J3OABC \
    --web-acl-arn arn:aws:wafv2:us-east-1:123456789012:global/webacl/web-global-
example/626900da-5f64-418b-ba9b-743f3746cABC
```

출력:

```
{
    "ETag": "E3QE7ED60U0ABC",
    "Id": "E1XNX8R2GOAABC",
    "WebACLArn": "arn:aws:wafv2:us-east-1:123456789012:global/webacl/web-global-
example/626900da-5f64-418b-ba9b-743f3746cABC"
}
```

자세한 내용은 Amazon CloudFront 개발자 안내서의 Use AWS WAF protections를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AssociateDistributionWebAcl 섹션을 참조하세요.

create-cloud-front-origin-access-identity

다음 코드 예시는 create-cloud-front-origin-access-identity의 사용 방법을 보여 줍니다.

AWS CLI

CloudFront 원본 액세스 ID 생성

다음 예시에서는 명령줄 인수로 OAI 구성을 제공하여 CloudFront 오리진 액세스 ID(OAI)를 생성합니다.

```
aws cloudfront create-cloud-front-origin-access-identity \
    --cloud-front-origin-access-identity-config \
        CallerReference="cli-example",Comment="Example OAI"
```

다음 예시와 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
```

`OAI-config.json` 파일은 다음을 포함한 현재 디렉터리의 JSON 문서입니다.

```
{
    "CallerReference": "cli-example",
    "Comment": "Example OAI"
}
```

명령줄 인수로 OAI 구성을 제공하든 JSON 파일로 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"
        }
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateCloudFrontOriginAccessIdentity 섹션을 참조하세요.

`create-connection-group`

다음 코드 예시는 `create-connection-group`의 사용 방법을 보여 줍니다.

AWS CLI

CloudFront에서 연결 그룹을 만들려면

다음 `create-connection-group` 예제에서는 활성화된 연결 그룹을 만들고, 애니캐스트 고정 IP 목록을 지정하고, IPv6를 비활성화합니다.

```
aws cloudfront create-connection-group \
    --name cg-with-anycast-ip-list \
    --no-ipv6-enabled \
    --enabled \
    --anycast-ip-list-id aip_CCkW6gKrDiBD4n78123ABC \
    --tags "Items=[{Key=abc,Value=123}]"
```

출력:

```
{
    "ETag": "E23ZP02F085ABC",
    "ConnectionGroup": {
        "Id": "cg_2yb6uj74B4PCbfhT31WFdiSABC",
        "Name": "cg-with-anycast-ip-list",
        "Arn": "arn:aws:cloudfront::123456789012:connection-group/
cg_2yb6uj74B4PCbfhT31WFdiSABC",
        "CreatedTime": "2025-06-16T16:25:50.061000+00:00",
        "LastModifiedTime": "2025-06-16T16:25:50.061000+00:00",
        "Tags": {
            "Items": [
                {
                    "Key": "abc",
                    "Value": "123"
                }
            ]
        },
        "Ipv6Enabled": false,
        "RoutingEndpoint": "dj6xusxq65abc.cloudfront.net",
        "AnycastIpListId": "aip_CCkW6gKrDiBD4n78123ABC",
        "Status": "InProgress",
        "Enabled": true,
        "IsDefault": false
```

```
    }
}
```

자세한 내용은 Amazon CloudFront 개발자 안내서의 Create custom connection group (optional)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateConnectionGroup 섹션을 참조하세요.

create-distribution-tenant

다음 코드 예시는 `create-distribution-tenant`의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 사용자 지정 인증서를 사용하는 CloudFront 배포 테넌트를 생성하는 방법

다음 `create-distribution-tenant` 예제에서는 WAF 비활성화, 지리적 제한 추가, 다른 TLS 인증서 사용 등의 사용자 지정을 포함하는 CloudFront 배포 테넌트를 만듭니다.

```
aws cloudfront create-distribution-tenant \
    --cli-input-json file://tenant.json
```

`tenant.json`의 콘텐츠:

```
{
    "DistributionId": "E1XNX8R2GOAABC",
    "Domains": [
        {
            "Domain": "example.com"
        }
    ],
    "Parameters": [
        {
            "Name": "testParam",
            "Value": "defaultValue"
        }
    ],
    "ConnectionGroupId": "cg_2whCJoXMYCjHcxaLGrkllvyABC",
    "Enabled": false,
    "Tags": {
        "Items": [
            {
```

```
                "Key": "tag",
                "Value": "tagValue"
            }
        ]
    },
    "Name": "new-tenant-customizations",
    "Customizations": {
        "GeoRestrictions": {
            "Locations": ["DE"],
            "RestrictionType": "whitelist"
        },
        "WebAcl": {
            "Action": "disable"
        },
        "Certificate": {
            "Arn": "arn:aws:acm:us-east-1:123456789012:certificate/ec53f564-
ea5a-4e4a-a0a2-e3c989449abc"
        }
    }
}
```

출력:

```
{
    "ETag": "E23ZP02F085ABC",
    "DistributionTenant": {
        "Id": "dt_2yN5tYwVbPKr7m2IB69M1yp1AB",
        "DistributionId": "E1XNX8R2GOAABC",
        "Name": "new-tenant-customizations",
        "Arn": "arn:aws:cloudfront::123456789012:distribution-tenant/
dt_2yN5tYwVbPKr7m2IB69M1yp1AB",
        "Domains": [
            {
                "Domain": "example.com",
                "Status": "active"
            }
        ],
        "Tags": {
            "Items": [
                {
                    "Key": "tag",
                    "Value": "tagValue"
                }
```

```
            ]
        },
        "Customizations": {
            "WebAcl": {
                "Action": "disable"
            },
            "Certificate": {
                "Arn": "arn:aws:acm:us-east-1:123456789012:certificate/ec53f564-
ea5a-4e4a-a0a2-e3c989449abc"
            },
            "GeoRestrictions": {
                "RestrictionType": "whitelist",
                "Locations": [
                    "DE"
                ]
            }
        },
        "Parameters": [
            {
                "Name": "testParam",
                "Value": "defaultValue"
            }
        ],
        "ConnectionGroupId": "cg_2whCJoXMYCjHcxaLGrkllvyABC",
        "CreatedTime": "2025-06-11T17:20:06.432000+00:00",
        "LastModifiedTime": "2025-06-11T17:20:06.432000+00:00",
        "Enabled": false,
        "Status": "InProgress"
    }
}
```

예제 2: 상속된 인증서로 배포 테넌트를 생성하는 방법

다음 `create-distribution-tenant` 예제에서는 배포 테넌트를 생성하고 다중 테넌트 배포에서 상속된 TLS 인증서를 지정합니다.

```
aws cloudfront create-distribution-tenant \
    --cli-input-json file://tenant.json
```

`tenant.json`의 콘텐츠:

```
{
    "DistributionId": "E1HVIAU7U12ABC",
```

```
    "Domains": [
        {
            "Domain": "example.com"
        }
    ],
    "Parameters": [
        {
            "Name": "tenantName",
            "Value": "first-tenant"
        }
    ],
    "Enabled": true,
    "Name": "new-tenant-no-cert"
}
```

출력:

```
{
    "ETag": "E23ZP02F0ABC",
    "DistributionTenant": {
        "Id": "dt_2zhRB0vBe0B72LZCVy1mgzI1AB",
        "DistributionId": "E1HVIAU7U12ABC",
        "Name": "new-tenant-no-cert",
        "Arn": "arn:aws:cloudfront::123456789012:distribution-tenant/
dt_2zhRB0vBe0B72LZCVy1mgzI1AB",
        "Domains": [
            {
                "Domain": "example.com",
                "Status": "active"
            }
        ],
        "Parameters": [
            {
                "Name": "tenantName",
                "Value": "first-tenant"
            }
        ],
        "ConnectionGroupId": "cg_2yQEwpipGFN0hhA0ZemPabOABC",
        "CreatedTime": "2025-07-10T20:59:38.414000+00:00",
        "LastModifiedTime": "2025-07-10T20:59:38.414000+00:00",
        "Enabled": true,
        "Status": "InProgress"
    }
```

```
}
```

예제 3: CloudFront 호스팅 검증 토큰을 사용하여 CloudFront 배포 테넌트를 생성하는 방법

다음 `create-distribution-tenant` 예제에서는 배포 테넌트를 생성하고 도메인 이름에 CloudFront 호스팅 검증 토큰을 사용합니다.

```
aws cloudfront create-distribution-tenant \
    --cli-input-json file://tenant.json
```

`tenant.json`의 콘텐츠:

```
{
    "DistributionId": "E2GJ5J9QN12ABC",
    "Domains": [
        {
            "Domain": "example.com"
        }
    ],
    "Parameters": [
        {
            "Name": "tenantName",
            "Value": "first-tenant"
        }
    ],
    "ConnectionGroupId": "cg_2yQEwpipGFN0hhA0ZemPabOABC",
    "Enabled": true,
    "Name": "new-tenant-cf-hosted",
    "ManagedCertificateRequest": {
        "ValidationTokenHost": "cloudfront"
    }
}
```

중요: 이 명령을 성공적으로 실행하려면 새 도메인(example.com)이 배포 테넌트와 연결된 연결 그룹의 라우팅 엔드포인트를 가리키는 CNAME DNS 레코드를 구성해야 합니다. CloudFront에서 이 요청을 성공적으로 완료하려면 먼저 이 CNAME 레코드도 전파해야 합니다.

출력:

```
{
    "ETag": "E23ZP02F0ABC",
```

```
    "DistributionTenant": {
        "Id": "dt_2zhStKrA524GvvTWJX92Ozl1AB",
        "DistributionId": "E2GJ5J9QN12ABC",
        "Name": "new-tenant-cf-hosted",
        "Arn": "arn:aws:cloudfront::123456789012:distribution-tenant/
dt_2zhStKrA524GvvTWJX92Ozl1AB",
        "Domains": [
            {
                "Domain": "example.com",
                "Status": "inactive"
            }
        ],
        "Parameters": [
            {
                "Name": "tenantName",
                "Value": "first-tenant"
            }
        ],
        "ConnectionGroupId": "cg_2zhSaGatwwXjTjE42nneZzqABC",
        "CreatedTime": "2025-07-10T21:13:46.416000+00:00",
        "LastModifiedTime": "2025-07-10T21:13:46.416000+00:00",
        "Enabled": true,
        "Status": "InProgress"
    }
}
```

예제 4: 자체 호스팅 검증 토큰을 사용하여 CloudFront 배포 테넌트를 생성하는 방법

다음 `create-distribution-tenant` 예제에서는 CloudFront 배포 테넌트를 생성하고 자체 호스팅 검증 토큰을 사용합니다.

```
aws cloudfront create-distribution-tenant \
    --cli-input-json file://tenant.json
```

`tenant.json`의 콘텐츠:

```
{
    "DistributionId": "E2GJ5J9QN12ABC",
    "Domains": [
        {
            "Domain": "example.com"
        }
```

```
    ],
    "Parameters": [
        {
            "Name": "tenantName",
            "Value": "first-tenant"
        }
    ],
    "Enabled": true,
    "Name": "new-tenant-self-hosted",
    "ManagedCertificateRequest": {
        "ValidationTokenHost": "self-hosted"
    }
}
```

출력:

```
{
    "ETag": "E23ZP02F0ABC",
    "DistributionTenant": {
        "Id": "dt_2zhTFBV93OfFJJ3YMdNM5BC1AB",
        "DistributionId": "E2GJ5J9QN12ABC",
        "Name": "new-tenant-self-hosted",
        "Arn": "arn:aws:cloudfront::123456789012:distribution-tenant/
dt_2zhTFBV93OfFJJ3YMdNM5BC1AB",
        "Domains": [
            {
                "Domain": "example.com",
                "Status": "inactive"
            }
        ],
        "Parameters": [
            {
                "Name": "tenantName",
                "Value": "first-tenant"
            }
        ],
        "ConnectionGroupId": "cg_2yQEwpipGFN0hhA0ZemPabOABC",
        "CreatedTime": "2025-07-10T21:16:39.828000+00:00",
        "LastModifiedTime": "2025-07-10T21:16:39.828000+00:00",
        "Enabled": true,
        "Status": "InProgress"
    }
}
```

중요: 이 명령을 실행하면 검증 없이 배포 테넌트가 생성됩니다. 트래픽 수신을 시작할 준비가 되었을 때 관리형 인증서 요청을 검증하고 DNS를 구성하려면 Amazon CloudFront 개발자 안내서의 도메인 설정 완료를 참조하세요.

배포 테넌트 생성에 대한 자세한 내용은 Amazon CloudFront 개발자 안내서의 배포 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateDistributionTenant 섹션을 참조하세요.

`create-distribution-with-tags`

다음 코드 예시는 `create-distribution-with-tags`의 사용 방법을 보여 줍니다.

AWS CLI

태그를 사용하여 CloudFront 배포 생성

다음 `create-distribution-with-tags` 예제에서는 `dist-config-with-tags.json`이라는 JSON 파일에서 배포 구성과 태그를 제공하여 두 개의 태그가 있는 배포를 생성합니다.

```
aws cloudfront create-distribution-with-tags \
    --distribution-config-with-tags file://dist-config-with-tags.json
```

`dist-config-with-tags.json` 파일은 현재 폴더의 JSON 문서입니다. 두 개의 태그가 포함된 파일 상단의 `Tags` 객체를 기록해 둡니다.

`Name = ExampleDistributionProject = ExampleProject`

`dist-config-with-tags.json`의 콘텐츠:

```
{
    "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": "amzn-s3-demo-bucket.s3.amazonaws.com-cli-example",
                    "DomainName": "amzn-s3-demo-bucket.s3.amazonaws.com",
                    "OriginPath": "",
                    "CustomHeaders": {
                        "Quantity": 0
                    },
                    "S3OriginConfig": {
                        "OriginAccessIdentity": ""
                    }
                }
            ]
        },
        "OriginGroups": {
            "Quantity": 0
        },
        "DefaultCacheBehavior": {
            "TargetOriginId": "amzn-s3-demo-bucket.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": "amzn-s3-demo-bucket.s3.amazonaws.com-cli-example",
                        "DomainName": "amzn-s3-demo-bucket.s3.amazonaws.com",
                        "OriginPath": "",
                        "CustomHeaders": {
```

```
                        "Quantity": 0
                    },
                    "S3OriginConfig": {
                        "OriginAccessIdentity": ""
                    }
                }
            ]
        },
        "OriginGroups": {
            "Quantity": 0
        },
        "DefaultCacheBehavior": {
            "TargetOriginId": "amzn-s3-demo-bucket.s3.amazonaws.com-cli-
example",
            "ForwardedValues": {
                "QueryString": false,
                "Cookies": {
                    "Forward": "none"
                },
                "Headers": {
                    "Quantity": 0
                },
                "QueryStringCacheKeys": {
                    "Quantity": 0
                }
            },
            "TrustedSigners": {
                "Enabled": false,
                "Quantity": 0
            },
            "ViewerProtocolPolicy": "allow-all",
            "MinTTL": 0,
            "AllowedMethods": {
                "Quantity": 2,
                "Items": [
                    "HEAD",
                    "GET"
                ],
                "CachedMethods": {
                    "Quantity": 2,
                    "Items": [
                        "HEAD",
                        "GET"
                    ]
```

```
                    }
                },
                "SmoothStreaming": false,
                "DefaultTTL": 86400,
                "MaxTTL": 31536000,
                "Compress": false,
                "LambdaFunctionAssociations": {
                    "Quantity": 0
                },
                "FieldLevelEncryptionId": ""
            },
            "CacheBehaviors": {
                "Quantity": 0
            },
            "CustomErrorResponses": {
                "Quantity": 0
            },
            "Comment": "",
            "Logging": {
                "Enabled": false,
                "IncludeCookies": false,
                "Bucket": "",
                "Prefix": ""
            },
            "PriceClass": "PriceClass_All",
            "Enabled": true,
            "ViewerCertificate": {
                "CloudFrontDefaultCertificate": true,
                "MinimumProtocolVersion": "TLSv1",
                "CertificateSource": "cloudfront"
            },
            "Restrictions": {
                "GeoRestriction": {
                    "RestrictionType": "none",
                    "Quantity": 0
                }
            },
            "WebACLId": "",
            "HttpVersion": "http2",
            "IsIPV6Enabled": true
        }
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateDistributionWithTags 섹션을 참조하세요.

create-distribution

다음 코드 예시는 create-distribution의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: CloudFront 배포를 생성하는 방법

다음 create-distribution 예제에서는 명령줄 인수를 사용하여 이름이 amzn-s3-demo-bucket인 S3 버킷에 대한 배포를 생성하고 index.html을 기본 루트 객체로 지정합니다.

```
aws cloudfront create-distribution \
    --origin-domain-name amzn-s3-demo-bucket.s3.amazonaws.com \
    --default-root-object index.html
```

출력:

```
{
    "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": "amzn-s3-demo-bucket.s3.amazonaws.com-cli-example",
                        "DomainName": "amzn-s3-demo-bucket.s3.amazonaws.com",
                        "OriginPath": "",
                        "CustomHeaders": {
                            "Quantity": 0
                        },
                        "S3OriginConfig": {
                            "OriginAccessIdentity": ""
                        }
                    }
                ]
            },
            "OriginGroups": {
                "Quantity": 0
            },
            "DefaultCacheBehavior": {
                "TargetOriginId": "amzn-s3-demo-bucket.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
        }
    }
}
```

예제 2: JSON 파일을 사용하여 CloudFront 배포를 만드는 방법

다음 `create-distribution` 예제에서는 JSON 파일을 사용하여 이름이 `amzn-s3-demo-bucket`인 S3 버킷에 대한 배포를 생성하고 `index.html`을 기본 루트 객체로 지정합니다.

```
aws cloudfront create-distribution \
    --distribution-config file://dist-config.json
```

`dist-config.json`의 콘텐츠:

```
{
    "CallerReference": "cli-example",
    "Aliases": {
        "Quantity": 0
    },
    "DefaultRootObject": "index.html",
    "Origins": {
        "Quantity": 1,
        "Items": [
            {
                "Id": "amzn-s3-demo-bucket.s3.amazonaws.com-cli-example",
                "DomainName": "amzn-s3-demo-bucket.s3.amazonaws.com",
                "OriginPath": "",
                "CustomHeaders": {
                    "Quantity": 0
                },
                "S3OriginConfig": {
                    "OriginAccessIdentity": ""
                }
            }
        ]
    },
    "OriginGroups": {
        "Quantity": 0
    },
    "DefaultCacheBehavior": {
        "TargetOriginId": "amzn-s3-demo-bucket.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
}
```

샘플 출력은 예제 1을 참조하시기 바랍니다.

예제 3: 인증서를 사용하여 CloudFront 다중 테넌트 배포를 생성하는 방법

다음 `create-distribution` 예제에서는 다중 테넌트를 지원하는 CloudFront 배포를 생성하고는 TLS 인증서를 지정합니다.

```
aws cloudfront create-distribution \
    --distribution-config file://dist-config.json
```

`dist-config.json`의 콘텐츠:

```
{
    "CallerReference": "cli-example-with-cert",
    "Comment": "CLI example distribution",
    "DefaultRootObject": "index.html",
```

```
    "Origins": {
        "Quantity": 1,
        "Items": [
            {
                "Id": "amzn-s3-demo-bucket.s3.us-east-1.amazonaws.com",
                "DomainName": "amzn-s3-demo-bucket.s3.us-east-1.amazonaws.com",
                "OriginPath": "/{{tenantName}}",
                "CustomHeaders": {
                    "Quantity": 0
                },
                "S3OriginConfig": {
                    "OriginAccessIdentity": ""
                }
            }
        ]
    },
    "DefaultCacheBehavior": {
        "TargetOriginId": "amzn-s3-demo-bucket.s3.us-east-1.amazonaws.com",
        "CachePolicyId": "658327ea-f89d-4fab-a63d-7e88639e5ABC",
        "ViewerProtocolPolicy": "allow-all",
        "AllowedMethods": {
            "Quantity": 2,
            "Items": ["HEAD", "GET"],
            "CachedMethods": {
                "Quantity": 2,
                "Items": ["HEAD", "GET"]
            }
        }
    },
    "Enabled": true,
    "ViewerCertificate": {
        "ACMCertificateArn": "arn:aws:acm:us-
east-1:123456789012:certificate/191306a1-db01-49ca-90ef-fc414ee5dabc",
        "SSLSupportMethod": "sni-only"
    },
    "HttpVersion": "http2",
    "ConnectionMode": "tenant-only",
    "TenantConfig": {
        "ParameterDefinitions": [
            {
                "Name": "tenantName",
                "Definition": {
                    "StringSchema": {
                        "Comment": "tenantName parameter",
```

```
                        "DefaultValue": "root",
                        "Required": false
                    }
                }
            }
        ]
    }
}
```

출력:

```
{
    "Location": "https://cloudfront.amazonaws.com/2020-05-31/distribution/
E1HVIAU7UABC",
    "ETag": "E20LT7R1BABC",
    "Distribution": {
        "Id": "E1HVIAU7U12ABC",
        "ARN": "arn:aws:cloudfront::123456789012:distribution/E1HVIAU7U12ABC",
        "Status": "InProgress",
        "LastModifiedTime": "2025-07-10T20:33:31.117000+00:00",
        "InProgressInvalidationBatches": 0,
        "DomainName": "example.com",
        "ActiveTrustedSigners": {
            "Enabled": false,
            "Quantity": 0
        },
        "ActiveTrustedKeyGroups": {
            "Enabled": false,
            "Quantity": 0
        },
        "DistributionConfig": {
            "CallerReference": "cli-example-with-cert",
            "DefaultRootObject": "index.html",
            "Origins": {
                "Quantity": 1,
                "Items": [
                    {
                        "Id": "amzn-s3-demo-bucket.s3.us-east-1.amazonaws.com",
                        "DomainName": "amzn-s3-demo-bucket.s3.us-
east-1.amazonaws.com",
                        "OriginPath": "/{{tenantName}}",
                        "CustomHeaders": {
                            "Quantity": 0
```

```
                },
                "S3OriginConfig": {
                    "OriginAccessIdentity": ""
                },
                "ConnectionAttempts": 3,
                "ConnectionTimeout": 10,
                "OriginShield": {
                    "Enabled": false
                },
                "OriginAccessControlId": ""
            }
        ]
    },
    "OriginGroups": {
        "Quantity": 0
    },
    "DefaultCacheBehavior": {
        "TargetOriginId": "amzn-s3-demo-bucket.s3.us-east-1.amazonaws.com",
        "TrustedKeyGroups": {
            "Enabled": false,
            "Quantity": 0
        },
        "ViewerProtocolPolicy": "allow-all",
        "AllowedMethods": {
            "Quantity": 2,
            "Items": ["HEAD", "GET"],
            "CachedMethods": {
                "Quantity": 2,
                "Items": ["HEAD", "GET"]
            }
        },
        "Compress": false,
        "LambdaFunctionAssociations": {
            "Quantity": 0
        },
        "FunctionAssociations": {
            "Quantity": 0
        },
        "FieldLevelEncryptionId": "",
        "CachePolicyId": "658327ea-f89d-4fab-a63d-7e88639e5ABC",
        "GrpcConfig": {
            "Enabled": false
        }
    },
```

```
            "CacheBehaviors": {
                "Quantity": 0
            },
            "CustomErrorResponses": {
                "Quantity": 0
            },
            "Comment": "CLI example distribution",
            "Logging": {
                "Enabled": false,
                "IncludeCookies": false,
                "Bucket": "",
                "Prefix": ""
            },
            "Enabled": true,
            "ViewerCertificate": {
                "CloudFrontDefaultCertificate": false,
                "ACMCertificateArn": "arn:aws:acm:us-
east-1:123456789012:certificate/1954f095-11b6-4daf-9952-0c308a00abc",
                "SSLSupportMethod": "sni-only",
                "MinimumProtocolVersion": "TLSv1.2_2021",
                "Certificate": "arn:aws:acm:us-
east-1:123456789012:certificate/1954f095-11b6-4daf-9952-0c308a00abc",
                "CertificateSource": "acm"
            },
            "Restrictions": {
                "GeoRestriction": {
                    "RestrictionType": "none",
                    "Quantity": 0
                }
            },
            "WebACLId": "",
            "HttpVersion": "http2",
            "TenantConfig": {
                "ParameterDefinitions": [
                    {
                        "Name": "tenantName",
                        "Definition": {
                            "StringSchema": {
                                "Comment": "tenantName parameter",
                                "DefaultValue": "root",
                                "Required": false
                            }
                        }
                    }
```

```
                ]
            },
            "ConnectionMode": "tenant-only"
        }
    }
}
```

자세한 내용은 Amazon CloudFront 개발자 안내서의 [배포 작업]을 참조하세요.

예제 4: 인증서 없이 CloudFront 다중 테넌트 배포를 생성하는 방법

다음 `create-distribution` 예제에서는 TLS 인증서 없이 다중 테넌트를 지원하는 CloudFront 배포를 생성합니다.

```
aws cloudfront create-distribution \
    --distribution-config file://dist-config.json
```

`dist-config.json`의 콘텐츠:

```
{
    "CallerReference": "cli-example",
    "Comment": "CLI example distribution",
    "DefaultRootObject": "index.html",
    "Origins": {
        "Quantity": 1,
        "Items": [
            {
                "Id": "amzn-s3-demo-bucket.s3.us-east-1.amazonaws.com",
                "DomainName": "amzn-s3-demo-bucket.s3.us-east-1.amazonaws.com",
                "OriginPath": "/{{tenantName}}",
                "CustomHeaders": {
                    "Quantity": 0
                },
                "S3OriginConfig": {
                    "OriginAccessIdentity": ""
                }
            }
        ]
    },
    "DefaultCacheBehavior": {
        "TargetOriginId": "amzn-s3-demo-bucket.s3.us-east-1.amazonaws.com",
        "CachePolicyId": "658327ea-f89d-4fab-a63d-7e88639e5ABC",
        "ViewerProtocolPolicy": "allow-all",
```

```
        "AllowedMethods": {
            "Quantity": 2,
            "Items": [
                "HEAD",
                "GET"
            ],
            "CachedMethods": {
                "Quantity": 2,
                "Items": [
                    "HEAD",
                    "GET"
                ]
            }
        }
    },
    "Enabled": true,
    "HttpVersion": "http2",
    "ConnectionMode": "tenant-only",
    "TenantConfig": {
        "ParameterDefinitions": [
            {
                "Name": "tenantName",
                "Definition": {
                    "StringSchema": {
                        "Comment": "tenantName parameter",
                        "DefaultValue": "root",
                        "Required": false
                    }
                }
            }
        ]
    }
}
```

출력:

```
{
    "Location": "https://cloudfront.amazonaws.com/2020-05-31/distribution/
E2GJ5J9QN12ABC",
    "ETag": "E37YLVVQIABC",
    "Distribution": {
        "Id": "E2GJ5J9QNABC",
        "ARN": "arn:aws:cloudfront::123456789012:distribution/E2GJ5J9QN12ABC",
```

```
        "Status": "InProgress",
        "LastModifiedTime": "2025-07-10T20:35:20.565000+00:00",
        "InProgressInvalidationBatches": 0,
        "DomainName": "example.com",
        "ActiveTrustedSigners": {
            "Enabled": false,
            "Quantity": 0
        },
        "ActiveTrustedKeyGroups": {
            "Enabled": false,
            "Quantity": 0
        },
        "DistributionConfig": {
            "CallerReference": "cli-example-no-cert",
            "DefaultRootObject": "index.html",
            "Origins": {
                "Quantity": 1,
                "Items": [
                    {
                        "Id": "amzn-s3-demo-bucket.s3.us-east-1.amazonaws.com",
                        "DomainName": "amzn-s3-demo-bucket.s3.us-
east-1.amazonaws.com",
                        "OriginPath": "/{{tenantName}}",
                        "CustomHeaders": {
                            "Quantity": 0
                        },
                        "S3OriginConfig": {
                            "OriginAccessIdentity": ""
                        },
                        "ConnectionAttempts": 3,
                        "ConnectionTimeout": 10,
                        "OriginShield": {
                            "Enabled": false
                        },
                        "OriginAccessControlId": ""
                    }
                ]
            },
            "OriginGroups": {
                "Quantity": 0
            },
            "DefaultCacheBehavior": {
                "TargetOriginId": "amzn-s3-demo-bucket.s3.us-east-1.amazonaws.com",
                "TrustedKeyGroups": {
```

```
                    "Enabled": false,
                    "Quantity": 0
                },
                "ViewerProtocolPolicy": "allow-all",
                "AllowedMethods": {
                    "Quantity": 2,
                    "Items": [
                        "HEAD",
                        "GET"
                    ],
                    "CachedMethods": {
                        "Quantity": 2,
                        "Items": [
                            "HEAD",
                            "GET"
                        ]
                    }
                },
                "Compress": false,
                "LambdaFunctionAssociations": {
                    "Quantity": 0
                },
                "FunctionAssociations": {
                    "Quantity": 0
                },
                "FieldLevelEncryptionId": "",
                "CachePolicyId": "658327ea-f89d-4fab-a63d-7e88639e5ABC",
                "GrpcConfig": {
                    "Enabled": false
                }
            },
            "CacheBehaviors": {
                "Quantity": 0
            },
            "CustomErrorResponses": {
                "Quantity": 0
            },
            "Comment": "CLI example distribution",
            "Logging": {
                "Enabled": false,
                "IncludeCookies": false,
                "Bucket": "",
                "Prefix": ""
            },
```

```
            "Enabled": true,
            "ViewerCertificate": {
                "CloudFrontDefaultCertificate": true,
                "SSLSupportMethod": "sni-only",
                "MinimumProtocolVersion": "TLSv1",
                "CertificateSource": "cloudfront"
            },
            "Restrictions": {
                "GeoRestriction": {
                    "RestrictionType": "none",
                    "Quantity": 0
                }
            },
            "WebACLId": "",
            "HttpVersion": "http2",
            "TenantConfig": {
                "ParameterDefinitions": [
                    {
                        "Name": "tenantName",
                        "Definition": {
                            "StringSchema": {
                                "Comment": "tenantName parameter",
                                "DefaultValue": "root",
                                "Required": false
                            }
                        }
                    }
                ]
            },
            "ConnectionMode": "tenant-only"
        }
    }
}
```

자세한 내용은 Amazon CloudFront 개발자 안내서의 배포 구성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateDistribution을 참조하세요.

create-field-level-encryption-config

다음 코드 예시에서는 `create-field-level-encryption-config` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

CloudFront 필드 수준 암호화 구성 생성

다음 예시에서는 JSON 파일 `fle-config.json`에 구성 파라미터를 제공하여 필드 수준 암호화 구성을 생성합니다. 필드 수준 암호화 구성을 생성하려면 먼저 필드 수준 암호화 프로파일이 있어야 합니다. 프로파일을 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 보호]를 참조하세요.

```
aws cloudfront create-field-level-encryption-config \
    --field-level-encryption-config file://fle-config.json
```

`fle-config.json` 파일은 다음을 포함한 현재 문서의 JSON 문서입니다.

```
{
    "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"
                        }
                    ]
                }
            }
        }
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [CreateFieldLevelEncryptionConfig](#) 섹션을 참조하세요.

`create-field-level-encryption-profile`

다음 코드 예시에서는 `create-field-level-encryption-profile` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

CloudFront 필드 수준 암호화 프로파일을 생성하는 방법

다음 예시에서는 JSON 파일 `fle-profile-config.json`에 파라미터를 제공하여 필드 수준 암호화 프로파일을 생성합니다. 필드 수준 암호화 프로파일을 생성하려면 먼저 CloudFront 퍼블릭 키가 있어야 합니다. CloudFront 퍼블릭 키를 생성하려면 create-public-key 명령을 참조하세요.

CloudFront 필드 수준 암호화에 대한 자세한 내용은 Amazon CloudFront 개발자 안내서의 [필드 수준 암호화를 사용하여 민감한 데이터 보호]를 참조하세요.

```
aws cloudfront create-field-level-encryption-profile \
    --field-level-encryption-profile-config file://fle-profile-config.json
```

`fle-profile-config.json` 파일은 다음을 포함한 현재 문서의 JSON 문서입니다.

```
{
    "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"
                            ]
                        }
                    }
                ]
            }
        }
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateFieldLevelEncryptionProfile 섹션을 참조하세요.

create-invalidation-for-distribution-tenant

다음 코드 예시는 create-invalidation-for-distribution-tenant의 사용 방법을 보여 줍니다.

AWS CLI

배포 테넌트에 대한 CloudFront 무효화를 만들려면

다음 create-invalidation-for-distribution-tenant 예제는 CloudFront 배포 테넌트의 모든 파일에 대한 무효화를 만듭니다.

```
aws cloudfront create-invalidation-for-distribution-tenant \
    --id dt_2wjDZi3hD1ivOXf6rpZJO1AB \
```

```
    --invalidation-batch '{"Paths": {"Quantity": 1, "Items": ["/*"]},
 "CallerReference": "invalidation-$(date +%s)"}'
```

출력:

```
{
    "Location": "https://cloudfront.amazonaws.com/2020-05-31/distribution-tenant/
dt_2wjDZi3hD1ivOXf6rpZJO1AB/invalidation/I2JGL2F1ZAA426PGG0YLLKABC",
    "Invalidation": {
        "Id": "I2JGL2F1ZAA426PGG0YLLKABC",
        "Status": "InProgress",
        "CreateTime": "2025-05-07T16:59:25.947000+00:00",
        "InvalidationBatch": {
            "Paths": {
                "Quantity": 1,
                "Items": [
                    "/*"
                ]
            },
            "CallerReference": "invalidation-$(date +%s)"
        }
    }
}
```

자세한 내용은 Amazon CloudFront 개발자 안내서의 Invalidate files to remove content를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateInvalidationForDistributionTenant 섹션을 참조하세요.

`create-invalidation`

다음 코드 예시는 `create-invalidation`의 사용 방법을 보여 줍니다.

AWS CLI

CloudFront 배포에 대한 무효화를 생성하려면

다음 `create-invalidation` 예제는 지정된 CloudFront 배포의 지정된 파일에 대한 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 예제에서는 AWS CLI가 자동으로 랜덤 `CallerReference`를 생성했습니다. 직접 `CallerReference`를 지정하거나 무효화 매개변수를 명령줄 인수로 전달하지 않으려면 JSON 파일을 사용할 수 있습니다. 다음 예제에서는 `inv-batch.json`라는 JSON 파일에 무효화 매개 변수를 제공하여 두 파일에 대한 무효화를 생성합니다

```
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"
        }
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의CreateInvalidation 섹션을 참조하세요.

`create-public-key`

다음 코드 예시에서는 `create-public-key` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

CloudFront 퍼블릭 키를 생성하려면

다음 예제에서는 `pub-key-config.json`이라는 JSON 파일로 파라미터를 제공하여 CloudFront 퍼블릭 키를 생성합니다. 이 명령을 사용하려면 먼저 PEM으로 인코딩된 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
```

`pub-key-config.json` 파일은 다음을 포함한 현재 문서의 JSON 문서입니다. 참고로 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"
        }
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [CreatePublicKey](https://docs.aws.amazon.com/goto/aws-cli/cloudfront-2020-05-31/CreatePublicKey)를 참조하세요.

delete-cloud-front-origin-access-identity

다음 코드 예시에서는 delete-cloud-front-origin-access-identity 코드를 사용하는 방법을 보여줍니다.

AWS CLI

CloudFront 원본 액세스 ID 삭제

다음 예시에서는 ID가 E74FTE3AEXAMPLE인 오리진 액세스 ID(OAI)를 삭제하기 위한 요청입니다. OAI를 삭제하는 방법 OAI의 ID와 ETag가 있어야 합니다. OAI ID는 create-cloud-front-origin-access-identity 및 list-cloud-front-origin-access-identities 명령의 출력에 반환됩니다. ETag를 가져오려면 get-cloud-front-origin-access-identity 또는 get-cloud-front-origin-access-identity-config 명령을 사용합니다. --if-match 옵션을 사용하여 OAI의 ETag를 제공합니다.

```
aws cloudfront delete-cloud-front-origin-access-identity \
    --id E74FTE3AEXAMPLE \
    --if-match E2QWRUHEXAMPLE
```

이 명령이 제대로 실행되면 출력이 표시되지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteCloudFrontOriginAccessIdentity 섹션을 참조하세요.

delete-connection-group

다음 코드 예시는 delete-connection-group의 사용 방법을 보여 줍니다.

AWS CLI

연결 그룹을 삭제하려면

다음 delete-connection-group 예제에서는 연결 그룹을 삭제합니다. 연결 그룹은 비활성화되어 있어야 하며 CloudFront 리소스와 연결할 수 없습니다.

```
aws cloudfront delete-connection-group \
    --id cg_2wjLpjbHkLUdhWAjHllcOeABC \
    --if-match ETVPDKIKX0DABC
```

이 명령이 제대로 실행되면 출력이 표시되지 않습니다.

연결 그룹 관리에 대한 자세한 내용은 Amazon CloudFront 개발자 안내서의 Create custom connection group (optional)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteConnectionGroup 섹션을 참조하세요.

delete-distribution-tenant

다음 코드 예시는 delete-distribution-tenant의 사용 방법을 보여 줍니다.

AWS CLI

배포 테넌트를 삭제하려면

다음 delete-distribution-tenant 예제에서는 ETag가 ETVPDKIKX0DABC인 배포 테넌트를 삭제합니다. 배포 테넌트는 비활성화되어 있어야 하며 CloudFront 리소스와 연결할 수 없습니다.

```
aws cloudfront delete-distribution-tenant \
    --id dt_2wjMUbg3NHZEQ7OfoalP5zi1AB \
    --if-match ETVPDKIKX0DABC
```

이 명령이 제대로 실행되면 출력이 표시되지 않습니다.

자세한 내용은 Amazon CloudFront 개발자 안내서의 Delete a distribution을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteDistributionTenant 섹션을 참조하세요.

delete-distribution

다음 코드 예시는 delete-distribution의 사용 방법을 보여 줍니다.

AWS CLI

CloudFront 배포를 삭제하려면

다음 예제에서는 ID EDFDVBD6EXAMPLE을 사용하여 CloudFront 배포를 삭제합니다. 배포를 삭제하려면 먼저 배포를 비활성화해야 합니다. 배포를 비활성화하려면 update-distribution 명령을 사용하세요. 자세한 정보는 update-distribution 예시를 참조하세요.

배포가 비활성화된 경우 이를 삭제할 수 있습니다. 배포를 삭제하려면 배포의 ETag를 제공하는 --if-match 옵션을 사용해야 합니다. ETag를 가져오려면 get-distribution 또는 get-distribution-config 명령을 사용하세요.

```
aws cloudfront delete-distribution \
```

```
    --id EDFDVBD6EXAMPLE \
    --if-match E2QWRUHEXAMPLE
```

이 명령이 제대로 실행되면 출력이 표시되지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteDistribution 섹션을 참조하세요.

delete-field-level-encryption-config

다음 코드 예시에서는 delete-field-level-encryption-config 코드를 사용하는 방법을 보여줍니다.

AWS CLI

CloudFront 필드 수준 암호화 구성을 삭제하는 방법

다음 예시에서는 ID가 C3KM2WVD605UAY인 CloudFront 필드 수준 암호화 구성을 삭제합니다. 필드 수준 암호화 구성을 삭제하려면 ID와 ETag가 있어야 합니다. ID는 create-field-level-encryption-config 및 list-field-level-encryption-configs 명령의 출력으로 반환됩니다. ETag를 가져오려면 get-field-level-encryption 또는 get-field-level-encryption-config 명령을 사용합니다. --if-match 옵션을 사용하여 구성의 ETag를 제공합니다.

```
aws cloudfront delete-field-level-encryption-config \
    --id C3KM2WVD605UAY \
    --if-match E26M4BIAV81ZF6
```

이 명령이 제대로 실행되면 출력이 표시되지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteFieldLevelEncryptionConfig 섹션을 참조하세요.

delete-field-level-encryption-profile

다음 코드 예시에서는 delete-field-level-encryption-profile 코드를 사용하는 방법을 보여줍니다.

AWS CLI

CloudFront 필드 수준 암호화 프로파일을 삭제하는 방법

다음 예시에서는 ID가 PPK0UOSIF5WSV인 CloudFront 필드 수준 암호화 프로파일을 삭제합니다. 필드 수준 암호화 프로파일을 삭제하려면 ID와 ETag가 있어야 합니다. ID는 create-field-level-encryption-profile 및 list-field-level-encryption-profiles 명령의 출력으로 반환됩니다. ETag를 가져오려면 get-field-level-encryption-profile 또는 get-field-level-encryption-profile-config 명령을 사용합니다. --if-match 옵션을 사용하여 프로파일의 ETag를 제공합니다.

```
aws cloudfront delete-field-level-encryption-profile \
    --id PPK0UOSIF5WSV \
    --if-match EJETYFJ9CL66D
```

이 명령이 제대로 실행되면 출력이 표시되지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteFieldLevelEncryptionProfile](https://docs.aws.amazon.com/goto/aws-cli/cloudfront-2020-05-31/DeleteFieldLevelEncryptionProfile) 섹션을 참조하세요.

delete-public-key

다음 코드 예시에서는 delete-public-key 코드를 사용하는 방법을 보여줍니다.

AWS CLI

CloudFront 퍼블릭 키 삭제

다음 예시에서는 ID가 KDFB19YGCR002인 CloudFront 퍼블릭 키를 삭제합니다. 퍼블릭 키를 삭제하려면 ID와 ETag가 있어야 합니다. ID는 create-public-key 및 list-public-keys 명령의 출력으로 반환됩니다. ETag를 가져오려면 get-public-key 또는 get-public-key-config 명령을 사용합니다. --if-match 옵션을 사용하여 퍼블릭 키의 ETag를 제공합니다.

```
aws cloudfront delete-public-key \
    --id KDFB19YGCR002 \
    --if-match E2QWRUHEXAMPLE
```

이 명령이 제대로 실행되면 출력이 표시되지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 [DeletePublicKey](https://docs.aws.amazon.com/goto/aws-cli/cloudfront-2020-05-31/DeletePublicKey) 섹션을 참조하세요.

disassociate-distribution-tenant-web-acl

다음 코드 예시는 disassociate-distribution-tenant-web-acl의 사용 방법을 보여 줍니다.

AWS CLI

배포 테넌트에서 웹 ACL의 연결을 해제하려면

다음 `disassociate-distribution-tenant-web-acl` 예제에서는 ETag가 `E1PA6795UKMABC`인 배포 테넌트에서 웹 ACL의 연결을 해제합니다.

```
aws cloudfront disassociate-distribution-tenant-web-acl \
    --id dt_2wjDZi3hD1ivOXf6rpZJOSNE1AB \
    --if-match E1PA6795UKMABC
```

출력:

```
{
    "ETag": "E13V1IB3VIYABC",
    "Id": "dt_2wjDZi3hD1ivOXf6rpZJOSNE1AB"
}
```

자세한 내용은 Amazon CloudFront 개발자 안내서의 [Disable AWS WAF security protections](https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/disable-waf.html)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DisassociateDistributionTenantWebAcl](https://awscli.amazonaws.com/v2/documentation/api/latest/reference/cloudfront/disassociate-distribution-tenant-web-acl.html) 섹션을 참조하세요.

`disassociate-distribution-web-acl`

다음 코드 예시는 `disassociate-distribution-web-acl`의 사용 방법을 보여 줍니다.

AWS CLI

CloudFront 배포에서 웹 ACL의 연결을 해제하려면

다음 `disassociate-distribution-web-acl` 예제에서는 웹 ACL과 ETag가 `E13V1IB3VIYABC`인 CloudFront 배포 간의 연결을 삭제합니다.

```
aws cloudfront disassociate-distribution-web-acl \
    --id E1XNX8R2GOAABC \
    --if-match EEZQ9Z24VM1ABC
```

출력:

```
{
    "ETag": "E2YWS1C2J3OABC",
    "Id": "E1XNX8R2GOAABC"
}
```

자세한 내용은 Amazon CloudFront 개발자 안내서의 Disable AWS WAF security protections를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisassociateDistributionWebAcl 섹션을 참조하세요.

get-cloud-front-origin-access-identity-config

다음 코드 예시는 get-cloud-front-origin-access-identity-config의 사용 방법을 보여 줍니다.

AWS CLI

CloudFront 원본 액세스 ID 구성을 가져오려면

다음 예제에서는 ID E74FTE3AEXAMPLE을 사용하여 ETag를 포함한 CloudFront 오리진 액세스 ID(OAI)에 대한 메타데이터를 가져옵니다. OAI ID는 create-cloud-front-origin-access-identity 및 list-cloud-front-origin-access-identities 명령의 출력에 반환됩니다.

```
aws cloudfront get-cloud-front-origin-access-identity-config --id E74FTE3AEXAMPLE
```

출력:

```
{
    "ETag": "E2QWRUHEXAMPLE",
    "CloudFrontOriginAccessIdentityConfig": {
        "CallerReference": "cli-example",
        "Comment": "Example OAI"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetCloudFrontOriginAccessIdentityConfig 섹션을 참조하세요.

get-cloud-front-origin-access-identity

다음 코드 예시에서는 get-cloud-front-origin-access-identity 코드를 사용하는 방법을 보여줍니다.

AWS CLI

CloudFront 원본 액세스 ID를 가져오려면

다음 예제는 ID가 E74FTE3AEXAMPLE인 CloudFront 오리진 액세스 ID(OAI) 및 그에 딸린 ETag와 관련된 S3 카노니컬 ID를 가져옵니다. OAI ID는 create-cloud-front-origin-access-identity 및 list-cloud-front-origin-access-identities 명령의 출력에 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"
        }
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [GetCloudFrontOriginAccessIdentity](#) 섹션을 참조하세요.

get-connection-group-by-routing-endpoint

다음 코드 예시는 `get-connection-group-by-routing-endpoint`의 사용 방법을 보여 줍니다.

AWS CLI

엔드포인트를 라우팅하여 연결 그룹을 가져오려면

다음 `get-connection-group-by-routing-endpoint` 예제에서는 라우팅 엔드포인트를 사용하여 연결 그룹에 대한 정보를 검색합니다.

```
aws cloudfront get-connection-group-by-routing-endpoint \
    --routing-endpoint dvdg9gprgabc.cloudfront.net
```

출력:

```
{
    "ETag": "E23ZP02F085ABC",
```

```
    "ConnectionGroup": {
        "Id": "cg_2wjDWTBKTlRB87cAaUQFaakABC",
        "Name": "connection-group-2",
        "Arn": "arn:aws:cloudfront::123456789012:connection-group/
cg_2wjDWTBKTlRB87cAaUQFaakABC",
        "CreatedTime": "2025-05-06T15:42:00.790000+00:00",
        "LastModifiedTime": "2025-05-06T15:42:00.790000+00:00",
        "Ipv6Enabled": true,
        "RoutingEndpoint": "dvdg9gprgabc.cloudfront.net",
        "Status": "Deployed",
        "Enabled": true,
        "IsDefault": false
    }
}
```

자세한 내용은 Amazon CloudFront 개발자 안내서의 Create custom connection group (optional)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetConnectionGroupByRoutingEndpoint 섹션을 참조하세요.

get-connection-group

다음 코드 예시는 get-connection-group의 사용 방법을 보여 줍니다.

AWS CLI

CloudFront 연결 그룹을 가져오려면

다음 get-connection-group 예제에서는 CloudFront 연결 그룹에 대한 정보를 검색합니다.

```
aws cloudfront get-connection-group \
    --identifier cg_2wjDWTBKTlRB87cAaUQFaakABC
```

출력:

```
{
    "ETag": "E23ZP02F085ABC",
    "ConnectionGroup": {
        "Id": "cg_2wjDWTBKTlRB87cAaUQFaakABC",
        "Name": "connection-group-2",
        "Arn": "arn:aws:cloudfront::123456789012:connection-group/
cg_2wjDWTBKTlRB87cAaUQFaakABC",
```

```
        "CreatedTime": "2025-05-06T15:42:00.790000+00:00",
        "LastModifiedTime": "2025-05-06T15:42:00.790000+00:00",
        "Ipv6Enabled": true,
        "RoutingEndpoint": "dvdg9gprgabc.cloudfront.net",
        "Status": "Deployed",
        "Enabled": true,
        "IsDefault": false
    }
}
```

자세한 내용은 Amazon CloudFront 개발자 안내서의 Create custom connection group (optional)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetConnectionGroup 섹션을 참조하세요.

`get-distribution-config`

다음 코드 예시는 `get-distribution-config`의 사용 방법을 보여 줍니다.

AWS CLI

CloudFront 배포 구성을 가져오려면

다음 예제에서는 ID `EDFDVBD6EXAMPLE`을 사용하여 `ETag`를 포함한 CloudFront 배포에 대한 메타데이터를 가져옵니다. 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": "amzn-s3-demo-bucket.s3.amazonaws.com-cli-example",
                    "DomainName": "amzn-s3-demo-bucket.s3.amazonaws.com",
                    "OriginPath": "",
                    "CustomHeaders": {
                        "Quantity": 0
                    },
                    "S3OriginConfig": {
                        "OriginAccessIdentity": ""
                    }
                }
            ]
        },
        "OriginGroups": {
            "Quantity": 0
        },
        "DefaultCacheBehavior": {
            "TargetOriginId": "amzn-s3-demo-bucket.s3.amazonaws.com-cli-example",
            "ForwardedValues": {
                "QueryString": false,
                "Cookies": {
                    "Forward": "none"
                },
                "Headers": {
                    "Quantity": 0
                },
                "QueryStringCacheKeys": {
                    "Quantity": 0
                }
            },
            "TrustedSigners": {
                "Enabled": false,
                "Quantity": 0
            },
            "ViewerProtocolPolicy": "allow-all",
            "MinTTL": 0,
            "AllowedMethods": {
                "Quantity": 2,
                "Items": [
                    "HEAD",
                    "GET"
                ],
                "CachedMethods": {
                    "Quantity": 2,
                    "Items": [
```

```
                        "HEAD",
                        "GET"
                    ]
                }
            },
            "SmoothStreaming": false,
            "DefaultTTL": 86400,
            "MaxTTL": 31536000,
            "Compress": false,
            "LambdaFunctionAssociations": {
                "Quantity": 0
            },
            "FieldLevelEncryptionId": ""
        },
        "CacheBehaviors": {
            "Quantity": 0
        },
        "CustomErrorResponses": {
            "Quantity": 0
        },
        "Comment": "",
        "Logging": {
            "Enabled": false,
            "IncludeCookies": false,
            "Bucket": "",
            "Prefix": ""
        },
        "PriceClass": "PriceClass_All",
        "Enabled": true,
        "ViewerCertificate": {
            "CloudFrontDefaultCertificate": true,
            "MinimumProtocolVersion": "TLSv1",
            "CertificateSource": "cloudfront"
        },
        "Restrictions": {
            "GeoRestriction": {
                "RestrictionType": "none",
                "Quantity": 0
            }
        },
        "WebACLId": "",
        "HttpVersion": "http2",
        "IsIPV6Enabled": true
    }
```

```
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetDistributionConfig를 참조하세요.

get-distribution-tenant-by-domain

다음 코드 예시는 `get-distribution-tenant-by-domain`의 사용 방법을 보여 줍니다.

AWS CLI

도메인별로 배포 테넌트 정보를 가져오려면

다음 `get-distribution-tenant-by-domain` 예제에서는 지정된 도메인을 사용하여 배포 테넌트에 대한 정보를 검색합니다.

```
aws cloudfront get-distribution-tenant-by-domain \
    --domain example.com
```

출력:

```
{
    "ETag": "E23ZP02F085ABC",
    "DistributionTenant": {
        "Id": "dt_2xVInRKCfUzQHgxosDs9hiLk1AB",
        "DistributionId": "E1XNX8R2GOAABC",
        "Name": "example-tenant-4",
        "Arn": "arn:aws:cloudfront::123456789012:distribution-tenant/
dt_2xVInRKCfUzQHgxosDs9hiLk1AB",
        "Domains": [
            {
                "Domain": "example.com",
                "Status": "active"
            }
        ],
        "Parameters": [
            {
                "Name": "testParam",
                "Value": "defaultValue"
            }
        ],
        "ConnectionGroupId": "cg_2whCJoXMYCjHcxaLGrkllvyABC",
        "CreatedTime": "2025-05-23T16:16:20.871000+00:00",
```

```
        "LastModifiedTime": "2025-05-23T16:16:20.871000+00:00",
        "Enabled": false,
        "Status": "Deployed"
    }
}
```

자세한 내용은 Amazon CloudFront 개발자 안내서의 Understand how multi-tenant distributions work를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetDistributionTenantByDomain 섹션을 참조하세요.

get-distribution-tenant

다음 코드 예시는 `get-distribution-tenant`의 사용 방법을 보여 줍니다.

AWS CLI

CloudFront 배포 테넌트에 대한 세부 정보를 가져오려면

다음 `get-distribution-tenant` 예제에서는 CloudFront 배포 테넌트에 대한 정보를 검색합니다.

```
aws cloudfront get-distribution-tenant \
    --id dt_2wjDZi3hD1ivOXf6rpZJOSNE1AB
```

출력:

```
{
    "ETag": "E23ZP02F085ABC",
    "DistributionTenant": {
        "Id": "dt_2wjDZi3hD1ivOXf6rpZJOSNE1AB",
        "DistributionId": "E1XNX8R2GOAABC",
        "Name": "example-tenant-2",
        "Arn": "arn:aws:cloudfront::123456789012:distribution-tenant/
dt_2wjDZi3hD1ivOXf6rpZJOSNE1AB",
        "Domains": [
            {
                "Domain": "example.com",
                "Status": "inactive"
            }
        ],
        "ConnectionGroupId": "cg_2wjDWTBKTlRB87cAaUQFaakABC",
        "CreatedTime": "2025-05-06T15:42:28.542000+00:00",
```

```
        "LastModifiedTime": "2025-05-06T15:42:37.724000+00:00",
        "Enabled": true,
        "Status": "InProgress"
    }
}
```

자세한 내용은 Amazon CloudFront 개발자 안내서의 [Understand how multi-tenant distributions work](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [GetDistributionTenant](#) 섹션을 참조하세요.

get-distribution

다음 코드 예시는 `get-distribution`의 사용 방법을 보여 줍니다.

AWS CLI

CloudFront 배포를 가져오려면

다음 `get-distribution` 예제에서는 `ETag`를 비롯해 `EDFDVBD6EXAMPLE` ID를 가진 CloudFront 배포를 가져옵니다. 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": "amzn-s3-demo-bucket.s3.amazonaws.com-cli-example",
                        "DomainName": "amzn-s3-demo-bucket.s3.amazonaws.com",
                        "OriginPath": "",
                        "CustomHeaders": {
                            "Quantity": 0
                        },
                        "S3OriginConfig": {
                            "OriginAccessIdentity": ""
                        }
                    }
                ]
            },
            "OriginGroups": {
                "Quantity": 0
            },
            "DefaultCacheBehavior": {
                "TargetOriginId": "amzn-s3-demo-bucket.s3.amazonaws.com-cli-
example",
                "ForwardedValues": {
                    "QueryString": false,
                    "Cookies": {
                        "Forward": "none"
                    },
                    "Headers": {
                        "Quantity": 0
                    },
                    "QueryStringCacheKeys": {
                        "Quantity": 0
                    }
                },
                "TrustedSigners": {
                    "Enabled": false,
                    "Quantity": 0
                },
                "ViewerProtocolPolicy": "allow-all",
                "MinTTL": 0,
                "AllowedMethods": {
```

```
                "Quantity": 2,
                "Items": [
                    "HEAD",
                    "GET"
                ],
                "CachedMethods": {
                    "Quantity": 2,
                    "Items": [
                        "HEAD",
                        "GET"
                    ]
                }
            },
            "SmoothStreaming": false,
            "DefaultTTL": 86400,
            "MaxTTL": 31536000,
            "Compress": false,
            "LambdaFunctionAssociations": {
                "Quantity": 0
            },
            "FieldLevelEncryptionId": ""
        },
        "CacheBehaviors": {
            "Quantity": 0
        },
        "CustomErrorResponses": {
            "Quantity": 0
        },
        "Comment": "",
        "Logging": {
            "Enabled": false,
            "IncludeCookies": false,
            "Bucket": "",
            "Prefix": ""
        },
        "PriceClass": "PriceClass_All",
        "Enabled": true,
        "ViewerCertificate": {
            "CloudFrontDefaultCertificate": true,
            "MinimumProtocolVersion": "TLSv1",
            "CertificateSource": "cloudfront"
        },
        "Restrictions": {
            "GeoRestriction": {
```

```
                    "RestrictionType": "none",
                    "Quantity": 0
                }
            },
            "WebACLId": "",
            "HttpVersion": "http2",
            "IsIPV6Enabled": true
        }
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetDistribution 섹션을 참조하세요.

get-field-level-encryption-config

다음 코드 예시에서는 `get-field-level-encryption-config` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

CloudFront 필드 수준 암호화 구성에 대한 메타데이터를 가져오는 방법

다음 예시에서는 ID가 `C3KM2WVD605UAY`인 CloudFront 필드 수준 암호화 구성에 대한 메타데이터 (`ETag` 포함)를 가져옵니다.

```
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"
                        }
                    ]
                }
            }
        }
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetFieldLevelEncryptionConfig 섹션을 참조하세요.

`get-field-level-encryption-profile-config`

다음 코드 예시에서는 `get-field-level-encryption-profile-config` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

CloudFront 필드 수준 암호화 프로파일 구성을 가져오는 방법

다음 예시에서는 ID가 `PPK0UOSIF5WSV`인 CloudFront 필드 수준 암호화 프로파일에 대한 메타데이터(`ETag` 포함)를 가져옵니다.

```
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"
                        ]
                    }
                }
            ]
        }
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetFieldLevelEncryptionProfileConfig 섹션을 참조하세요.

get-field-level-encryption-profile

다음 코드 예시에서는 get-field-level-encryption-profile 코드를 사용하는 방법을 보여줍니다.

AWS CLI

CloudFront 필드 수준 암호화 프로파일을 가져오는 방법

다음 예시에서는 ID가 PPK0UOSIF5WSV인 CloudFront 필드 수준 암호화 프로파일(ETag 포함)을 가져옵니다.

```
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"
                            ]
                        }
                    }
                ]
            }
        }
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetFieldLevelEncryptionProfile 섹션을 참조하세요.

`get-field-level-encryption`

다음 코드 예시에서는 `get-field-level-encryption` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

CloudFront 필드 수준 암호화 구성을 가져오는 방법

다음 예시에서는 ID가 `C3KM2WVD605UAY`인 CloudFront 필드 수준 암호화 구성(ETag 포함)을 가져옵니다.

```
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"
                        }
                    ]
                }
            }
        }
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetFieldLevelEncryption 섹션을 참조하세요.

get-invalidation-for-distribution-tenant

다음 코드 예시는 `get-invalidation-for-distribution-tenant`의 사용 방법을 보여 줍니다.

AWS CLI

배포 테넌트에 대한 무효화를 가져오려면

다음 `get-invalidation-for-distribution-tenant` 예제에서는 배포 테넌트의 무효화에 대한 정보를 가져옵니다.

```
aws cloudfront get-invalidation-for-distribution-tenant \
    --distribution-tenant-id dt_2wjDZi3hD1ivOXf6rpZJOSNE1AB \
```

```
    --id I4CU23QAPKMUDUU06F9OFGFABC
```

출력:

```
{
    "Invalidation": {
        "Id": "I4CU23QAPKMUDUU06F9OFGFABC",
        "Status": "Completed",
        "CreateTime": "2025-05-06T15:46:12.824000+00:00",
        "InvalidationBatch": {
            "Paths": {
                "Quantity": 2,
                "Items": [
                    "/example/invalidation",
                    "/more/invalidations"
                ]
            },
            "CallerReference": "007ee5a6-d0a0-42be-bb61-e7b915969b48"
        }
    }
}
```

자세한 내용은 Amazon CloudFront 개발자 안내서의 Invalidate files to remove content를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetInvalidationForDistributionTenant 섹션을 참조하세요.

get-invalidation

다음 코드 예시는 get-invalidation의 사용 방법을 보여 줍니다.

AWS CLI

CloudFront 무효화를 가져오는 방법

다음 예시에서는 ID가 EDFDVBD6EXAMPLE인 CloudFront 배포의 ID I2J0I21PCUYOIK를 통해 무효화를 가져옵니다.

```
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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetInvalidation 섹션을 참조하세요.

get-managed-certificate-details

다음 코드 예시는 `get-managed-certificate-details`의 사용 방법을 보여 줍니다.

AWS CLI

관리형 인증서 세부 정보를 가져오려면

다음 `get-managed-certificate-details` 예제에서는 CloudFront 관리형 ACM 인증서의 세부 정보를 검색합니다.

```
aws cloudfront get-managed-certificate-details \
    --identifier dt_2wjDZi3hD1ivOXf6rpZJOSNE1AB
```

출력:

```
{
    "ManagedCertificateDetails": {
        "CertificateArn": "arn:aws:acm:us-
east-1:123456789012:certificate/655dc1fe-6d37-451d-a013-c2db3a034abc",
        "CertificateStatus": "pending-validation",
        "ValidationTokenHost": "self-hosted",
```

```
        "ValidationTokenDetails": [
            {
                "Domain": "example.com",
                "RedirectTo": "validation.us-east-1.acm-
validations.aws/123456789012/.well-known/pki-validation/
b315c9ae21284e7918bb9f3f422ab1c7.txt",
                "RedirectFrom": "example.com/.well-known/pki-validation/
b315c9ae21284e7918bb9f3f422ac3c7.txt"
            }
        ]
    }
}
```

자세한 내용은 Amazon CloudFront 개발자 안내서의 Request certificates for your CloudFront distribution tenant를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetManagedCertificateDetails 섹션을 참조하세요.

get-public-key-config

다음 코드 예시는 `get-public-key-config`의 사용 방법을 보여 줍니다.

AWS CLI

CloudFront 퍼블릭 키 구성을 가져오는 방법

다음 예시에서는 ID가 `KDFB19YGCR002`인 CloudFront 퍼블릭 키에 대한 메타데이터(`ETag` 포함)를 가져옵니다. 퍼블릭 키 ID는 create-public-key 및 list-public-keys 명령에 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"
        }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetPublicKeyConfig 섹션을 참조하세요.

get-public-key

다음 코드 예시에서는 get-public-key 코드를 사용하는 방법을 보여줍니다.

AWS CLI

CloudFront 퍼블릭 키 가져오기

다음 예시에서는 ID가 KDFB19YGCR002인 CloudFront 퍼블릭 키(ETag 포함)를 가져옵니다. 퍼블릭 키 ID는 create-public-key 및 list-public-keys 명령에 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/1U0WcmZX2u70dYcSIzB1ofZtcn4cJ
\nenHBAzO3ohBY/L1tQGJfS2A+omnN6H16VZE1JCK8XSJyfze7MDLcUyHZETdxuvRb
\nA9X343/vMAuQPnhinFJ8Wdy8YBXSPpy7r95ylUQd9LfYTBzVZYG2tSesplcOkjM3\n2Uu
+oMWxQAw1NINnSLPinMVsutJy6ZqlV3McWNWe4T+STGtWhrPNqJEn45sIcCx4\nq
+kGZ2NQ0FyIyT2eiLKOX5Rgb/a36E/aMk4VoDsaenBQgG7WLTnstb9sr7MIhS6A\nrwIDAQAB\n-----END
 PUBLIC KEY-----\n",
            "Comment": "example public key"
        }
```

```
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetPublicKey 섹션을 참조하세요.

`list-cloud-front-origin-access-identities`

다음 코드 예시에서는 `list-cloud-front-origin-access-identities` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

CloudFront 오리진 액세스 ID를 나열하려면

다음 예제에서는 AWS 계정의 CloudFront 원본 액세스 ID(OAI) 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"
            }
        ]
```

```
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListCloudFrontOriginAccessIdentities 섹션을 참조하세요.

list-connection-groups

다음 코드 예시는 `list-connection-groups`의 사용 방법을 보여 줍니다.

AWS CLI

연결 그룹을 나열하려면

다음 `list-connection-groups` 예제에서는 AWS 계정에서 사용 가능한 연결 그룹을 나열합니다.

```
aws cloudfront list-connection-groups
```

출력:

```
{
    "ConnectionGroups": [
        {
            "Id": "cg_2whCJoXMYCjHcxaLGrkllvyABC",
            "Name": "CreatedByCloudFront-cg_2whCJoXMYCjHcxaLGrkllvyABC",
            "Arn": "arn:aws:cloudfront::123456789012:connection-group/
cg_2whCJoXMYCjHcxaLGrkllvyABC",
            "RoutingEndpoint": "d3sx0pso7m5abc.cloudfront.net",
            "CreatedTime": "2025-05-05T22:32:29.630000+00:00",
            "LastModifiedTime": "2025-05-05T22:32:29.630000+00:00",
            "ETag": "E23ZP02F085ABC",
            "Enabled": true,
            "Status": "Deployed",
            "IsDefault": true
        },
        {
            "Id": "cg_2wjDWTBKTlRB87cAaUQFaakABC",
            "Name": "connection-group-2",
            "Arn": "arn:aws:cloudfront::123456789012:connection-group/
cg_2wjDWTBKTlRB87cAaUQFaakABC",
            "RoutingEndpoint": "dvdg9gprgabc.cloudfront.net",
```

```
            "CreatedTime": "2025-05-06T15:42:00.790000+00:00",
            "LastModifiedTime": "2025-05-06T15:42:00.790000+00:00",
            "ETag": "E23ZP02F085ABC",
            "Enabled": true,
            "Status": "Deployed",
            "IsDefault": false
        }
    ]
}
```

자세한 내용은 Amazon CloudFront 개발자 안내서의 Create custom connection group (optional)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListConnectionGroups 섹션을 참조하세요.

list-distribution-tenants-by-customization

다음 코드 예시는 list-distribution-tenants-by-customization의 사용 방법을 보여 줍니다.

AWS CLI

사용자 지정을 기준으로 배포 테넌트를 나열하려면

다음 list-distribution-tenants-by-customization 예제에서는 지정된 웹 ACL을 사용하는 배포 테넌트를 나열합니다.

```
aws cloudfront list-distribution-tenants-by-customization \
    --web-acl-arn arn:aws:wafv2:us-east-1:123456789012:global/webacl/
CreatedByCloudFront-0273cd2f/a3c19bce-42b5-48a1-a8d4-b2bb2f28eabc
```

출력:

```
{
    "DistributionTenantList": [
        {
            "Id": "dt_2wjDZi3hD1ivOXf6rpZJOSNE1AB",
            "DistributionId": "E1XNX8R2GOAABC",
            "Name": "example-tenant-2",
            "Arn": "arn:aws:cloudfront::123456789012:distribution-tenant/
dt_2wjDZi3hD1ivOXf6rpZJOSNE1AB",
            "Domains": [
                {
```

```
                    "Domain": "example.com",
                    "Status": "inactive"
                }
            ],
            "ConnectionGroupId": "cg_2wjDWTBKTlRB87cAaUQFaakABC",
            "Customizations": {
                "WebAcl": {
                    "Action": "override",
                    "Arn": "arn:aws:wafv2:us-east-1:123456789012:global/webacl/
CreatedByCloudFront-0273cd2f/a3c19bce-42b5-48a1-a8d4-b2bb2f28eabc"
                },
                "GeoRestrictions": {
                    "RestrictionType": "whitelist",
                    "Locations": [
                        "AL"
                    ]
                }
            },
            "CreatedTime": "2025-05-06T15:42:28.542000+00:00",
            "LastModifiedTime": "2025-05-06T16:14:08.710000+00:00",
            "ETag": "E1F83G8C2ARABC",
            "Enabled": true,
            "Status": "Deployed"
        }
    ]
}
```

자세한 내용은 Amazon CloudFront 개발자 안내서의 Distribution tenant customizations를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListDistributionTenantsByCustomization 섹션을 참조하세요.

`list-distribution-tenants`

다음 코드 예시는 `list-distribution-tenants`의 사용 방법을 보여 줍니다.

AWS CLI

CloudFront 배포 테넌트를 나열하려면

다음 `list-distribution-tenants` 예제에서는 연결된 연결 그룹별로 AWS 계정의 CloudFront 배포 테넌트 3개를 나열합니다.

```
aws cloudfront list-distribution-tenants \
    --association-filter ConnectionGroupId=cg_2whCJoXMYCjHcxaLGrkllvyABC \
    --max-items 3
```

출력:


```
{
    "DistributionTenantList": [
        {
            "Id": "dt_2yMvQgam3QkJo2z54FDl91dk1AB",
            "DistributionId": "E1XNX8R2GOAABC",
            "Name": "new-tenant-customizations",
            "Arn": "arn:aws:cloudfront::123456789012:distribution-tenant/
dt_2yMvQgam3QkJo2z54FDl91dk1AB",
            "Domains": [
                {
                    "Domain": "example.com",
                    "Status": "active"
                }
            ],
            "ConnectionGroupId": "cg_2whCJoXMYCjHcxaLGrkllvyABC",
            "Customizations": {
                "WebAcl": {
                    "Action": "disable"
                },
                "GeoRestrictions": {
                    "RestrictionType": "whitelist",
                    "Locations": [
                        "DE"
                    ]
                }
            },
            "CreatedTime": "2025-06-11T15:54:02.142000+00:00",
            "LastModifiedTime": "2025-06-11T15:54:02.142000+00:00",
            "ETag": "E23ZP02F085ABC",
            "Enabled": false,
            "Status": "Deployed"
        },
        {
            "Id": "dt_2yMuV7NJuBcAB0cwwxMCBZQ1AB",
            "DistributionId": "E1XNX8R2GOAABC",
            "Name": "new-tenant",
```

```
            "Arn": "arn:aws:cloudfront::123456789012:distribution-tenant/
dt_2yMuV7NJuBcAB0cwwxMCBZQ1AB",
            "Domains": [
                {
                    "Domain": "1.example.com",
                    "Status": "active"
                }
            ],
            "ConnectionGroupId": "cg_2whCJoXMYCjHcxaLGrkllvyABC",
            "Customizations": {
                "GeoRestrictions": {
                    "RestrictionType": "whitelist",
                    "Locations": [
                        "DE"
                    ]
                }
            },
            "CreatedTime": "2025-06-11T15:46:23.466000+00:00",
            "LastModifiedTime": "2025-06-11T15:46:23.466000+00:00",
            "ETag": "E23ZP02F085ABC",
            "Enabled": false,
            "Status": "Deployed"
        },
        {
            "Id": "dt_2xVInRKCfUzQHgxosDs9hiLk1AB",
            "DistributionId": "E1XNX8R2GOAABC",
            "Name": "new-tenant-2",
            "Arn": "arn:aws:cloudfront::123456789012:distribution-tenant/
dt_2xVInRKCfUzQHgxosDs9hiLk1AB",
            "Domains": [
                {
                    "Domain": "2.example.com",
                    "Status": "active"
                }
            ],
            "ConnectionGroupId": "cg_2whCJoXMYCjHcxaLGrkllvyABC",
            "CreatedTime": "2025-05-23T16:16:20.871000+00:00",
            "LastModifiedTime": "2025-05-23T16:16:20.871000+00:00",
            "ETag": "E23ZP02F085ABC",
            "Enabled": false,
            "Status": "Deployed"
        }
    ],
    "NextToken": "eyJNYXJrZXIiOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAzfQ=="
```

```
}
```

자세한 내용은 Amazon CloudFront 개발자 안내서의 Understand how multi-tenant distributions work를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListDistributionTenants 섹션을 참조하세요.

list-distributions-by-connection-mode

다음 코드 예시는 `list-distributions-by-connection-mode`의 사용 방법을 보여 줍니다.

AWS CLI

연결 모드별로 CloudFront 배포를 나열하려면

다음 `list-distributions-by-connection-mode` 예제에서는 지정된 연결 모드의 CloudFront 배포를 나열합니다.

```
aws cloudfront list-distributions-by-connection-mode \
    --connection-mode tenant-only
```

출력:

```
{
    "DistributionList": {
        "Items": [
            {
                "Id": "E1XNX8R2GOAABC",
                "ARN": "arn:aws:cloudfront::123456789012:distribution/
E1XNX8R2GOAABC",
                "ETag": "EPT4JPJQDY1ABC",
                "Status": "Deployed",
                "LastModifiedTime": "2025-05-23T16:16:15.691000+00:00",
                "DomainName": "-",
                "Aliases": {
                    "Quantity": 0
                },
                "Origins": {
                    "Quantity": 1,
                    "Items": [
                        {
                            "Id": "example-cfn-simple-distribution123",
                            "DomainName": "example.com",
```

```
                            "OriginPath": "",
                            "CustomHeaders": {
                                "Quantity": 0
                            },
                            "S3OriginConfig": {
                                "OriginAccessIdentity": ""
                            },
                            "ConnectionAttempts": 3,
                            "ConnectionTimeout": 10,
                            "OriginShield": {
                                "Enabled": false
                            },
                            "OriginAccessControlId": "E2CJRMB5LKEABC"
                        }
                    ]
                },
                "OriginGroups": {
                    "Quantity": 0
                },
                "DefaultCacheBehavior": {
                    "TargetOriginId": "example-cfn-simple-distribution123",
                    "TrustedKeyGroups": {
                        "Enabled": false,
                        "Quantity": 0
                    },
                    "ViewerProtocolPolicy": "allow-all",
                    "AllowedMethods": {
                        "Quantity": 2,
                        "Items": [
                            "HEAD",
                            "GET"
                        ],
                        "CachedMethods": {
                            "Quantity": 2,
                            "Items": [
                                "HEAD",
                                "GET"
                            ]
                        }
                    },
                    "Compress": true,
                    "LambdaFunctionAssociations": {
                        "Quantity": 0
                    },
```

```
                    "FunctionAssociations": {
                        "Quantity": 0
                    },
                    "FieldLevelEncryptionId": "",
                    "CachePolicyId": "658327ea-f89d-4fab-a63d-7e88639e5abc",
                    "GrpcConfig": {
                        "Enabled": false
                    }
                },
                "CacheBehaviors": {
                    "Quantity": 0
                },
                "CustomErrorResponses": {
                    "Quantity": 0
                },
                "Comment": "",
                "PriceClass": "PriceClass_All",
                "Enabled": true,
                "ViewerCertificate": {
                    "CloudFrontDefaultCertificate": false,
                    "ACMCertificateArn": "arn:aws:acm:us-
east-1:123456789012:certificate/ec53f564-ea5a-4e4a-a0a2-e3c989449abc",
                    "SSLSupportMethod": "sni-only",
                    "MinimumProtocolVersion": "TLSv1.2_2021",
                    "Certificate": "arn:aws:acm:us-east-1:123456789012:certificate/
ec53f564-ea5a-4e4a-a0a2-e3c989449abc",
                    "CertificateSource": "acm"
                },
                "Restrictions": {
                    "GeoRestriction": {
                        "RestrictionType": "none",
                        "Quantity": 0
                    }
                },
                "WebACLId": "arn:aws:wafv2:us-east-1:123456789012:global/webacl/web-
global-example/626900da-5f64-418b-ba9b-743f3746cabc",
                "HttpVersion": "http2",
                "IsIPV6Enabled": false,
                "Staging": false,
                "ConnectionMode": "tenant-only"
            }
        ]
    }
```

```
}
```

자세한 내용은 Amazon CloudFront 개발자 안내서의 [Create custom connection group (optional)]을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ListDistributionsByConnectionMode] 섹션을 참조하세요.

list-distributions

다음 코드 예시는 `list-distributions`의 사용 방법을 보여 줍니다.

AWS CLI

CloudFront 배포를 나열하려면

다음 예제에서는 AWS 계정의 CloudFront 배포 목록을 가져옵니다.

```
aws cloudfront list-distributions
```

출력:

```
{
    "DistributionList": {
        "Items": [
            {
                "Id": "E23YS8OEXAMPLE",
                "ARN": "arn:aws:cloudfront::123456789012:distribution/
E23YS8OEXAMPLE",
                "Status": "Deployed",
                "LastModifiedTime": "2024-08-05T18:23:40.375000+00:00",
                "DomainName": "abcdefgh12ijk.cloudfront.net",
                "Aliases": {
                    "Quantity": 0
                },
                "Origins": {
                    "Quantity": 1,
                    "Items": [
                        {
                            "Id": "amzn-s3-demo-bucket.s3.us-east-1.amazonaws.com",
                            "DomainName": "amzn-s3-demo-bucket.s3.us-
east-1.amazonaws.com",
                            "OriginPath": "",
                            "CustomHeaders": {
```

```
                                "Quantity": 0
                            },
                            "S3OriginConfig": {
                                "OriginAccessIdentity": ""
                            },
                            "ConnectionAttempts": 3,
                            "ConnectionTimeout": 10,
                            "OriginShield": {
                                "Enabled": false
                            },
                            "OriginAccessControlId": "EIAP8PEXAMPLE"
                        }
                    ]
                },
                "OriginGroups": {
                    "Quantity": 0
                },
                "DefaultCacheBehavior": {
                    "TargetOriginId": "amzn-s3-demo-bucket.s3.us-
east-1.amazonaws.com",
                    "TrustedSigners": {
                        "Enabled": false,
                        "Quantity": 0
                    },
                    "TrustedKeyGroups": {
                        "Enabled": false,
                        "Quantity": 0
                    },
                    "ViewerProtocolPolicy": "allow-all",
                    "AllowedMethods": {
                        "Quantity": 2,
                        "Items": [
                            "HEAD",
                            "GET"
                        ],
                        "CachedMethods": {
                            "Quantity": 2,
                            "Items": [
                                "HEAD",
                                "GET"
                            ]
                        }
                    },
                    "SmoothStreaming": false,
```

```
                        "Compress": true,
                        "LambdaFunctionAssociations": {
                            "Quantity": 0
                        },
                        "FunctionAssociations": {
                            "Quantity": 0
                        },
                        "FieldLevelEncryptionId": "",
                        "CachePolicyId": "658327ea-f89d-4fab-a63d-7e886EXAMPLE"
                    },
                    "CacheBehaviors": {
                        "Quantity": 0
                    },
                    "CustomErrorResponses": {
                        "Quantity": 0
                    },
                    "Comment": "",
                    "PriceClass": "PriceClass_All",
                    "Enabled": true,
                    "ViewerCertificate": {
                        "CloudFrontDefaultCertificate": true,
                        "SSLSupportMethod": "vip",
                        "MinimumProtocolVersion": "TLSv1",
                        "CertificateSource": "cloudfront"
                    },
                    "Restrictions": {
                        "GeoRestriction": {
                            "RestrictionType": "none",
                            "Quantity": 0
                        }
                    },
                    "WebACLId": "",
                    "HttpVersion": "HTTP2",
                    "IsIPV6Enabled": true,
                    "Staging": false
                }
            ]
        }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListDistributions 섹션을 참조하세요.

list-domain-conflicts

다음 코드 예시는 `list-domain-conflicts`의 사용 방법을 보여 줍니다.

AWS CLI

CloudFront 배포에 대한 도메인 충돌을 나열하려면

다음 `list-domain-conflicts` 예제에서는 CloudFront 배포에 대한 도메인 충돌을 나열합니다.

```
aws cloudfront list-domain-conflicts \
    --domain example.com \
    --domain-control-validation-
resource "DistributionTenantId=dt_2x9GhoK0TZRsohWzv1b9It8J1AB"
```

출력:

```
{
    "DomainConflicts": [
        {
            "Domain": "example.com",
            "ResourceType": "distribution-tenant",
            "ResourceId": "***************ohWzv1b9It8J1AB",
            "AccountId": "123456789012"
        }
    ]
}
```

자세한 내용은 Amazon CloudFront 개발자 안내서의 Move an alternate domain name to a different distribution을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListDomainConflicts 섹션을 참조하세요.

list-field-level-encryption-configs

다음 코드 예시는 `list-field-level-encryption-configs`의 사용 방법을 보여 줍니다.

AWS CLI

CloudFront 필드 수준 암호화 구성을 나열하는 방법

다음 예시에서는 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"
                            }
                        ]
                    }
                }
            }
        ]
    }
}
```


- API 세부 정보는 AWS CLI 명령 참조의 ListFieldLevelEncryptionConfigs 섹션을 참조하세요.

`list-field-level-encryption-profiles`

다음 코드 예시에서는 `list-field-level-encryption-profiles` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

CloudFront 필드 수준 암호화 프로파일을 나열하는 방법

다음 예시에서는 AWS 계정의 CloudFront 필드 수준 암호화 구성을 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"
            }
        ]
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListFieldLevelEncryptionProfiles 섹션을 참조하세요.

list-invalidations-for-distribution-tenant

다음 코드 예시는 `list-invalidations-for-distribution-tenant`의 사용 방법을 보여 줍니다.

AWS CLI

CloudFront 배포 테넌트에 대한 무효화를 나열하려면

다음 `list-invalidations-for-distribution-tenant` 예제에서는 CloudFront 배포 테넌트에 대한 무효화를 나열합니다.

```
aws cloudfront list-invalidations-for-distribution-tenant \
    --id dt_2wjDZi3hD1ivOXf6rpZJOSNE1AB
```

출력:

```
{
    "InvalidationList": {
        "Items": [
            {
                "Id": "I4CU23QAPKMUDUU06F9OFGFABC",
                "CreateTime": "2025-05-06T15:46:12.824000+00:00",
                "Status": "Completed"
            }
        ]
    }
}
```

자세한 내용은 Amazon CloudFront 개발자 안내서의 Invalidate files to remove content를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListInvalidationsForDistributionTenant 섹션을 참조하세요.

list-invalidations

다음 코드 예시는 `list-invalidations`의 사용 방법을 보여 줍니다.

AWS CLI

CloudFront 무효화를 나열하는 방법

다음 예시에서는 ID가 EDFDVBD6EXAMPLE인 CloudFront 배포에 대한 무효화 목록을 가져옵니다.

```
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
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListInvalidations 섹션을 참조하세요.

`list-public-keys`

다음 코드 예시에서는 `list-public-keys` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

CloudFront 퍼블릭 키를 나열하는 방법

다음 예시에서는 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/1U0WcmZX2u70dYcSIzB1ofZtcn4cJ
\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"
            }
        ]
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [ListPublicKeys](https://docs.aws.amazon.com/goto/aws-cli/cloudfront-2020-05-31/ListPublicKeys) 섹션을 참조하세요.

`list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

CloudFront 배포의 태그 나열

다음 예시에서는 CloudFront 배포 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"
            }
        ]
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource 섹션을 참조하세요.

sign

다음 코드 예시에서는 `sign` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

CloudFront URL에 서명하는 방법

다음 예시는 CloudFront URL에 서명합니다. URL에 서명하는 방법 키 페어 ID(AWS관리 콘솔에서 액세스 키 ID라고 함)와 신뢰할 수 있는 서명자의 CloudFront 키 페어의 프라이빗 키가 필요합니다. 서명된 URL에 대한 자세한 내용은 Amazon CloudFront 개발자 안내서의 서명된 URL 및 서명된 쿠키로 프라이빗 콘텐츠 제공을 참조하세요.

```
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
```

- API 세부 정보는 AWS CLI 명령 참조의 Sign을 참조하세요.

tag-resource

다음 코드 예시에서는 `tag-resource` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

CloudFront 배포에 태그 지정

다음 `tag-resource` 예시에서는 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"}]'
```

명령줄 인수를 사용하는 대신 다음 예시와 같이 JSON 파일로 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.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

untag-resource

다음 코드 예시에서는 untag-resource 코드를 사용하는 방법을 보여줍니다.

AWS CLI

CloudFront 배포에서 태그를 제거하는 방법

다음 예시에서는 명령줄 인수를 사용하여 CloudFront 배포에서 두 태그를 제거합니다.

```
aws cloudfront untag-resource \
    --resource arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE \
    --tag-keys Items=Name,Project
```

명령줄 인수를 사용하는 대신 다음 예시와 같이 JSON 파일로 태그 키를 제공할 수 있습니다.

```
aws cloudfront untag-resource \
    --resource arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE \
    --tag-keys file://tag-keys.json
```

tag-keys.json 파일은 다음을 포함한 현재 문서의 JSON 문서입니다.

```
{
    "Items": [
        "Name",
        "Project"
    ]
}
```

이 명령이 제대로 실행되면 출력이 표시되지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource 섹션을 참조하세요.

update-cloud-front-origin-access-identity

다음 코드 예시에서는 update-cloud-front-origin-access-identity 코드를 사용하는 방법을 보여줍니다.

AWS CLI

CloudFront 원본 액세스 ID 업데이트

다음 예시에서는 ID가 E74FTE3AEXAMPLE인 오리진 액세스 ID(OAI)를 업데이트하기 위한 요청입니다. 업데이트할 수 있는 유일한 필드는 OAI의 Comment입니다.

OAI를 업데이트하려면 OAI의 ID와 ETag가 있어야 합니다. OAI ID는 create-cloud-front-origin-access-identity 및 list-cloud-front-origin-access-identities 명령의 출력에 반환됩니다. ETag를 가져오려면 get-cloud-front-origin-access-identity 또는 get-cloud-front-origin-access-identity-config 명령을 사용합니다. `--if-match` 옵션을 사용하여 OAI의 ETag를 제공합니다.

```
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 예시와 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
```

`OAI-config.json` 파일은 다음을 포함한 현재 디렉터리의 JSON 문서입니다.

```
{
    "CallerReference": "cli-example",
    "Comment": "Example OAI Updated"
}
```

명령줄 인수로 OAI 구성을 제공하든 JSON 파일로 제공하든 출력은 동일합니다.

```
{
    "ETag": "E9LHASXEXAMPLE",
    "CloudFrontOriginAccessIdentity": {
        "Id": "E74FTE3AEXAMPLE",
        "S3CanonicalUserId":
 "cd13868f797c227fbea2830611a26fe0a21ba1b826ab4bed9b7771c9aEXAMPLE",
        "CloudFrontOriginAccessIdentityConfig": {
            "CallerReference": "cli-example",
            "Comment": "Example OAI Updated"
        }
```

```
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateCloudFrontOriginAccessIdentity 섹션을 참조하세요.

update-connection-group

다음 코드 예시는 update-connection-group의 사용 방법을 보여 줍니다.

AWS CLI

CloudFront 연결 그룹을 업데이트하려면

다음 update-connection-group 예제에서는 CloudFront 연결 그룹을 비활성화하고 IPv6를 비활성화합니다.

```
aws cloudfront update-connection-group \
    --id cg_2yHsDkcPKeUlVkk3aEgLKcjABC \
    --no-ipv6-enabled \
    --no-enabled \
    --if-match E3UN6WX5RRO2ABC
```

출력:

```
{
    "ETag": "E1F83G8C2ARABC",
    "ConnectionGroup": {
        "Id": "cg_2yHsDkcPKeUlVkk3aEgLKcjABC",
        "Name": "cg-example",
        "Arn": "arn:aws:cloudfront::123456789012:connection-group/
cg_2yHsDkcPKeUlVkk3aEgLKcjABC",
        "CreatedTime": "2025-06-09T20:58:35.481000+00:00",
        "LastModifiedTime": "2025-06-11T16:25:54.280000+00:00",
        "Ipv6Enabled": false,
        "RoutingEndpoint": "du9xp1elo1abc.cloudfront.net",
        "Status": "InProgress",
        "Enabled": false,
        "IsDefault": false
    }
}
```

자세한 내용은 Amazon CloudFront 개발자 안내서의 Create custom connection group (optional)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateConnectionGroup 섹션을 참조하세요.

update-distribution-tenant

다음 코드 예시는 `update-distribution-tenant`의 사용 방법을 보여 줍니다.

AWS CLI

CloudFront 배포 테넌트를 업데이트하려면

다음 `update-distribution-tenant` 예제에서는 CloudFront 배포 테넌트를 새 파라미터 값으로 업데이트하고 지역 제한에 국가를 추가합니다.

```
aws cloudfront update-distribution-tenant \
    --cli-input-json file://update-tenant.json
```

`update-tenant.json`의 콘텐츠:

```
{
    "Id": "dt_2yMvQgam3QkJo2z54FDl91dk1AB",
    "IfMatch": "E1F83G8C2ARABC",
    "Parameters": [
        {
            "Name": "testParam",
            "Value": "newParameterValue"
        }
    ],
    "Customizations": {
        "WebAcl": {
            "Action": "disable"
        },
        "GeoRestrictions": {
            "RestrictionType": "whitelist",
            "Locations": [
                "DE",
                "GB",
                "ES"
            ]
        }
```

```
    }
}
```

출력:

```
{
    "ETag": "E1PA6795UKMABC",
    "DistributionTenant": {
        "Id": "dt_2yMvQgam3QkJo2z54FDl91dk1AB",
        "DistributionId": "E1XNX8R2GOAABC",
        "Name": "new-tenant-customizations",
        "Arn": "arn:aws:cloudfront::123456789012:distribution-tenant/
dt_2yMvQgam3QkJo2z54FDl91dk1AB",
        "Domains": [
            {
                "Domain": "example.com",
                "Status": "active"
            }
        ],
        "Customizations": {
            "WebAcl": {
                "Action": "disable"
            },
            "GeoRestrictions": {
                "RestrictionType": "whitelist",
                "Locations": [
                    "DE",
                    "ES",
                    "GB"
                ]
            }
        },
        "Parameters": [
            {
                "Name": "testParam",
                "Value": "newParameterValue"
            }
        ],
        "ConnectionGroupId": "cg_2whCJoXMYCjHcxaLGrkllvyABC",
        "CreatedTime": "2025-06-11T15:54:02.142000+00:00",
        "LastModifiedTime": "2025-06-11T16:42:45.531000+00:00",
        "Enabled": false,
        "Status": "InProgress"
```

```
    }
}
```

자세한 내용은 Amazon CloudFront 개발자 안내서의 Distribution tenant customizations를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateDistributionTenant 섹션을 참조하세요.

update-distribution

다음 코드 예시는 update-distribution의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: CloudFront 배포의 기본 루트 객체를 업데이트하는 방법

다음 예제에서는 EDFDVBD6EXAMPLE ID를 가진 CloudFront 배포에서 기본 루트 객체를 index.html로 업데이트합니다.

```
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
        }
    }
}
```

예제 2: CloudFront 배포를 업데이트하는 방법

다음 예제에서는 `dist-config-disable.json`이라는 JSON 파일로 배포 구성을 제공하여 ID `EMLARXS9EXAMPLE`을 사용한 CloudFront 배포를 비활성화합니다. 배포를 업데이트하려면 배포의 `ETag`를 제공하는 `--if-match` 옵션을 사용해야 합니다. `ETag`를 가져오려면 get-distribution 또는 get-distribution-config 명령을 사용하세요. JSON 파일에서 `Enabled` 필드가 `false`로 설정되어 있습니다.

다음 예제를 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
```

`dist-config-disable.json`의 콘텐츠:

```
{
    "CallerReference": "cli-1574382155-496510",
```

```
    "Aliases": {
        "Quantity": 0
    },
    "DefaultRootObject": "index.html",
    "Origins": {
        "Quantity": 1,
        "Items": [
            {
                "Id": "amzn-s3-demo-bucket.s3.amazonaws.com-1574382155-273939",
                "DomainName": "amzn-s3-demo-bucket.s3.amazonaws.com",
                "OriginPath": "",
                "CustomHeaders": {
                    "Quantity": 0
                },
                "S3OriginConfig": {
                    "OriginAccessIdentity": ""
                }
            }
        ]
    },
    "OriginGroups": {
        "Quantity": 0
    },
    "DefaultCacheBehavior": {
        "TargetOriginId": "amzn-s3-demo-bucket.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": "amzn-s3-demo-
bucket.s3.amazonaws.com-1574382155-273939",
                        "DomainName": "amzn-s3-demo-bucket.s3.amazonaws.com",
                        "OriginPath": "",
                        "CustomHeaders": {
                            "Quantity": 0
                        },
                        "S3OriginConfig": {
                            "OriginAccessIdentity": ""
```

```
                        }
                    }
                ]
            },
            "OriginGroups": {
                "Quantity": 0
            },
            "DefaultCacheBehavior": {
                "TargetOriginId": "amzn-s3-demo-
bucket.s3.amazonaws.com-1574382155-273939",
                "ForwardedValues": {
                    "QueryString": false,
                    "Cookies": {
                        "Forward": "none"
                    },
                    "Headers": {
                        "Quantity": 0
                    },
                    "QueryStringCacheKeys": {
                        "Quantity": 0
                    }
                },
                "TrustedSigners": {
                    "Enabled": false,
                    "Quantity": 0
                },
                "ViewerProtocolPolicy": "allow-all",
                "MinTTL": 0,
                "AllowedMethods": {
                    "Quantity": 2,
                    "Items": [
                        "HEAD",
                        "GET"
                    ],
                    "CachedMethods": {
                        "Quantity": 2,
                        "Items": [
                            "HEAD",
                            "GET"
                        ]
                    }
                },
                "SmoothStreaming": false,
                "DefaultTTL": 86400,
```

```
                "MaxTTL": 31536000,
                "Compress": false,
                "LambdaFunctionAssociations": {
                    "Quantity": 0
                },
                "FieldLevelEncryptionId": ""
            },
            "CacheBehaviors": {
                "Quantity": 0
            },
            "CustomErrorResponses": {
                "Quantity": 0
            },
            "Comment": "",
            "Logging": {
                "Enabled": false,
                "IncludeCookies": false,
                "Bucket": "",
                "Prefix": ""
            },
            "PriceClass": "PriceClass_All",
            "Enabled": false,
            "ViewerCertificate": {
                "CloudFrontDefaultCertificate": true,
                "MinimumProtocolVersion": "TLSv1",
                "CertificateSource": "cloudfront"
            },
            "Restrictions": {
                "GeoRestriction": {
                    "RestrictionType": "none",
                    "Quantity": 0
                }
            },
            "WebACLId": "",
            "HttpVersion": "http2",
            "IsIPV6Enabled": true
        }
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateDistribution을 참조하세요.

`update-domain-association`

다음 코드 예시는 `update-domain-association`의 사용 방법을 보여 줍니다.

AWS CLI

도메인 연결을 업데이트하려면

다음 `update-domain-association` 예제에서는 `ETag`가 `E23ZP02F085ABC`인 배포 테넌트에 대한 도메인 연결을 업데이트합니다.

```
aws cloudfront update-domain-association \
    --domain example.com \
    --target-resource DistributionTenantId=dt_2x9GhoK0TZRsohWzv1b9It8J1AB \
    --if-match E23ZP02F085ABC
```

출력:

```
{
    "ETag": "ETVPDKIKX0ABC",
    "Domain": "example.com",
    "ResourceId": "dt_2x9GhoK0TZRsohWzv1b9It8J1AB"
}
```

자세한 내용은 Amazon CloudFront 개발자 안내서의 Move an alternate domain name to a different distribution을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateDomainAssociation 섹션을 참조하세요.

`update-field-level-encryption-config`

다음 코드 예시는 `update-field-level-encryption-config`의 사용 방법을 보여 줍니다.

AWS CLI

CloudFront 필드 수준 암호화 구성을 업데이트하는 방법

다음 예시에서는 JSON 파일에 파라미터를 제공하여 UD가 `C3KM2WVD605UAY`인 필드 수준 암호화 구성을 업데이트합니다.

필드 수준 암호화 구성을 업데이트하려면 구성의 ID 및 `ETag`가 있어야 합니다. ID는 create-field-level-encryption-config 및 list-field-level-encryption-configs 명령의 출력으로 반환됩니다. `ETag`를

가져오려면 get-field-level-encryption 또는 get-field-level-encryption-config 명령을 사용합니다. `--if-match` 옵션을 사용하여 구성의 ETag를 제공합니다.

```
aws cloudfront update-field-level-encryption-config \
    --id C3KM2WVD605UAY \
    --if-match E2P4Z4VU7TY5SG \
    --field-level-encryption-config file://fle-config.json
```

`fle-config.json` 파일은 다음을 포함한 현재 디렉터리의 JSON 문서입니다.

```
{
    "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"
                        }
                    ]
                }
            }
        }
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateFieldLevelEncryptionConfig 섹션을 참조하세요.

`update-field-level-encryption-profile`

다음 코드 예시에서는 `update-field-level-encryption-profile` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

CloudFront 필드 수준 암호화 프로파일을 업데이트하는 방법

다음 예시에서는 ID가 `PPK0UOSIF5WSV`인 필드 수준 암호화 프로파일을 업데이트합니다. 이 예시에서는 JSON 파일에 파라미터를 제공하여 프로파일의 `Name`과 `Comment`를 업데이트하고 두 번째 항목을 추가합니다.

필드 수준 암호화 프로파일을 업데이트하려면 프로파일의 ID와 `ETag`가 있어야 합니다. ID는 create-field-level-encryption-profile 및 list-field-level-encryption-profiles 명령의 출력으로 반환됩

니다. ETag를 가져오려면 get-field-level-encryption-profile 또는 get-field-level-encryption-profile-config 명령을 사용합니다. `--if-match` 옵션을 사용하여 프로파일의 ETag를 제공합니다.

```
aws cloudfront update-field-level-encryption-profile \
    --id PPK0UOSIF5WSV \
    --if-match E1QQG65FS2L2GC \
    --field-level-encryption-profile-config file://fle-profile-config.json
```

`fle-profile-config.json` 파일은 다음을 포함한 현재 디렉터리의 JSON 문서입니다.

```
{
    "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"
                                ]
                            }
                        }
                    ]
                }
            }
        }
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateFieldLevelEncryptionProfile 섹션을 참조하세요.

verify-dns-configuration

다음 코드 예시는 verify-dns-configuration의 사용 방법을 보여 줍니다.

AWS CLI

도메인의 DNS 구성을 확인하려면

다음 verify-dns-configuration 예제에서는 도메인의 DNS 구성을 확인합니다.

```
aws cloudfront verify-dns-configuration \
    --domain example.com \
    --identifier dt_2x9GhoK0TZRsohWzv1b9It8J1AB
```

출력:

```
{
    "DnsConfigurationList": [
        {
```

```
            "Domain": "example.com",
            "Status": "valid-configuration"
        }
    ]
}
```

자세한 내용은 Amazon CloudFront 개발자 안내서의 Move an alternate domain name to a different distribution을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 VerifyDnsConfiguration 섹션을 참조하세요.

AWS CLI를 사용한 Amazon CloudSearch 예제

다음 코드 예제에서는 Amazon CloudSearch에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여 줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

upload-documents

다음 코드 예시는 `upload-documents`의 사용 방법을 보여 줍니다.

AWS CLI

다음 `upload-documents` 명령은 JSON 형식의 문서 배치를 Amazon CloudSearch 도메인에 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 UploadDocuments를 참조하세요.

CloudTrail examples using AWS CLI

다음 코드 예제에서는 CloudTrail에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여 줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

작업

add-tags

다음 코드 예시는 add-tags의 사용 방법을 보여 줍니다.

AWS CLI

추적에 태그를 추가하는 방법

다음 add-tags 명령은 Trail1에 대한 태그를 추가합니다.

```
aws cloudtrail add-tags --resource-id arn:aws:cloudtrail:us-
east-1:123456789012:trail/Trail1 --tags-
list Key=name,Value=Alice Key=location,Value=us
```

- API 세부 정보는 AWS CLI 명령 참조의 AddTags 섹션을 참조하세요.

`create-subscription`

다음 코드 예시는 `create-subscription`의 사용 방법을 보여 줍니다.

AWS CLI

추적에 대한 AWS 리소스를 생성하고 구성하는 방법

다음 `create-subscription` 명령은 `Trail1`에 대한 새 S3 버킷 및 SNS 주제를 생성합니다.

```
aws cloudtrail create-subscription \
    --name Trail1 \
    --s3-new-bucket amzn-s3-demo-bucket \
    --sns-new-topic my-topic
```

출력:

```
Setting up new S3 bucket amzn-s3-demo-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": "amzn-s3-demo-bucket",
                "SnsTopicName": "my-topic",
                "HomeRegion": "us-east-1"
            }
        ],
        "ResponseMetadata": {
        "HTTPStatusCode": 200,
        "RequestId": "f39e51f6-c615-11e5-85bd-d35ca21ee3e2"
        }
    }
Starting CloudTrail service...
Logs will be delivered to my-bucket
```

- API 세부 정보는 AWS CLI 명령 참조의 [CreateSubscription](#)을 참조합니다.

`create-trail`

다음 코드 예시는 `create-trail`의 사용 방법을 보여 줍니다.

AWS CLI

추적을 생성하려면

다음 `create-trail` 예제에서는 이름이 `Trail1`인 다중 리전 추적을 생성하고 S3 버킷을 지정합니다.

```
aws cloudtrail create-trail \
    --name Trail1 \
    --s3-bucket-name amzn-s3-demo-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": "amzn-s3-demo-bucket"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [CreateTrail](#)을 참조하세요.

`delete-trail`

다음 코드 예시는 `delete-trail`의 사용 방법을 보여 줍니다.

AWS CLI

추적을 삭제하는 방법

다음 `delete-trail` 명령은 이름인 `Trail1`인 추적을 삭제합니다.

```
aws cloudtrail delete-trail --name Trail1
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteTrail을 참조하세요.

describe-trails

다음 코드 예시는 describe-trails의 사용 방법을 보여 줍니다.

AWS CLI

추적을 설명하는 방법

다음 describe-trails 예제에서는 Trail1 및 Trail2의 설정을 반환합니다.

```
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": "amzn-s3-demo-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": "amzn-s3-demo-bucket2",
            "KmsKeyId": "arn:aws:kms:us-
east-1:123456789012:key/4c5ae5ac-3c13-421e-8335-c7868ef6a769",
            "HomeRegion": "us-east-1"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeTrail을 참조하세요.

get-event-selectors

다음 코드 예시는 get-event-selectors의 사용 방법을 보여 줍니다.

AWS CLI

추적의 이벤트 선택기 설정을 보는 방법

다음 get-event-selectors 명령은 Trail1에 대한 설정을 반환합니다.

```
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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetEventSelectors를 참조하세요.

get-trail-status

다음 코드 예시는 get-trail-status의 사용 방법을 보여 줍니다.

AWS CLI

추적 상태를 가져오는 방법

다음 `get-trail-status` 명령은 `Trail1`에 대한 전송 및 로깅 세부 정보를 반환합니다.

```
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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetTrailStatus를 참조하세요.

`list-public-keys`

다음 코드 예시는 `list-public-keys`의 사용 방법을 보여 줍니다.

AWS CLI

추적의 모든 퍼블릭 키를 나열하는 방법

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"
     }
  ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [ListPublicKeys](#) 섹션을 참조하세요.

`list-tags`

다음 코드 예시는 `list-tags`의 사용 방법을 보여 줍니다.

AWS CLI

추적의 태그를 나열하는 방법

다음 `list-tags` 명령은 `Trail1` 및 `Trail2`에 대한 태그를 나열합니다.

```
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"
                }
            ]
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListTags를 참조하세요.

lookup-events

다음 코드 예시는 lookup-events의 사용 방법을 보여 줍니다.

AWS CLI

추적에 대한 이벤트를 조회하는 방법

다음 lookup-events 명령은 속성 EventName별로 API 활동 이벤트를 검색합니다

```
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": []
      }
  ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [LookupEvents](https://docs.aws.amazon.com/goto/aws-cli/cloudtrail-2013-11-01/LookupEvents)를 참조하세요.

put-event-selectors

다음 코드 예시는 `put-event-selectors`의 사용 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 형식을 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 `put-event-selectors` 예제에서는 `myTrail`이라는 추적에 대한 고급 이벤트 선택기를 생성하여 모든 관리 이벤트를 기록하고, 하나의 S3 버킷을 제외한 모든 S3 PutObject 및 DeleteObject API 직접 호출을 기록하고, `myFunction`이라는 Lambda 함수에 대한 데이터 API 직접 호출을 기록하고, `myTopic`이라는 SNS 주제에 대한 게시 API 직접 호출을 기록합니다.

```
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:::amzn-s3-demo-bucket/"] }]},{"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:::amzn-s3-demo-bucket/"
                ]
            }
        ]
    },
    {
        "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 설명서에서 고급 이벤트 선택기를 사용하여 이벤트 로깅을 참조하세요.

예제 2: 모든 관리 이벤트 및 데이터 이벤트를 기록하도록 트레일에 대한 이벤트 선택기 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 함수 및 DynnamoDB 테이블에 대한 관리 이벤트 및 데이터 이벤트를 로깅할 수 있습니다. 다른 리소스 형식의 데이터 이벤트를 기록하려면 고급 이벤트 선택기를 사용해야 합니다.

다음 put-event-selectors 예제에서는 모든 관리 이벤트, 두 개의 Amazon S3 버킷/접두사 조합에 대한 데이터 이벤트, hello-world-python-function이라는 단일 TrailName Lambda 함수에 대한 데이터 이벤트를 포함하도록 AWS이라는 추적에 대한 이벤트 선택기를 생성합니다.

```
aws cloudtrail put-event-selectors \
    --trail-name TrailName \
    --event-selectors '[{"ReadWriteType": "All","IncludeManagementEvents":
 true,"DataResources": [{"Type":"AWS::S3::Object", "Values": ["arn:aws:s3:::amzn-
s3-demo-bucket/prefix","arn:aws:s3:::amzn-s3-demo-bucket2/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:::amzn-s3-demo-bucket/prefix",
                        "arn:aws:s3:::amzn-s3-demo-bucket2/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 설명서에서 기본 이벤트 선택기를 사용하여 이벤트 로깅을 참조하세요.

예제 3: 관리 이벤트, S3 객체의 모든 S3 데이터 이벤트 및 계정의 함수에 대한 모든 Lambda 데이터 이벤트를 기록하도록 트레일에 대한 이벤트 선택기 구성

다음 `put-event-selectors` 예제에서는 모든 관리 이벤트와 AWS 계정의 모든 Amazon S3 버킷 및 AWS Lambda 함수에 대한 모든 데이터 이벤트를 포함하는 `TrailName2`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 설명서에서 기본 이벤트 선택기를 사용하여 이벤트 로깅을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutEventSelectors를 참조하세요.

`remove-tags`

다음 코드 예시는 `remove-tags`의 사용 방법을 보여 줍니다.

AWS CLI

추적의 태그를 제거하는 방법

다음 `remove-tags` 명령은 지정된 `Trail1` 태그를 지정합니다.

```
aws cloudtrail remove-tags --resource-id arn:aws:cloudtrail:us-
east-1:123456789012:trail/Trail1 --tags-list Key=name Key=location
```

- API 세부 정보는 AWS CLI 명령 참조의 RemoveTags 섹션을 참조하세요.

`start-logging`

다음 코드 예시는 `start-logging`의 사용 방법을 보여 줍니다.

AWS CLI

추적에 대해 로깅을 시작하는 방법

다음 `start-logging` 명령을 `Trail1`에 대한 로깅을 켭니다.

```
aws cloudtrail start-logging --name Trail1
```

- API 세부 정보는 AWS CLI 명령 참조의 StartLogging을 참조하세요.

`stop-logging`

다음 코드 예시는 `stop-logging`의 사용 방법을 보여 줍니다.

AWS CLI

추적 로깅을 중지하는 방법

다음 `stop-logging` 명령은 `Trail1`에 대한 로깅을 끕니다.

```
aws cloudtrail stop-logging --name Trail1
```

- API 세부 정보는 AWS CLI 명령 참조의 StopLogging을 참조하세요.

`update-subscription`

다음 코드 예시는 `update-subscription`의 사용 방법을 보여 줍니다.

AWS CLI

추적에 대한 구성 설정을 업데이트하는 방법

다음 `update-subscription` 예제에서는 추적을 업데이트하여 새 S3 버킷 및 SNS 주제를 지정합니다.

```
aws cloudtrail update-subscription \
    --name Trail1 \
    --s3-new-bucket amzn-s3-demo-bucket \
    --sns-new-topic my-topic-new
```

출력:

```
Setting up new S3 bucket amzn-s3-demo-bucket...
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": "amzn-s3-demo-bucket",
            "SnsTopicName": "my-topic-new",
            "HomeRegion": "us-east-1"
        }
    ],
    "ResponseMetadata": {
        "HTTPStatusCode": 200,
        "RequestId": "31126f8a-c616-11e5-9cc6-2fd637936879"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [UpdateSubscription](#)을 참조하세요.

update-trail

다음 코드 예시는 update-trail의 사용 방법을 보여 줍니다.

AWS CLI

추적을 업데이트하는 방법

다음 update-trail 예제에서는 로그 전송에 기존 버킷을 사용하도록 추적을 업데이트합니다.

```
aws cloudtrail update-trail \
    --name Trail1 \
    --s3-bucket-name amzn-s3-demo-bucket
```

출력:

```
{
    "IncludeGlobalServiceEvents": true,
    "Name": "Trail1",
    "TrailARN": "arn:aws:cloudtrail:us-west-2:123456789012:trail/Trail1",
    "LogFileValidationEnabled": false,
    "IsMultiRegionTrail": true,
    "S3BucketName": "amzn-s3-demo-bucket"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateTrail을 참조하세요.

validate-logs

다음 코드 예시는 validate-logs의 사용 방법을 보여 줍니다.

AWS CLI

로그 파일을 검증하는 방법

다음 validate-logs 명령은 Trail1에 대한 로그의 유효성을 확인합니다.

```
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
```

- API 세부 정보는 AWS CLI 명령 참조의 ValidateLogs를 참조하세요.

AWS CLI를 사용하는 CloudWatch 예제

다음 코드 예제에서는 CloudWatch에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여 줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

delete-alarms

다음 코드 예시는 `delete-alarms`의 사용 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.
```

- API 세부 정보는 AWS CLI Command Reference의 DeleteAlarms를 참조하세요.

`delete-anomaly-detector`

다음 코드 예시는 `delete-anomaly-detector`의 사용 방법을 보여 줍니다.

AWS CLI

지정된 이상 탐지 모델을 삭제하려면 다음을 수행합니다.

다음 `delete-anomaly-detector` 예제에서는 지정된 계정에서 이상 탐지기 모델을 삭제합니다.

```
aws cloudwatch delete-anomaly-detector \
    --namespace AWS/Logs \
    --metric-name IncomingBytes \
    --stat SampleCount
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon CloudWatch 사용 설명서의 이상 탐지 모델 삭제 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteAnomalyDetector을 참조하세요.

`delete-dashboards`

다음 코드 예시는 `delete-dashboards`의 사용 방법을 보여 줍니다.

AWS CLI

지정된 대시보드를 삭제하려면 다음을 수행합니다.

다음 `delete-dashboards` 예제에서는 지정된 계정에서 `Dashboard-A` 및 `Dashboard-B`라는 대시보드 2개를 삭제합니다.

```
aws cloudwatch delete-dashboards \
    --dashboard-names Dashboard-A Dashboard-B
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon CloudWatch 사용 설명서의 Amazon CloudWatch 대시보드 사용 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteDashboards를 참조하세요.

`delete-insight-rules`

다음 코드 예시는 `delete-insight-rules`의 사용 방법을 보여 줍니다.

AWS CLI

지정된 Contributor Insights 규칙을 삭제하려면

다음 `delete-insight-rules` 예제에서는 지정된 계정에서 `Rule-A` 및 `Rule-B`라는 Contributor Insights 2개를 삭제합니다.

```
aws cloudwatch delete-insight-rules \
    --rule-names Rule-A Rule-B
```

출력:

```
{
    "Failures": []
}
```

자세한 내용을 알아보려면 Amazon CloudWatch 사용 설명서의 Contributor Insights를 사용하여 카디널리티가 높은 데이터 분석하기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteInsightRules를 참조하세요.

`delete-metric-stream`

다음 코드 예시는 `delete-metric-stream`의 사용 방법을 보여 줍니다.

AWS CLI

지정된 지표 스트림을 삭제하려면

다음 `delete-metric-stream` 예제에서는 지정된 계정에서 `QuickPartial-gSCKvO`라는 지표 스트림를 삭제합니다.

```
aws cloudwatch delete-metric-stream \
```

```
    --name QuickPartial-gSCKvO
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon CloudWatch 사용 설명서의 지표 스트림 사용 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteMetricStream를 참조하세요.

describe-alarm-history

다음 코드 예시는 `describe-alarm-history`의 사용 방법을 보여 줍니다.

AWS CLI

경보에 대한 기록을 검색하는 방법

다음 예제에서는 `describe-alarm-history` 명령을 사용하여 'myalarm'이라는 Amazon CloudWatch 경보에 대한 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI Command Reference의 DescribeAlarmHistory를 참조하세요.

describe-alarms-for-metric

다음 코드 예시는 `describe-alarms-for-metric`의 사용 방법을 보여 줍니다.

AWS CLI

지표와 관련된 경보에 대한 정보를 표시하는 방법

다음 예제에서는 `describe-alarms-for-metric` 명령을 사용하여 Amazon EC2 CPUUtilization 지표 및 ID i-0c986c72의 인스턴스와 관련된 모든 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI Command Reference의 DescribeAlarmsForMetric을 참조하세요.

describe-alarms

다음 코드 예시는 `describe-alarms`의 사용 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI Command Reference의 DescribeAlarms를 참조하세요.

describe-anomaly-detectors

다음 코드 예시는 describe-anomaly-detectors의 사용 방법을 보여 줍니다.

AWS CLI

이상 탐지 모델 목록을 검색하려면 다음을 수행합니다.

다음 describe-anomaly-detectors 예제에서는 지정된 계정의 AWS/Logs 네임스페이스와 연결된 이상 탐지기 모델에 대한 정보를 표시합니다.

```
aws cloudwatch describe-anomaly-detectors \
    --namespace AWS/Logs
```

출력:


```
{
    "AnomalyDetectors": [
        {
            "Namespace": "AWS/Logs",
            "MetricName": "IncomingBytes",
            "Dimensions": [],
            "Stat": "SampleCount",
            "Configuration": {
                "ExcludedTimeRanges": []
            },
            "StateValue": "TRAINED",
            "SingleMetricAnomalyDetector": {
                "AccountId": "123456789012",
                "Namespace": "AWS/Logs",
                "MetricName": "IncomingBytes",
                "Dimensions": [],
                "Stat": "SampleCount"
            }
        },
        {
            "Namespace": "AWS/Logs",
            "MetricName": "IncomingBytes",
            "Dimensions": [
                {
                    "Name": "LogGroupName",
                    "Value": "demo"
                }
            ],
            "Stat": "Average",
            "Configuration": {
                "ExcludedTimeRanges": []
            },
            "StateValue": "PENDING_TRAINING",
            "SingleMetricAnomalyDetector": {
                "AccountId": "123456789012",
```

```
                "Namespace": "AWS/Logs",
                "MetricName": "IncomingBytes",
                "Dimensions": [
                    {
                        "Name": "LogGroupName",
                        "Value": "demo"
                    }
                ],
                "Stat": "Average"
            }
        }
    ]
}
```

자세한 내용은 Amazon CloudWatch 사용 설명서의 CloudWatch 이상 탐지 사용 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeAnomalyDetectors를 참조하세요.

describe-insight-rules

다음 코드 예시는 `describe-insight-rules`의 사용 방법을 보여 줍니다.

AWS CLI

Contributor Insights 규칙 목록을 검색하려면

다음 `describe-insight-rules` 예제에서는 지정된 계정의 모든 Contributor Insight 규칙을 보여줍니다.

```
aws cloudwatch describe-insight-rules
```

출력:

```
{
    "InsightRules": [
        {
            "Name": "Rule-A",
            "State": "ENABLED",
            "Schema": "CloudWatchLogRule/1",
            "Definition": "{\n\t\"AggregateOn\": \"Count\",\n\t\"Contribution\":
 {\n\t\t\"Filters\": [],\n\t\t\"Keys\": [\n\t\t\t\"$.requestId\"\n\t\t]\n\t},\n
\t\"LogFormat\": \"JSON\",\n\t\"Schema\": {\n\t\t\"Name\": \"CloudWatchLogRule
```

```
\",\n\t\t\"Version\": 1\n\t},\n\t\"LogGroupARNs\": [\n\t\t\"arn:aws:logs:us-
east-1:123456789012:log-group:demo\"\n\t]\n}",
            "ManagedRule": false
        },
        {
            "Name": "Rule-B",
            "State": "ENABLED",
            "Schema": "CloudWatchLogRule/1",
            "Definition": "{\n\t\"AggregateOn\": \"Count\",\n\t\"Contribution\":
 {\n\t\t\"Filters\": [],\n\t\t\"Keys\": [\n\t\t\t\"$.requestId\"\n\t\t]\n\t},\n
\t\"LogFormat\": \"JSON\",\n\t\"Schema\": {\n\t\t\"Name\": \"CloudWatchLogRule
\",\n\t\t\"Version\": 1\n\t},\n\t\"LogGroupARNs\": [\n\t\t\"arn:aws:logs:us-
east-1:123456789012:log-group:demo-1\"\n\t]\n}",
            "ManagedRule": false
        }
    ]
}
```

자세한 내용을 알아보려면 Amazon CloudWatch 사용 설명서의 Contributor Insights를 사용하여 카디널리티가 높은 데이터 분석하기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeInsightRules를 참조하세요.

disable-alarm-actions

다음 코드 예시는 `disable-alarm-actions`의 사용 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.

- API 세부 정보는 AWS CLI Command Reference의 DisableAlarmActions를 참조하세요.

disable-insight-rules

다음 코드 예시는 `disable-insight-rules`의 사용 방법을 보여 줍니다.

AWS CLI

지정된 Contributor Insight 규칙을 비활성화하려면

다음 `disable-insight-rules` 예제에서는 지정된 계정에서 `Rule-A` 및 `Rule-B`라는 두 개의 Contributor Insight 규칙을 비활성화합니다.

```
aws cloudwatch disable-insight-rules \
    --rule-names Rule-A Rule-B
```

출력:

```
{
    "Failures": []
}
```

자세한 내용을 알아보려면 Amazon CloudWatch 사용 설명서의 Contributor Insights를 사용하여 카디널리티가 높은 데이터 분석하기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisableInsightRules를 참조하세요.

`enable-alarm-actions`

다음 코드 예시는 `enable-alarm-actions`의 사용 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.

- API 세부 정보는 AWS CLI Command Reference의 EnableAlarmActions를 참조하세요.

`enable-insight-rules`

다음 코드 예시는 `enable-insight-rules`의 사용 방법을 보여 줍니다.

AWS CLI

지정된 Contributor Insight 규칙을 활성화하려면

다음 enable-insight-rules 예제에서는 지정된 계정에서 Rule-A 및 Rule-B라는 Contributor Insight 2개를 활성화합니다.

```
aws cloudwatch enable-insight-rules \
    --rule-names Rule-A Rule-B
```

출력:

```
{
    "Failures": []
}
```

자세한 내용을 알아보려면 Amazon CloudWatch 사용 설명서의 Contributor Insights를 사용하여 카디널리티가 높은 데이터 분석하기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 EnableInsightRules를 참조하세요.

get-dashboard

다음 코드 예시는 get-dashboard의 사용 방법을 보여 줍니다.

AWS CLI

대시보드에 대한 정보 검색

다음 get-dashboard 예제는 특정 계정에서 Dashboard-A라는 대시보드에 대한 정보를 표시합니다.

```
aws cloudwatch get-dashboard \
    --dashboard-name Dashboard-A
```

출력:

```
{
    "DashboardArn": "arn:aws:cloudwatch::123456789012:dashboard/Dashboard-A",
    "DashboardBody": "{\"widgets\":[{\"type\":\"metric\",\"x\":0,\"y\":0,\"width
\":6,\"height\":6,\"properties\":{\"view\":\"timeSeries\",\"stacked\":false,
```

```
\"metrics\":[[\"AWS/EC2\",\"NetworkIn\",\"InstanceId\",\"i-0131f062232ade043\"],[\".
\",\"NetworkOut\",\".\",\".\"]],\"region\":\"us-east-1\"}}]}",
    "DashboardName": "Dashboard-A"
}
```

자세한 내용은 Amazon CloudWatch 사용 설명서의 Amazon CloudWatch 대시보드 사용 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetDashboard를 참조하세요.

get-insight-rule-report

다음 코드 예시는 `get-insight-rule-report`의 사용 방법을 보여 줍니다.

AWS CLI

Contributor Insights 규칙에서 수집한 시계열 데이터를 검색하려면

다음 `get-insight-rule-report` 예제에서는 Contributor Insights 규칙에서 수집한 시계열 데이터를 반환합니다.

```
aws cloudwatch get-insight-rule-report \
    --rule-name Rule-A \
    --start-time 2024-10-13T20:15:00Z \
    --end-time 2024-10-13T20:30:00Z \
    --period 300
```

출력:

```
{
    "KeyLabels": [
        "PartitionKey"
    ],
    "AggregationStatistic": "Sum",
    "AggregateValue": 0.5,
    "ApproximateUniqueCount": 1,
    "Contributors": [
        {
            "Keys": [
                "RequestID"
            ],
            "ApproximateAggregateValue": 0.5,
```

```
            "Datapoints": [
                {
                    "Timestamp": "2024-10-13T21:00:00+00:00",
                    "ApproximateValue": 0.5
                }
            ]
        }
    ],
    "RuleAttributes": []
}
```

자세한 내용을 알아보려면 Amazon CloudWatch 사용 설명서의 Contributor Insights를 사용하여 카디널리티가 높은 데이터 분석하기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetInsightRuleReport를 참조하세요.

get-metric-data

다음 코드 예시는 get-metric-data의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 수학 표현식을 사용하여 지정된 EC2의 평균 총 IOPS를 가져오려면

다음 get-metric-data 예제에서는 EBSReadOps 및 지표를 결합하는 지표 수학 확장을 i-abcdef 사용하여 InstanceID가 있는 EC2 인스턴스에 대한 CloudWatch EBSWriteOps 지표 값을 검색합니다.

```
aws cloudwatch get-metric-data \
    --metric-data-queries file://file.json \
    --start-time 2024-09-29T22:10:00Z \
    --end-time 2024-09-29T22:15:00Z
```

file.json의 콘텐츠:

```
[
    {
        "Id": "m3",
        "Expression": "(m1+m2)/300",
        "Label": "Avg Total IOPS"
    },
    {
```

```
        "Id": "m1",
        "MetricStat": {
            "Metric": {
                "Namespace": "AWS/EC2",
                "MetricName": "EBSReadOps",
                "Dimensions": [
                    {
                        "Name": "InstanceId",
                        "Value": "i-abcdef"
                    }
                ]
            },
            "Period": 300,
            "Stat": "Sum",
            "Unit": "Count"
        },
        "ReturnData": false
    },
    {
        "Id": "m2",
        "MetricStat": {
            "Metric": {
                "Namespace": "AWS/EC2",
                "MetricName": "EBSWriteOps",
                "Dimensions": [
                    {
                        "Name": "InstanceId",
                        "Value": "i-abcdef"
                    }
                ]
            },
            "Period": 300,
            "Stat": "Sum",
            "Unit": "Count"
        },
        "ReturnData": false
    }
]
```

출력:

```
{
    "MetricDataResults": [
```

```
        {
            "Id": "m3",
            "Label": "Avg Total IOPS",
            "Timestamps": [
                "2024-09-29T22:10:00+00:00"
            ],
            "Values": [
                96.85
            ],
            "StatusCode": "Complete"
        }
    ],
    "Messages": []
}
```

예제 2: CloudWatch 결제 지표를 사용하여 예상 AWS 요금을 모니터링하려면 다음을 수행합니다.

다음 `get-metric-data` 예제에서는 AWS/Billing 네임스페이스에서 `EstimatedCharges` CloudWatch 지표를 검색합니다.

```
aws cloudwatch get-metric-data \
    --metric-data-queries '[{"Id":"m1","MetricStat":{"Metric":
{"Namespace":"AWS/Billing","MetricName":"EstimatedCharges","Dimensions":
[{"Name":"Currency","Value":"USD"}]},"Period":21600,"Stat":"Maximum"}}]' \
    --start-time 2024-09-26T12:00:00Z \
    --end-time 2024-09-26T18:00:00Z \
    --region us-east-1
```

출력:

```
{
    "MetricDataResults": [
        {
            "Id": "m1",
            "Label": "EstimatedCharges",
            "Timestamps": [
                "2024-09-26T12:00:00+00:00"
            ],
            "Values": [
                542.38
            ],
            "StatusCode": "Complete"
        }
```

```
    ],
    "Messages": []
}
```

자세한 내용은 Amazon CloudWatch 사용 설명서에서 CloudWatch 지표에 수학 표현식 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetMetricData를 참조하세요.

get-metric-statistics

다음 코드 예시는 `get-metric-statistics`의 사용 방법을 보여 줍니다.

AWS CLI

EC2 인스턴스별 CPU 사용률을 가져오는 방법

다음 예제에서는 `get-metric-statistics` 명령을 사용하여 ID i-abcdef의 EC2 인스턴스에 대한 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 더 많은 예시는 Amazon CloudWatch 개발자 안내서의 지표에 대한 통계 얻기를 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
```

- API 세부 정보는 AWS CLI Command Reference의 [GetMetricStatistics](#)를 참조하세요.

`get-metric-stream`

다음 코드 예시는 `get-metric-stream`의 사용 방법을 보여 줍니다.

AWS CLI

지표 스트림에 대한 정보를 검색하려면

다음 `get-metric-stream` 예제는 지정된 계정에서 `QuickFull-GuaFbs`라는 지표 스트림에 대한 정보를 표시합니다.

```
aws cloudwatch get-metric-stream \
    --name QuickFull-GuaFbs
```

출력:

```
{
    "Arn": "arn:aws:cloudwatch:us-east-1:123456789012:metric-stream/QuickFull-GuaFbs",
    "Name": "QuickFull-GuaFbs",
    "FirehoseArn": "arn:aws:firehose:us-east-1:123456789012:deliverystream/MetricStreams-QuickFull-GuaFbs-WnySbECG",
    "RoleArn": "arn:aws:iam::123456789012:role/service-role/MetricStreams-FirehosePutRecords-JN10W9B3",
    "State": "running",
    "CreationDate": "2024-10-11T18:48:59.187000+00:00",
    "LastUpdateDate": "2024-10-11T18:48:59.187000+00:00",
    "OutputFormat": "json",
    "IncludeLinkedAccountsMetrics": false
}
```

자세한 내용은 Amazon CloudWatch 사용 설명서의 지표 스트림 사용 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetMetricStream을 참조하세요.

get-metric-widget-image

다음 코드 예시는 get-metric-widget-image의 사용 방법을 보여 줍니다.

AWS CLI

CPUUtilization의 스냅샷 그래프를 검색하려면 다음을 수행합니다.

다음 get-metric-widget-image 예제에서는 ID가 i-abcde인 EC2 인스턴스의 지표 CPUUtilization에 대한 스냅샷 그래프를 검색하고, 검색된 이미지를 로컬 시스템에 'image.png'라는 파일로 저장합니다.

```
aws cloudwatch get-metric-widget-image \
    --metric-widget '{"metrics":[["AWS/EC2","CPUUtilization","InstanceId","i-abcde"]]}' \
    --output-format png \
    --output text | base64 --decode > image.png
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 GetMetricWidgetImage를 참조하세요.

list-dashboards

다음 코드 예시는 `list-dashboards`의 사용 방법을 보여 줍니다.

AWS CLI

대시보드 목록을 검색하려면 다음을 수행합니다.

다음 `list-dashboards` 예제에서는 지정된 계정의 모든 대시보드를 나열합니다.

```
aws cloudwatch list-dashboards
```

출력:

```
{
    "DashboardEntries": [
        {
            "DashboardName": "Dashboard-A",
            "DashboardArn": "arn:aws:cloudwatch::123456789012:dashboard/Dashboard-
A",
            "LastModified": "2024-10-11T18:40:11+00:00",
            "Size": 271
        },
        {
            "DashboardName": "Dashboard-B",
            "DashboardArn": "arn:aws:cloudwatch::123456789012:dashboard/Dashboard-
B",
            "LastModified": "2024-10-11T18:44:41+00:00",
            "Size": 522
        }
    ]
}
```

자세한 내용은 Amazon CloudWatch 사용 설명서의 Amazon CloudWatch 대시보드 사용 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListDashboards 섹션을 참조하세요.

`list-metric-streams`

다음 코드 예시는 `list-metric-streams`의 사용 방법을 보여 줍니다.

AWS CLI

지표 스트림 목록을 검색하려면

다음 `list-metric-streams` 예제에서는 지정된 계정의 모든 지표 스트림을 나열합니다.

```
aws cloudwatch list-metric-streams
```

출력:

```
{
    "Entries": [
        {
            "Arn": "arn:aws:cloudwatch:us-east-1:123456789012:metric-stream/
QuickFull-GuaFbs",
            "CreationDate": "2024-10-11T18:48:59.187000+00:00",
            "LastUpdateDate": "2024-10-11T18:48:59.187000+00:00",
            "Name": "QuickFull-GuaFbs",
            "FirehoseArn": "arn:aws:firehose:us-east-1:123456789012:deliverystream/
MetricStreams-QuickFull-GuaFbs-WnySbECG",
            "State": "running",
            "OutputFormat": "json"
        }
    ]
}
```

자세한 내용은 Amazon CloudWatch 사용 설명서의 지표 스트림 사용 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListMetricStreams를 참조하세요.

`list-metrics`

다음 코드 예시는 `list-metrics`의 사용 방법을 보여 줍니다.

AWS CLI

Amazon SNS에 대한 지표를 나열하는 방법

다음 `list-metrics` 예제는 Amazon SNS에 대한 지표를 표시합니다.

```
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"
        }
```

```
    ]
}
```

- API 세부 정보는 AWS CLI Command Reference의 ListMetrics를 참조하세요.

`list-tags-for-resource`

다음 코드 예시는 `list-tags-for-resource`의 사용 방법을 보여 줍니다.

AWS CLI

기존 경보와 연결된 태그를 나열하려면*

다음 `list-tags-for-resource` 예제에서는 지정된 계정에서 demo라는 경보와 연결된 모든 태그를 나열합니다.

```
aws cloudwatch list-tags-for-resource \
    --resource-arn arn:aws:cloudwatch:us-east-1:123456789012:alarm:demo
```

출력:

```
{
    "Tags": [
        {
            "Key": "stack",
            "Value": "Production"
        },
        {
            "Key": "team",
            "Value": "Devops"
        }
    ]
}
```

자세한 내용은 Amazon CloudWatch 사용 설명서의 경보 및 태그 지정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource를 참조하세요.

`put-anomaly-detector`

다음 코드 예시는 `put-anomaly-detector`의 사용 방법을 보여 줍니다.

AWS CLI

이상 탐지 모델을 생성하려면 다음을 수행합니다.

다음 `put-anomaly-detector` 예제에서는 CloudWatch 지표에 대한 이상 탐지 모델을 생성합니다.

```
aws cloudwatch put-anomaly-detector \
    --namespace AWS/Logs \
    --metric-name IncomingBytes \
    --stat SampleCount
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon CloudWatch 사용 설명서의 CloudWatch 이상 탐지 사용 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutAnomalyDetector를 참조하세요.

put-composite-alarm

다음 코드 예시는 `put-composite-alarm`의 사용 방법을 보여 줍니다.

AWS CLI

복합 CloudWatch 경보를 생성하려면

다음 `put-composite-alarm` 예제에서는 지정된 계정에 `ProdAlarm`이라는 복합 경보를 생성합니다.

```
aws cloudwatch put-composite-alarm \
    --alarm-name ProdAlarm \
    --alarm-rule "ALARM(CPUUtilizationTooHigh) AND ALARM(MemUsageTooHigh)" \
    --alarm-actions arn:aws:sns:us-east-1:123456789012:demo \
    --actions-enabled
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon CloudWatch 사용 설명서의 복합 경보 생성 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutCompositeAlarm을 참조하세요.

put-dashboard

다음 코드 예시는 put-dashboard의 사용 방법을 보여 줍니다.

AWS CLI

대시보드 생성

다음 put-dashboard 예제에서는 지정된 계정에 Dashboard-A라는 대시보드를 생성합니다.

```
aws cloudwatch put-dashboard \
    --dashboard-name Dashboard-A \
    --dashboard-body '{"widgets":
[{"height":6,"width":6,"y":0,"x":0,"type":"metric","properties":
{"view":"timeSeries","stacked":false,"metrics":
[["Namespace","CPUUtilization","Environment","Prod","Type","App"]],"region":"us-
east-1"}}]}'
```

출력:

```
{
    "DashboardValidationMessages": []
}
```

자세한 내용은 Amazon CloudWatch 사용 설명서의 Amazon CloudWatch 경보 생성 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutDashboard를 참조하세요.

put-insight-rule

다음 코드 예시는 put-insight-rule의 사용 방법을 보여 줍니다.

AWS CLI

Contributor Insights 규칙을 생성하려면

다음 put-insight-rule 예제에서는 지정된 계정에서 VPCFlowLogsContributorInsights라는 Contributor Insight 규칙을 생성합니다.

```
aws cloudwatch put-insight-rule \
    --rule-name VPCFlowLogsContributorInsights \
```

```
    --rule-definition file://insight-rule.json \
    --rule-state ENABLED
```

`insight-rule.json`의 콘텐츠:

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "AggregateOn": "Count",
    "Contribution": {
        "Filters": [],
        "Keys": [
            "tcp-flag"
        ]
    },
    "LogFormat": "CLF",
    "LogGroupNames": [
        "/vpc/flowlogs/*"
    ],
    "Fields": {
        "23": "tcp-flag"
    }
}
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon CloudWatch 사용 설명서의 CloudWatch에서 Contributor Insights 규칙 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutInsightRule을 참조하세요.

`put-metric-alarm`

다음 코드 예시는 `put-metric-alarm`의 사용 방법을 보여 줍니다.

AWS CLI

CPU 사용률이 70%를 초과할 때 Amazon Simple Notification Service 이메일 메시지 보내기

다음 예제에서는 `put-metric-alarm` 명령을 사용하여 CPU 사용률이 70%를 초과할 때 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
```

- API 세부 정보는 AWS CLI 명령 참조의 PutMetricAlarm을 참조하세요.

put-metric-data

다음 코드 예시는 `put-metric-data`의 사용 방법을 보여 줍니다.

AWS CLI

Amazon CloudWatch에 사용자 지정 지표를 게시하는 방법

다음 예제에서는 `put-metric-data` 명령을 사용하여 Amazon CloudWatch에 사용자 지정 지표를 게시합니다.

```
aws cloudwatch put-metric-data --namespace "Usage Metrics" --metric-data file://
metric.json
```

지표 자체의 값은 JSON 파일인 `metric.json`에 저장됩니다.

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
```

- API 세부 정보는 AWS CLI Command Reference의 PutMetricData를 참조하세요.

`put-metric-stream`

다음 코드 예시는 `put-metric-stream`의 사용 방법을 보여 줍니다.

AWS CLI

지표 스트림을 생성하려면

다음 `put-metric-stream` 예제에서는 지정된 계정에 `QuickFull-GuaFb`라는 지표 스트림을 생성합니다.

```
aws cloudwatch put-metric-stream \
    --name QuickFull-GuaFbs \
    --firehose-arn arn:aws:firehose:us-east-1:123456789012:deliverystream/
MetricStreams-QuickFull-GuaFbs-WnySbECG \
    --role-arn arn:aws:iam::123456789012:role/service-role/MetricStreams-
FirehosePutRecords-JN10W9B3 \
    --output-format json \
    --no-include-linked-accounts-metrics
```

출력:

```
{
    "Arn": "arn:aws:cloudwatch:us-east-1:123456789012:metric-stream/QuickFull-GuaFbs"
}
```

자세한 내용은 Amazon CloudWatch 사용 설명서의 지표 스트림 설정 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutMetricStream을 참조하세요.

set-alarm-state

다음 코드 예시는 `set-alarm-state`의 사용 방법을 보여 줍니다.

AWS CLI

경보 상태를 일시적으로 변경하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 SetAlarmState를 참조하세요.

start-metric-streams

다음 코드 예시는 `start-metric-streams`의 사용 방법을 보여 줍니다.

AWS CLI

지정된 지표 스트림을 시작하려면

다음 `start-metric-streams` 예제에서는 지정된 계정에서 `QuickFull-GuaFbs`라는 지표 스트림을 시작합니다.

```
aws cloudwatch start-metric-streams \
```

```
    --names QuickFull-GuaFbs
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon CloudWatch 사용 설명서의 지표 스트림 사용 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartMetricStreams를 참조하세요.

stop-metric-streams

다음 코드 예시는 stop-metric-streams의 사용 방법을 보여 줍니다.

AWS CLI

지정된 지표 스트림을 중지하려면

다음 stop-metric-streams 예제에서는 지정된 계정에서 QuickFull-GuaFbs라는 지표 스트림을 중지합니다.

```
aws cloudwatch stop-metric-streams \
    --names QuickFull-GuaFbs
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon CloudWatch 사용 설명서의 지표 스트림 사용 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StopMetricStreams를 참조하세요.

tag-resource

다음 코드 예시는 tag-resource의 사용 방법을 보여 줍니다.

AWS CLI

지정된 리소스에 하나 이상의 태그를 추가하려면

다음 tag-resource 예제에서는 지정된 계정에서 이름이 demo인 Cloudwatch 경보에 태그 2개를 추가합니다.

```
aws cloudwatch tag-resource \
    --resource-arn arn:aws:cloudwatch:us-east-1:123456789012:alarm:demo \
    --tags Key=stack,Value=Production Key=team,Value=Devops
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon CloudWatch 사용 설명서에서 Amazon CloudWatch 리소스 태깅을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

`untag-resource`

다음 코드 예시는 `untag-resource`의 사용 방법을 보여 줍니다.

AWS CLI

지정된 리소스에서 하나 이상의 태그를 제거하려면

다음 `untag-resource` 예제에서는 지정된 계정에서 이름이 demo인 Cloudwatch 경보에서 태그 2개를 제거합니다.

```
aws cloudwatch untag-resource \
    --resource-arn arn:aws:cloudwatch:us-east-1:123456789012:alarm:demo \
    --tag-keys stack team
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon CloudWatch 사용 설명서에서 Amazon CloudWatch 리소스 태깅을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

AWS CLI를 사용하는 CloudWatch Logs 예제

다음 코드 예제에서는 CloudWatch Logs에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여 줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

create-log-group

다음 코드 예시는 create-log-group의 사용 방법을 보여 줍니다.

AWS CLI

다음 명령은 이름이 my-logs인 로그 그룹을 생성합니다.

```
aws logs create-log-group --log-group-name my-logs
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateLogGroup를 참조하세요.

create-log-stream

다음 코드 예시는 create-log-stream의 사용 방법을 보여 줍니다.

AWS CLI

다음 명령은 my-logs 로그 그룹에서 이름이 20150601인 로그 스트림을 생성합니다.

```
aws logs create-log-stream --log-group-name my-logs --log-stream-name 20150601
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateLogStream을 참조하세요.

delete-log-group

다음 코드 예시는 delete-log-group의 사용 방법을 보여 줍니다.

AWS CLI

다음 명령은 이름이 my-logs인 로그 그룹을 삭제합니다.

```
aws logs delete-log-group --log-group-name my-logs
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteLogGroup을 참조하세요.

delete-log-stream

다음 코드 예시는 delete-log-stream의 사용 방법을 보여 줍니다.

AWS CLI

다음 명령은 이름이 my-logs인 로그 스트림에서 이름이 20150531인 로그 그룹을 삭제합니다.

```
aws logs delete-log-stream --log-group-name my-logs --log-stream-name 20150531
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteLogStream을 참조하세요.

delete-retention-policy

다음 코드 예시는 delete-retention-policy의 사용 방법을 보여 줍니다.

AWS CLI

다음 명령은 이전에 이름이 my-logs인 로그 그룹에 적용되었던 보존 정책을 제거합니다.

```
aws logs delete-retention-policy --log-group-name my-logs
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteRetentionPolicy를 참조하세요.

describe-log-groups

다음 코드 예시는 describe-log-groups의 사용 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:*"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeLogGroups를 참조하세요.

describe-log-streams

다음 코드 예시는 describe-log-streams의 사용 방법을 보여 줍니다.

AWS CLI

다음 명령은 로그 그룹 my-logs의 2015 접두사로 시작하는 모든 로그 스트림을 보여 줍니다.

```
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
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeLogStreams를 참조하세요.

get-log-events

다음 코드 예시는 get-log-events의 사용 방법을 보여 줍니다.

AWS CLI

다음 명령은 my-logs 로그 그룹에서 이름이 20150601인 로그 스트림의 로그 이벤트를 검색합니다.

```
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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetLogEvents를 참조하세요.

put-log-events

다음 코드 예시는 put-log-events의 사용 방법을 보여 줍니다.

AWS CLI

다음 명령은 my-logs 로그 그룹에서 이름이 20150601인 로그 스트림에 로그 이벤트를 설정합니다.

```
aws logs put-log-events --log-group-name my-logs --log-stream-name 20150601 --log-events file://events
```

출력:

```
{
    "nextSequenceToken": "49542672486831074009579604567656788214806863282469607346"
}
```

위의 예제에서는 현재 디렉터리에 있는 events 파일에서 이벤트의 JSON 배열을 읽습니다.

```
[
  {
    "timestamp": 1433190184356,
    "message": "Example Event 1"
  },
  {
    "timestamp": 1433190184358,
    "message": "Example Event 2"
  },
  {
    "timestamp": 1433190184360,
    "message": "Example Event 3"
  }
]
```

이후 호출할 때마다 이전 호출에서 제공한 시퀀스 토큰 옵션을 다음 시퀀스 토큰으로 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [PutLogEvents](#)를 참조하세요.

`put-retention-policy`

다음 코드 예시는 `put-retention-policy`의 사용 방법을 보여 줍니다.

AWS CLI

다음 명령은 `my-logs` 로그 그룹에 5일 보존 정책을 추가합니다.

```
aws logs put-retention-policy --log-group-name my-logs --retention-in-days 5
```

- API 세부 정보는 AWS CLI 명령 참조의 [PutRetentionPolicy](#)를 참조하세요.

AWS CLI를 사용한 CloudWatch Network Monitoring 예시

다음 코드 예시에서는 CloudWatch Network Monitoring과 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- [작업](#)

작업

`create-monitor`

다음 코드 예시에서는 `create-monitor`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 집계 기간이 있는 네트워크 모니터 생성

다음 create-monitor 예시에서는 aggregationPeriod가 30초로 설정된 Example_NetworkMonitor라는 모니터를 생성합니다. 모니터에 연결된 프로브가 없기 때문에 모니터의 초기 state는 INACTIVE가 됩니다. 프로브가 추가될 때에만 상태가 ACTIVE로 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 CloudWatch 사용자 안내서의 Amazon CloudWatch Network Monitor 작동 방식을 참조하세요.

예시 2: TCP를 사용하고 태그도 포함하는 프로브 및 네트워크 모니터 생성

다음 create-monitor 예시에서는 Example_NetworkMonitor라는 모니터를 생성합니다. 또한 명령은 ICMP 프로토콜을 사용하고 태그를 포함하는 하나의 프로브를 생성합니다. aggregationPeriod가 요청에 전달되지 않으므로 60초가 기본값으로 설정됩니다. 프로브가 있는 모니터의 state는 모니터가 ACTIVE가 될 때까지 PENDING으로 유지됩니다. 이 작업은 몇 분 정도 걸릴 수 있으며, 이때 state가 ACTIVE로 변경되면 CloudWatch 지표를 볼 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 CloudWatch 사용자 안내서의 Amazon CloudWatch Network Monitor 작동 방식을 참조하세요.

예시 3: ICMP를 사용하고 태그도 포함하는 프로브 및 네트워크 모니터 생성

다음 `create-monitor` 예시에서는 `aggregationPeriod`가 30초인 `Example_NetworkMonitor`라는 모니터를 생성합니다. 또한 명령은 `ICMP` 프로토콜을 사용하고 태그를 포함하는 하나의 프로브를 생성합니다. `aggregationPeriod`가 요청에 전달되지 않으므로 `60`초가 기본값으로 설정됩니다. 프로브가 있는 모니터의 `state`는 모니터가 `ACTIVE`가 될 때까지 `PENDING`으로 유지됩니다. 이 작업은 몇 분 정도 걸릴 수 있으며, 이때 `state`가 `ACTIVE`로 변경되면 CloudWatch 지표를 볼 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 CloudWatch 사용자 안내서의 Amazon CloudWatch Network Monitor 작동 방식을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateMonitor를 참조하세요.

create-probe

다음 코드 예시에서는 create-probe의 사용 방법을 보여줍니다.

AWS CLI

예시 1: TCP를 사용하는 프로브를 생성하고 네트워크 모니터에 추가

다음 create-probe 예시에서는 TCP protocol을 사용하는 프로브를 생성하고 Example_NetworkMonitor라는 모니터에 프로브를 추가합니다. 일단 생성되면 프로브가 있는 모니터의 state는 모니터가 ACTIVE가 될 때까지 PENDING으로 유지됩니다. 이 작업은 몇 분 정도 걸릴 수 있으며, 이때 상태가 ACTIVE로 변경되면 CloudWatch 지표를 볼 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"
    }
}
```

예시 2: ICMP를 사용하는 프로브를 생성하고 네트워크 모니터에 추가

다음 `create-probe` 예시에서는 ICMP `protocol`을 사용하는 프로브를 생성하고 `Example_NetworkMonitor`라는 모니터에 프로브를 추가합니다. 일단 생성되면 프로브가 있는 모니터의 `state`는 모니터가 `ACTIVE`가 될 때까지 `PENDING`으로 유지됩니다. 이 작업은 몇 분 정도 걸릴 수 있으며, 이때 상태가 `ACTIVE`로 변경되면 CloudWatch 지표를 볼 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 CloudWatch 사용자 안내서의 [Amazon CloudWatch Network Monitor 작동 방식](#)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [CreateProbe](#)를 참조하세요.

`delete-monitor`

다음 코드 예시에서는 `delete-monitor`의 사용 방법을 보여줍니다.

AWS CLI

모니터 삭제

다음 `delete-monitor` 예시에서는 `Example_NetworkMonitor`라는 모니터를 삭제합니다.

```
aws networkmonitor delete-monitor \
    --monitor-name Example_NetworkMonitor
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon CloudWatch 사용자 안내서의 Amazon CloudWatch Network Monitor 작동 방식을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteMonitor를 참조하세요.

delete-probe

다음 코드 예시에서는 delete-probe의 사용 방법을 보여줍니다.

AWS CLI

프로브 삭제

다음 delete-probe 예시에서는 Example_NetworkMonitor라는 네트워크 모니터에서 ID가 probe-12345인 프로브를 삭제합니다.

```
aws networkmonitor delete-probe \
    --monitor-name Example_NetworkMonitor \
    --probe-id probe-12345
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon CloudWatch 사용자 안내서의 Amazon CloudWatch Network Monitor 작동 방식을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteProbe를 참조하세요.

get-monitor

다음 코드 예시에서는 get-monitor의 사용 방법을 보여줍니다.

AWS CLI

모니터 정보 가져오기

다음 get-monitor 예시에서는 Example_NetworkMonitor라는 모니터의 정보를 가져옵니다.

```
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 CloudWatch 사용자 안내서의 Amazon CloudWatch Network Monitor 작동 방식을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetMonitor를 참조하세요.

get-probe

다음 코드 예시에서는 get-probe의 사용 방법을 보여줍니다.

AWS CLI

프로브 세부 정보 보기

다음 get-probe 예시에서는 Example_NetworkMonitor라는 모니터에 연결되고 probeID가 probe-12345인 프로브의 세부 정보를 반환합니다.

```
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 CloudWatch 사용자 안내서의 Amazon CloudWatch Network Monitor 작동 방식을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetProbe를 참조하세요.

list-monitors

다음 코드 예시에서는 `list-monitors`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 모든 모니터 나열(단일 모니터)

다음 `list-monitors` 예시에서는 단일 모니터의 목록만 반환합니다. 모니터의 `state`는 `ACTIVE`이고 `aggregationPeriod`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 CloudWatch 사용자 안내서의 [Amazon CloudWatch Network Monitor 작동 방식]을 참조하세요.

예시 2: 모든 모니터 나열(여러 모니터)

다음 `list-monitors` 예시에서는 3개의 모니터 목록을 반환합니다. 한 모니터의 `state`는 `ACTIVE`이고 CloudWatch 지표를 생성합니다. 다른 두 모니터의 상태는 `INACTIVE`이고 CloudWatch 지표를 생성하지 않습니다. 세 모니터 모두 60초의 `aggregationPeriod`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 CloudWatch 사용자 안내서의 Amazon CloudWatch Network Monitor 작동 방식을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListMonitors를 참조하세요.

list-tags-for-resource

다음 코드 예시에서는 list-tags-for-resource의 사용 방법을 보여줍니다.

AWS CLI

리소스의 태그 나열

다음 list-tags-for-resource 예시에서는 Example_NetworkMonitor라는 모니터의 태그 목록을 반환합니다.

```
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 CloudWatch 사용자 안내서의 Amazon CloudWatch Network Monitor 작동 방식을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource를 참조하세요.

tag-resource

다음 코드 예시에서는 tag-resource의 사용 방법을 보여줍니다.

AWS CLI

리소스에 태그 지정

다음 tag-resource 예시에서는 Example_NetworkMonitor라는 모니터에 Environment=Dev 및 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 CloudWatch 사용자 안내서의 Amazon CloudWatch Network Monitor 작동 방식을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

untag-resource

다음 코드 예시에서는 untag-resource 코드를 사용하는 방법을 보여줍니다.

AWS CLI

리소스에서 태그 제거

다음 untag-resource 예시에서는 키와 값 페어가 Environment Application인 tag-keys 파라미터를 Example_NetworkMonitor라는 모니터와의 연결에서 제거합니다.

```
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 CloudWatch 사용자 안내서의 [Amazon CloudWatch Network Monitor 작동 방식]을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [UntagResource]를 참조하세요.

`update-monitor`

다음 코드 예시에서는 `update-monitor`의 사용 방법을 보여줍니다.

AWS CLI

모니터 업데이트

다음 `update-monitor` 예시에서는 모니터의 `aggregationPeriod`를 `60`초에서 `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 CloudWatch 사용자 안내서의 [Amazon CloudWatch Network Monitor 작동 방식]을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [UpdateMonitor]를 참조하세요.

`update-probe`

다음 코드 예시에서는 `update-probe`의 사용 방법을 보여줍니다.

AWS CLI

프로브 업데이트

다음 update-probe 예시에서는 프로브의 원래 destination IP 주소를 업데이트하고 packetSize도 60으로 업데이트합니다.

```
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 CloudWatch 사용자 안내서의 Amazon CloudWatch Network Monitor 작동 방식을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateProbe를 참조하세요.

AWS CLI를 사용한 CloudWatch Observability Access Monitor 예제

다음 코드 예제에서는 CloudWatch Observability Access Monitor와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

create-link

다음 코드 예시는 create-link의 사용 방법을 보여 줍니다.

AWS CLI

링크 생성

다음 create-link 예제에서는 소스 계정과 모니터링 계정에서 생성한 싱크 간에 링크를 생성합니다.

```
aws oam create-link \
    --label-template sourceAccount \
    --resource-types AWS::CloudWatch::Metric \
    --sink-identifier arn:aws:oam:us-east-2:123456789012:sink/a1b2c3d4-5678-90ab-cdef-example12345
```

출력:

```
{
    "Arn": "arn:aws:oam:us-east-2:123456789111:link/a1b2c3d4-5678-90ab-cdef-example11111",
    "Id": "a1b2c3d4-5678-90ab-cdef-example11111",
    "Label": "sourceAccount",
    "LabelTemplate": "sourceAccount",
    "ResourceTypes": [
        "AWS::CloudWatch::Metric"
    ],
    "SinkArn": "arn:aws:oam:us-east-2:123456789012:sink/a1b2c3d4-5678-90ab-cdef-example12345",
```

```
    "Tags": {}
}
```

자세한 내용은 Amazon CloudWatch 사용 설명서의 CloudWatch 교차 계정 관찰성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateLink 섹션을 참조하세요.

`create-sink`

다음 코드 예시는 `create-sink`의 사용 방법을 보여 줍니다.

AWS CLI

싱크를 생성하려면

다음 `create-sink` 예제에서는 현재 계정에 싱크를 생성하여 CloudWatch 교차 계정 관찰성에서 모니터링 계정으로 사용할 수 있습니다.

```
aws oam create-sink \
    --name DemoSink
```

출력:

```
{
    "Arn": "arn:aws:oam:us-east-2:123456789012:sink/a1b2c3d4-5678-90ab-cdef-
example12345",
    "Id": "a1b2c3d4-5678-90ab-cdef-example12345",
    "Name": "DemoSink",
    "Tags": {}
}
```

자세한 내용은 Amazon CloudWatch 사용 설명서의 CloudWatch 교차 계정 관찰성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateSink 섹션을 참조하세요.

`delete-link`

다음 코드 예시는 `delete-link`의 사용 방법을 보여 줍니다.

AWS CLI

링크 삭제

다음 `delete-link` 예제에서는 모니터링 계정 싱크와 소스 계정 간의 링크를 삭제합니다.

```
aws oam delete-link \
    --identifier arn:aws:oam:us-east-2:123456789111:link/a1b2c3d4-5678-90ab-cdef-example11111
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon CloudWatch 사용 설명서의 [CloudWatch 교차 계정 관찰성]을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteLink] 섹션을 참조하세요.

delete-sink

다음 코드 예시는 `delete-sink`의 사용 방법을 보여 줍니다.

AWS CLI

싱크를 삭제하려면

다음 `delete-sink` 예제에서는 싱크를 삭제합니다. 싱크에 대한 모든 링크를 삭제해야 해당 싱크를 삭제할 수 있습니다.

```
aws oam delete-sink \
    --identifier arn:aws:oam:us-east-2:123456789012:sink/a1b2c3d4-5678-90ab-cdef-example12345
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon CloudWatch 사용 설명서의 [CloudWatch 교차 계정 관찰성]을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteSink] 섹션을 참조하세요.

get-link

다음 코드 예시는 `get-link`의 사용 방법을 보여 줍니다.

AWS CLI

하나의 링크에 대한 전체 정보를 반환하려면

다음 `get-link` 예제에서는 링크에 대한 전체 정보를 반환합니다.

```
aws oam get-link \
    --identifier arn:aws:oam:us-east-2:123456789111:link/a1b2c3d4-5678-90ab-cdef-example11111
```

출력:

```
{
    "Arn": "arn:aws:oam:us-east-2:123456789111:link/a1b2c3d4-5678-90ab-cdef-example11111",
    "Id": "a1b2c3d4-5678-90ab-cdef-example11111",
    "Label": "sourceAccount",
    "LabelTemplate": "sourceAccount",
    "ResourceTypes": [
        "AWS::CloudWatch::Metric"
    ],
    "SinkArn": "arn:aws:oam:us-east-2:123456789012:sink/a1b2c3d4-5678-90ab-cdef-example12345",
    "Tags": {}
}
```

자세한 내용은 Amazon CloudWatch 사용 설명서의 CloudWatch 교차 계정 관찰성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetLink 섹션을 참조하세요.

get-sink-policy

다음 코드 예시는 get-sink-policy의 사용 방법을 보여 줍니다.

AWS CLI

싱크에 연결된 현재 싱크 정책을 반환하려면

다음 get-sink-policy 예제에서는 싱크에 연결된 현재 싱크 정책을 반환합니다.

```
aws oam get-sink-policy \
    --sink-identifier arn:aws:oam:us-east-2:123456789012:sink/a1b2c3d4-5678-90ab-cdef-example12345
```

출력:

```
{
```

```
    "SinkArn": "arn:aws:oam:us-east-2:123456789012:sink/a1b2c3d4-5678-90ab-cdef-example12345",
    "SinkId": "a1b2c3d4-5678-90ab-cdef-example12345",
    "Policy": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":
\"Allow\",\"Principal\":{\"AWS\":\"arn:aws:iam::123456789111:root\"},
\"Action\":[\"oam:CreateLink\",\"oam:UpdateLink\"],\"Resource\":\"*\",
\"Condition\":{\"ForAllValues:StringEquals\":{\"oam:ResourceTypes\":
[\"AWS::Logs::LogGroup\",\"AWS::CloudWatch::Metric\",\"AWS::XRay::Trace\",
\"AWS::ApplicationInsights::Application\"]}}}]}"
}
```

자세한 내용은 Amazon CloudWatch 사용 설명서의 CloudWatch 교차 계정 관찰성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetSinkPolicy를 참조하세요.

get-sink

다음 코드 예시는 get-sink의 사용 방법을 보여 줍니다.

AWS CLI

하나의 모니터링 계정 싱크에 대한 전체 정보를 반환하려면

다음 get-sink 예제에서는 모니터링 계정 싱크에 대한 전체 정보를 반환합니다.

```
aws oam get-sink \
    --identifier arn:aws:oam:us-east-2:123456789012:sink/a1b2c3d4-5678-90ab-cdef-example12345
```

출력:

```
{
    "Arn": "arn:aws:oam:us-east-2:123456789012:sink/a1b2c3d4-5678-90ab-cdef-example12345",
    "Id": "a1b2c3d4-5678-90ab-cdef-example12345",
    "Name": "DemoSink",
    "Tags": {}
}
```

자세한 내용은 Amazon CloudWatch 사용 설명서의 CloudWatch 교차 계정 관찰성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetSink 섹션을 참조하세요.

`list-attached-links`

다음 코드 예시는 `list-attached-links`의 사용 방법을 보여 줍니다.

AWS CLI

이 모니터링 계정 싱크에 연결된 소스 계정 링크 목록을 반환하려면

다음 `list-attached-links` 예제에서는 이 모니터링 계정 싱크에 연결된 소스 계정 링크 목록을 반환합니다.

```
aws oam list-attached-links \
    --sink-identifier arn:aws:oam:us-east-2:123456789012:sink/a1b2c3d4-5678-90ab-
cdef-example12345
```

출력:

```
{
    "Items": [{
        "Label": "Monitoring account",
        "LinkArn": "arn:aws:oam:us-east-2:123456789111:link/a1b2c3d4-5678-90ab-cdef-
example11111",
        "ResourceTypes": [
            "AWS::ApplicationInsights::Application",
            "AWS::Logs::LogGroup",
            "AWS::CloudWatch::Metric",
            "AWS::XRay::Trace"
        ]
    }]
}
```

자세한 내용은 Amazon CloudWatch 사용 설명서의 [CloudWatch 교차 계정 관찰성]을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ListAttachedLinks] 섹션을 참조하세요.

`list-links`

다음 코드 예시는 `list-links`의 사용 방법을 보여 줍니다.

AWS CLI

모니터링 계정 싱크 하나에 대한 링크 목록을 반환하려면

다음 list-links 예제에서는 모니터링 계정 싱크 하나에 대한 링크 목록을 반환합니다. 소스 계정에서 이 작업을 실행하여이 소스 계정에 있는 모니터링 계정 싱크에 대한 링크 목록을 반환합니다.

```
aws oam list-links
```

출력:

```
{
    "Items": [{
        "Arn": "arn:aws:oam:us-east-2:123456789111:link/a1b2c3d4-5678-90ab-cdef-
example11111",
        "Id": "a1b2c3d4-5678-90ab-cdef-example11111",
        "Label": "sourceAccount",
        "ResourceTypes": [
            "AWS::CloudWatch::Metric"
        ],
        "SinkArn": "arn:aws:oam:us-east-2:123456789012:sink/a1b2c3d4-5678-90ab-cdef-
example12345"
    }]
}
```

자세한 내용은 Amazon CloudWatch 사용 설명서의 CloudWatch 교차 계정 관찰성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListLinks 섹션을 참조하세요.

list-sinks

다음 코드 예시는 list-sinks의 사용 방법을 보여 줍니다.

AWS CLI

모니터링 계정에서 생성된 싱크 목록을 반환하려면

다음 list-sinks 예제에서는 모니터링 계정에서 생성된 싱크 목록을 반환합니다. 모니터링 계정에서 이 작업을 실행합니다.

```
aws oam list-sinks
```

출력:

```
{
```

```
    "Items": [
        {
            "Arn": "arn:aws:oam:us-east-2:123456789012:sink/a1b2c3d4-5678-90ab-cdef-example12345",
            "Id": "a1b2c3d4-5678-90ab-cdef-example12345",
            "Name": "DemoSink"
        }
    ]
}
```

자세한 내용은 Amazon CloudWatch 사용 설명서의 CloudWatch 교차 계정 관찰성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListSinks 섹션을 참조하세요.

list-tags-for-resource

다음 코드 예시는 list-tags-for-resource의 사용 방법을 보여 줍니다.

AWS CLI

리소스에 연결된 태그를 표시하려면

다음 list-tags-for-resource 예제에서는 싱크와 연결된 태그를 표시합니다.

```
aws oam list-tags-for-resource \
    --resource-arn arn:aws:oam:us-east-2:123456789012:sink/a1b2c3d4-5678-90ab-cdef-example12345
```

출력:

```
{
    "Tags": {
        "Team": "Devops"
    }
}
```

자세한 내용은 Amazon CloudWatch 사용 설명서의 CloudWatch 교차 계정 관찰성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource를 참조하세요.

put-sink-policy

다음 코드 예시는 put-sink-policy의 사용 방법을 보여 줍니다.

AWS CLI

리소스 정책을 생성하거나 업데이트하려면

다음 put-sink-policy 예제에서는 소스 계정에 모니터링 계정 싱크에 연결할 수 있는 권한을 부여하는 리소스 정책을 생성합니다.

```
aws oam put-sink-policy \
    --policy '{"Version":"2012-10-17","Statement":[{"Effect":"Allow","Principal":
{"AWS":"arn:aws:iam::123456789111:root"},"Action":
["oam:CreateLink","oam:UpdateLink"],"Resource":"*","Condition":
{"ForAllValues:StringEquals":{"oam:ResourceTypes":
["AWS::Logs::LogGroup","AWS::CloudWatch::Metric","AWS::XRay::Trace","AWS::ApplicationInsight
 \
    --sink-identifier arn:aws:oam:us-east-2:123456789012:sink/a1b2c3d4-5678-90ab-
cdef-example12345
```

출력:

```
{
    "SinkArn": "arn:aws:oam:us-east-2:123456789012:sink/a1b2c3d4-5678-90ab-cdef-
example12345",
    "SinkId": "a1b2c3d4-5678-90ab-cdef-example12345",
    "Policy": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":
\"Allow\",\"Principal\":{\"AWS\":\"arn:aws:iam::123456789111:root\"},
\"Action\":[\"oam:CreateLink\",\"oam:UpdateLink\"],\"Resource\":\"*\",
\"Condition\":{\"ForAllValues:StringEquals\":{\"oam:ResourceTypes\":
[\"AWS::Logs::LogGroup\",\"AWS::CloudWatch::Metric\",\"AWS::XRay::Trace\",
\"AWS::ApplicationInsights::Application\"]}}}]}"
}
```

자세한 내용은 Amazon CloudWatch 사용 설명서의 CloudWatch 교차 계정 관찰성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutSinkPolicy를 참조하세요.

tag-resource

다음 코드 예시는 tag-resource의 사용 방법을 보여 줍니다.

AWS CLI

지정된 리소스에 하나 이상의 태그를 지정하려면

다음 tag-resource 예제에서는 arn:aws:oam:us-east-2:123456789012:sink/a1b2c3d4-5678-90ab-cdef-example12345 싱크에 태그를 지정합니다.

```
aws oam tag-resource \
    --resource-arn arn:aws:oam:us-east-2:123456789012:sink/a1b2c3d4-5678-90ab-cdef-example12345 \
    --tags team=Devops
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon CloudWatch 사용 설명서의 CloudWatch 교차 계정 관찰성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

untag-resource

다음 코드 예시는 untag-resource의 사용 방법을 보여 줍니다.

AWS CLI

지정된 리소스에서 하나 이상의 태그를 제거하려면

다음 untag-resource 예제에서는 arn:aws:oam:us-east-2:123456789012:sink/a1b2c3d4-5678-90ab-cdef-example12345 싱크에서 team 키가 있는 태그를 제거합니다.

```
aws oam untag-resource \
    --resource-arn arn:aws:oam:us-east-2:123456789012:sink/f3f42f60-f0f2-425c-1234-12347bdd821f \
    --tag-keys team
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon CloudWatch 사용 설명서의 CloudWatch 교차 계정 관찰성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

update-link

다음 코드 예시는 update-link의 사용 방법을 보여 줍니다.

AWS CLI

소스 계정에서 연결된 모니터링 계정 싱크로 공유되는 데이터 유형을 변경하려면

다음 update-link 예제에서는 리소스 유형 AWS::CloudWatch::Metric 및 AWS::Logs::LogGroup으로 arn:aws:oam:us-east-2:123456789111:link/0123e691-e7ef-43fa-1234-c57c837fced0 링크를 업데이트합니다.

```
aws oam update-link \
    --identifier arn:aws:oam:us-east-2:123456789111:link/a1b2c3d4-5678-90ab-cdef-example11111 \
    --resource-types "AWS::CloudWatch::Metric" "AWS::Logs::LogGroup"
```

출력:

```
{
    "Arn": "arn:aws:oam:us-east-2:123456789111:link/a1b2c3d4-5678-90ab-cdef-example11111",
    "Id": "a1b2c3d4-5678-90ab-cdef-example11111",
    "Label": "sourceAccount",
    "LabelTemplate": "sourceAccount",
    "ResourceTypes": [
        "AWS::CloudWatch::Metric",
        "AWS::Logs::LogGroup"
    ],
    "SinkArn": "arn:aws:oam:us-east-2:123456789012:sink/a1b2c3d4-5678-90ab-cdef-example12345",
    "Tags": {}
}
```

자세한 내용은 Amazon CloudWatch 사용 설명서의 CloudWatch 교차 계정 관찰성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateLink 섹션을 참조하세요.

AWS CLI를 사용한 CloudWatch Observability Admin 예제

다음 코드 예제에서는 CloudWatch Observability Admin과 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

get-telemetry-evaluation-status-for-organization

다음 코드 예시는 get-telemetry-evaluation-status-for-organization의 사용 방법을 보여 줍니다.

AWS CLI

조직의 원격 측정 온보딩 상태를 가져오려면

다음 get-telemetry-evaluation-status-for-organization 예제에서는 조직에 대한 원격 측정 구성 기능의 현재 온보딩 상태를 반환합니다.

```
aws observabilityadmin get-telemetry-evaluation-status-for-organization
```

출력:

```
{
    "Status": "RUNNING"
}
```

자세한 내용은 Amazon CloudWatch 사용 설명서의 CloudWatch 원격 측정 구성 감사를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetTelemetryEvaluationStatusForOrganization 섹션을 참조하세요.

get-telemetry-evaluation-status

다음 코드 예시는 get-telemetry-evaluation-status의 사용 방법을 보여 줍니다.

AWS CLI

계정의 원격 측정 온보딩 상태를 가져오려면

다음 get-telemetry-evaluation-status 예제에서는 지정된 계정에서 원격 측정 구성 기능의 현재 온보딩 상태를 반환합니다.

```
aws observabilityadmin get-telemetry-evaluation-status
```

출력:

```
{
    "Status": "RUNNING"
}
```

자세한 내용은 Amazon CloudWatch 사용 설명서의 CloudWatch 원격 측정 구성 감사를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetTelemetryEvaluationStatus를 참조하세요.

list-resource-telemetry-for-organization

다음 코드 예시는 `list-resource-telemetry-for-organization`의 사용 방법을 보여 줍니다.

AWS CLI

조직의 원격 측정 구성을 검색하려면

다음 `list-resource-telemetry-for-organization` 예제에서는 원격 측정 구성에서 지원하는 AWS 리소스에 대한 조직의 원격 측정 구성 목록을 반환합니다.

```
aws observabilityadmin list-resource-telemetry-for-organization \
    --resource-types  AWS::EC2::Instance
```

출력:

```
{
    "TelemetryConfigurations": [
        {
            "AccountIdentifier": "111111111111",
            "TelemetryConfigurationState": {
                "Logs": "NotApplicable",
                "Metrics": "Disabled",
                "Traces": "NotApplicable"
            },
            "ResourceType": "AWS::EC2::Instance",
            "ResourceIdentifier": "i-a166400b",
```

```
            "ResourceTags": {
                "Name": "dev"
            },
            "LastUpdateTimeStamp": 1733168548521
        },
        {
            "AccountIdentifier": "222222222222",
            "TelemetryConfigurationState": {
                "Logs": "NotApplicable",
                "Metrics": "Disabled",
                "Traces": "NotApplicable"
            },
            "ResourceType": "AWS::EC2::Instance",
            "ResourceIdentifier": "i-b188560f",
            "ResourceTags": {
                "Name": "apache"
            },
            "LastUpdateTimeStamp": 1732744260182
        }
    ]
}
```

자세한 내용은 Amazon CloudWatch 사용 설명서의 CloudWatch 원격 측정 구성 감사를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListResourceTelemetryForOrganization을 참조하세요.

list-resource-telemetry

다음 코드 예시는 `list-resource-telemetry`의 사용 방법을 보여 줍니다.

AWS CLI

계정의 원격 측정 구성을 검색하려면

다음 `list-resource-telemetry` 예제에서는 지정된 계정의 원격 측정 구성에서 지원하는 AWS 리소스에 대한 원격 측정 구성 목록을 반환합니다.

```
aws observabilityadmin list-resource-telemetry \
    --resource-types AWS::EC2::Instance
```

출력:

```
{
    "TelemetryConfigurations": [
        {
            "AccountIdentifier": "111111111111",
            "TelemetryConfigurationState": {
                "Logs": "NotApplicable",
                "Metrics": "Disabled",
                "Traces": "NotApplicable"
            },
            "ResourceType": "AWS::EC2::Instance",
            "ResourceIdentifier": "i-0e979d278b040f856",
            "ResourceTags": {
                "Name": "apache"
            },
            "LastUpdateTimeStamp": 1732744260182
        }
    ]
}
```

자세한 내용은 Amazon CloudWatch 사용 설명서의 CloudWatch 원격 측정 구성 감사를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListResourceTelemetry 섹션을 참조하세요.

`start-telemetry-evaluation-for-organization`

다음 코드 예시는 `start-telemetry-evaluation-for-organization`의 사용 방법을 보여 줍니다.

AWS CLI

원격 측정 구성 기능을 활성화하려면

다음 `start-telemetry-evaluation-for-organization` 예제에서는 조직의 원격 측정 구성 기능을 활성화합니다.

```
aws observabilityadmin start-telemetry-evaluation-for-organization
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon CloudWatch 사용 설명서의 CloudWatch 원격 측정 감사 켜기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [StartTelemetryEvaluationForOrganization](#) 섹션을 참조하세요.

`start-telemetry-evaluation`

다음 코드 예시는 `start-telemetry-evaluation`의 사용 방법을 보여 줍니다.

AWS CLI

원격 측정 구성 기능을 활성화하려면

다음 `start-telemetry-evaluation` 예제에서는 지정된 계정에서 원격 측정 구성 기능을 활성화합니다.

```
aws observabilityadmin start-telemetry-evaluation
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 [Amazon CloudWatch 사용 설명서](#)의 CloudWatch 원격 측정 감사 켜기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [StartTelemetryEvaluation](#)을 참조하세요.

`stop-telemetry-evaluation-for-organization`

다음 코드 예시는 `stop-telemetry-evaluation-for-organization`의 사용 방법을 보여 줍니다.

AWS CLI

원격 측정 구성 기능을 비활성화하려면

다음 `stop-telemetry-evaluation-for-organization` 예제에서는 조직의 원격 측정 구성 기능을 비활성화합니다.

```
aws observabilityadmin stop-telemetry-evaluation-for-organization
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon CloudWatch 사용 설명서의 [CloudWatch 원격 측정 감사 끄기](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StopTelemetryEvaluationForOrganization을 참조하세요.

stop-telemetry-evaluation

다음 코드 예시는 `stop-telemetry-evaluation`의 사용 방법을 보여 줍니다.

AWS CLI

원격 측정 구성 기능을 비활성화하려면

다음 `stop-telemetry-evaluation` 예제에서는 지정된 계정에서 원격 측정 구성 기능을 비활성화합니다.

```
aws observabilityadmin stop-telemetry-evaluation
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon CloudWatch 사용 설명서의 CloudWatch 원격 측정 감사 끄기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StopTelemetryEvaluation을 참조하세요.

AWS CLI를 사용한 CloudWatch Synthetics 예제

다음 코드 예제에서는 CloudWatch Synthetics에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여 줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

associate-resource

다음 코드 예시는 `associate-resource`의 사용 방법을 보여 줍니다.

AWS CLI

카나리를 그룹과 연결하려면

다음 `associate-resource` 예제에서는 카나리를 demo_group이라는 그룹과 연결합니다.

```
aws synthetics associate-resource \
    --group-identifier demo_group \
    --resource-arn arn:aws:synthetics:us-east-1:123456789012:canary:demo_canary
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon CloudWatch 사용 설명서의 Synthetic 모니터링(카나리)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AssociateResource를 참조하세요.

`create-canary`

다음 코드 예시는 `create-canary`의 사용 방법을 보여 줍니다.

AWS CLI

카나리를 생성하려면

다음 `create-canary` 예제에서는 demo_canary라는 카나리를 생성합니다.

```
aws synthetics create-canary \
    --name demo_canary \
    --code '{"S3Bucket": "artifacts3bucket", "S3Key":"demo_canary.zip", "Handler":
 "index.lambda_handler"}' \
    --artifact-s3-location s3://amzn-s3-demo-bucket/demo_canary.zip \
    --execution-role-arn arn:aws:iam::123456789012:role/demo_canary_role \
    --schedule Expression="rate(10 minutes)" \
    --runtime-version syn-nodejs-puppeteer-9.1
```

출력:

```
{
    "Canary": {
        "Id": "a1b2c3d4-5678-90ab-cdef-example11111",
        "Name": "demo_canary",
        "Code": {
            "Handler": "index.lambda_handler"
```

```
        },
        "ExecutionRoleArn": "arn:aws:iam::123456789012:role/demo_canary_role",
        "Schedule": {
            "Expression": "rate(10 minutes)",
            "DurationInSeconds": 0
        },
        "RunConfig": {
            "TimeoutInSeconds": 600,
            "MemoryInMB": 1000,
            "ActiveTracing": false
        },
        "SuccessRetentionPeriodInDays": 31,
        "FailureRetentionPeriodInDays": 31,
        "Status": {
            "State": "CREATING",
            "StateReasonCode": "CREATE_PENDING"
        },
        "Timeline": {
            "Created": "2024-10-15T19:03:08.826000+05:30",
            "LastModified": "2024-10-15T19:03:08.826000+05:30"
        },
        "ArtifactS3Location": "amzn-s3-demo-bucket/demo_canary.zip",
        "RuntimeVersion": "syn-nodejs-puppeteer-9.1",
        "Tags": {}
    }
}
```

자세한 내용은 Amazon CloudWatch 사용 설명서의 Synthetic 모니터링(카나리)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateCanary를 참조하세요.

create-group

다음 코드 예시는 `create-group`의 사용 방법을 보여 줍니다.

AWS CLI

그룹 생성

다음 `create-group` 예시에서는 demo_group이라는 그룹을 생성합니다.

```
aws synthetics create-group \
    --name demo_group
```

출력:

```
{
    "Group": {
        "Id": "example123",
        "Name": "demo_group",
        "Arn": "arn:aws:synthetics:us-east-1:123456789012:group:example123",
        "Tags": {},
        "CreatedTime": "2024-10-15T14:47:23.811000+05:30",
        "LastModifiedTime": "2024-10-15T14:47:23.811000+05:30"
    }
}
```

자세한 내용은 Amazon CloudWatch 사용 설명서의 Synthetic 모니터링(카나리)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateGroup을 참조하세요.

delete-canary

다음 코드 예시는 delete-canary의 사용 방법을 보여 줍니다.

AWS CLI

카나리를 영구적으로 삭제하려면

다음 delete-canary 예제에서는 demo_canary라는 카나리를 삭제합니다.

```
aws synthetics delete-canary \
    --name demo_canary
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon CloudWatch 사용 설명서의 Synthetic 모니터링(카나리)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteCanary를 참조하세요.

delete-group

다음 코드 예시는 delete-group의 사용 방법을 보여 줍니다.

AWS CLI

그룹 삭제

다음 delete-group 예제에서는 demo_group이라는 그룹을 삭제합니다.

```
aws synthetics delete-group \
    --group-identifier demo_group
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon CloudWatch 사용 설명서의 Synthetic 모니터링(카나리)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteGroup을 참조하세요.

describe-canaries-last-run

다음 코드 예시는 describe-canaries-last-run의 사용 방법을 보여 줍니다.

AWS CLI

각 카나리의 최신 실행 정보를 보려면

다음 describe-canaries-last-run 예제에서는 생성한 각 카나리의 가장 최근 실행을 반환합니다.

```
aws synthetics describe-canaries-last-run
```

출력:

```
{
    "CanariesLastRun": [
        {
            "CanaryName": "demo_canary",
            "LastRun": {
                "Id": "a1b2c3d4-5678-90ab-cdef-example11111",
                "Name": "demo_canary",
                "Status": {
                    "State": "PASSED",
                    "StateReason": "",
                    "StateReasonCode": ""
                },
                "Timeline": {
                    "Started": "2024-10-15T19:20:39.691000+05:30",
                    "Completed": "2024-10-15T19:20:58.211000+05:30"
```

```
                },
                "ArtifactS3Location": "cw-syn-results-123456789012-us-east-1/canary/
us-east-1/demo_canary-abc-example1234/2024/10/15/13/50-39-690"
            }
        }
    ]
}
```

자세한 내용은 Amazon CloudWatch 사용 설명서의 Synthetic 모니터링(카나리)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeCanariesLastRun 섹션을 참조하세요.

describe-canaries

다음 코드 예시는 describe-canaries의 사용 방법을 보여 줍니다.

AWS CLI

계정의 카나리를 나열하려면

다음 describe-canaries 예제에서는 계정의 카나리에 대한 세부 정보를 나열합니다.

```
aws synthetics describe-canaries
```

출력:

```
{
    "Canaries": [
        {
            "Id": "a1b2c3d4-5678-90ab-cdef-example11111",
            "Name": "demo_canary",
            "Code": {
                "SourceLocationArn": "arn:aws:lambda:us-
east-1:123456789012:layer:cwsyn-demo_canary-a1b2c3d4-5678-90ab-cdef-
example11111b8:1",
                "Handler": "pageLoadBlueprint.handler"
            },
            "ExecutionRoleArn": "arn:aws:iam::123456789012:role/service-role/
CloudWatchSyntheticsRole-demo_canary-a12-a123bc456789",
            "Schedule": {
                "Expression": "rate(5 minutes)",
                "DurationInSeconds": 0
```

```
            },
            "RunConfig": {
                "TimeoutInSeconds": 300,
                "MemoryInMB": 1000,
                "ActiveTracing": false
            },
            "SuccessRetentionPeriodInDays": 31,
            "FailureRetentionPeriodInDays": 31,
            "Status": {
            "State": "RUNNING"
            },
            "Timeline": {
                "Created": "2024-10-15T18:55:15.168000+05:30",
                "LastModified": "2024-10-15T18:55:40.540000+05:30",
                "LastStarted": "2024-10-15T18:55:40.540000+05:30"
            },
            "ArtifactS3Location": "cw-syn-results-123456789012-us-east-1/canary/us-
east-1/demo_canary-a12-a123bc456789",
            "EngineArn": "arn:aws:lambda:us-east-1:123456789012:function:cwsyn-
demo_canary-a1b2c3d4-5678-90ab-cdef-example111118:1",
            "RuntimeVersion": "syn-nodejs-puppeteer-9.1",
            "Tags": {
                "blueprint": "heartbeat"
            }
        }
    ]
}
```

자세한 내용은 Amazon CloudWatch 사용 설명서의 Synthetic 모니터링(카나리)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeCanaries를 참조하세요.

describe-runtime-versions

다음 코드 예시는 describe-runtime-versions의 사용 방법을 보여 줍니다.

AWS CLI

합성 카나리 런타임 버전 목록을 반환하려면

다음 describe-runtime-versions 예제에서는 합성 카나리 런타임 버전 목록을 반환합니다.

```
aws synthetics describe-runtime-versions
```

출력:


```
{
    "RuntimeVersions": [
        {
            "VersionName": "syn-nodejs-puppeteer-9.1",
            "Description": "Security fixes and bug fix for date range error in har.
 Dependencies: Node JS 20.x, Puppeteer-core 22.12.1, Chromium 126.0.6478.126",
            "ReleaseDate": "2024-10-02T05:30:00+05:30"
        },
        {
            "VersionName": "syn-nodejs-puppeteer-9.0",
            "Description": "Upgraded Chromium and Puppeteer. Dependencies: Node JS
 20.x, Puppeteer-core 22.12.1, Chromium 126.0.6478.126",
            "ReleaseDate": "2024-07-22T05:30:00+05:30"
        },
        {
            "VersionName": "syn-nodejs-puppeteer-8.0",
            "Description": "Upgraded Chromium and Puppeteer. Dependencies: Node JS
 20.x, Puppeteer-core 22.10.0, Chromium 125.0.6422.112",
            "ReleaseDate": "2024-06-21T05:30:00+05:30"
        },
        {
            "VersionName": "syn-nodejs-puppeteer-7.0",
            "Description": "Upgraded Chromium and Puppeteer. Dependencies: Node JS
 18.x, Puppeteer-core 21.9.0, Chromium 121.0.6167.139",
            "ReleaseDate": "2024-03-08T05:30:00+05:30"
            },
        {
            "VersionName": "syn-nodejs-puppeteer-6.2",
            "Description": "Updated shared libraries for Chromium and added
 ephemeral storage monitoring. Dependencies: Node JS 18.x, Puppeteer-core 19.7.0,
 Chromium 111.0.5563.146",
            "ReleaseDate": "2024-02-02T05:30:00+05:30"
        },
        {
            "VersionName": "syn-nodejs-puppeteer-6.1",
            "Description": "Added puppeteer launch retry. Dependencies: Node JS
 18.x, Puppeteer-core 19.7.0, Chromium 111.0.5563.146",
            "ReleaseDate": "2023-11-13T05:30:00+05:30",
            "DeprecationDate": "2024-03-08T13:30:00+05:30"
        },
        {
            "VersionName": "syn-nodejs-puppeteer-6.0",
```

```
            "Description": "Reduced X-Ray traces of a canary run, improved duration
 metric and upgraded to NodeJS 18.x. Dependencies: Node JS 18.x, Puppeteer-core
 19.7.0, Chromium 111.0.5563.146",
            "ReleaseDate": "2023-09-15T05:30:00+05:30",
            "DeprecationDate": "2024-03-08T13:30:00+05:30"
        },
        {
            "VersionName": "syn-nodejs-puppeteer-5.2",
            "Description": "Updated shared libraries for Chromium. Dependencies:
 Node JS 16.x, Puppeteer-core 19.7.0, Chromium 111.0.5563.146",
            "ReleaseDate": "2024-02-01T05:30:00+05:30"
        },
        {
            "VersionName": "syn-nodejs-puppeteer-5.1",
            "Description": "Fixes a bug about missing request headers in har.
 Dependencies: Node JS 16.x, Puppeteer-core 19.7.0, Chromium 111.0.5563.146",
            "ReleaseDate": "2023-08-09T05:30:00+05:30",
            "DeprecationDate": "2024-03-08T13:30:00+05:30"
        },
        {
            "VersionName": "syn-nodejs-puppeteer-5.0",
            "Description": "Upgraded Puppeteer and Chromium. Dependencies: Node JS
 16.x, Puppeteer-core 19.7.0, Chromium 111.0.5563.146",
            "ReleaseDate": "2023-07-21T05:30:00+05:30",
            "DeprecationDate": "2024-03-08T13:30:00+05:30"
        },
        {
            "VersionName": "syn-nodejs-puppeteer-4.0",
            "Description": "Upgraded to NodeJS 16.x. Dependencies: Node JS 16.x,
 Puppeteer-core 5.5.0, Chromium 92.0.4512.0",
            "ReleaseDate": "2023-05-01T05:30:00+05:30",
            "DeprecationDate": "2024-03-08T13:30:00+05:30"
        }
    ]
}
```

자세한 내용은 Amazon CloudWatch 사용 설명서의 Synthetic 모니터링(카나리)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DDescribeRuntimeVersions 섹션을 참조하세요.

`disassociate-resource`

다음 코드 예시는 `disassociate-resource`의 사용 방법을 보여 줍니다.

AWS CLI

그룹에서 카나리를 제거하려면

다음 `disassociate-resource` 예제에서는 demo_group이라는 그룹에서 카나리를 제거합니다.

```
aws synthetics disassociate-resource \
    --group-identifier demo_group \
    --resource-arn arn:aws:synthetics:us-east-1:123456789012:canary:demo_canary
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon CloudWatch 사용 설명서의 Synthetic 모니터링(카나리)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisassociateResource를 참조하세요.

`get-canary-runs`

다음 코드 예시는 `get-canary-runs`의 사용 방법을 보여 줍니다.

AWS CLI

지정된 카나리에 대한 실행 목록을 검색하려면

다음 `get-canary-runs` 예제에서는 demo_canary라는 카나리의 실행 목록을 검색합니다.

```
aws synthetics get-canary-runs \
    --name demo_canary
```

출력:

```
{
    "CanaryRuns": [
        {
            "Id": "a1b2c3d4-5678-90ab-cdef-example11111",
            "Name": "demo_canary",
            "Status": {
                "State": "PASSED",
                "StateReason": "",
                "StateReasonCode": ""
            },
            "Timeline": {
```

```
                "Started": "2024-10-16T10:38:57.013000+05:30",
                "Completed": "2024-10-16T10:39:25.793000+05:30"
            },
            "ArtifactS3Location": "cw-syn-results-123456789012-us-east-1/canary/us-
east-1/demo_canary-abc-example1234/2024/10/15/13/50-39-690"
        }
    ]
}
```

자세한 내용은 Amazon CloudWatch 사용 설명서의 Synthetic 모니터링(카나리)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetCanaryRuns를 참조하세요.

get-canary

다음 코드 예시는 `get-canary`의 사용 방법을 보여 줍니다.

AWS CLI

하나의 카나리에 대한 전체 정보를 검색하려면

다음 `get-canary` 예제에서는 `demo_canary`라는 카나리에 대한 전체 정보를 검색합니다.

```
aws synthetics get-canary \
    --name demo_canary
```

출력:

```
{
    "Canary": {
        "Id": "a1b2c3d4-5678-90ab-cdef-example11111",
        "Name": "demo_canary",
        "Code": {
            "SourceLocationArn": "arn:aws:lambda:us-east-1:123456789012:layer:cwsyn-
demo_canary-a1b2c3d4-5678-90ab-cdef-example111118:1",
            "Handler": "pageLoadBlueprint.handler"
        },
        "ExecutionRoleArn": "arn:aws:iam::123456789012:role/demo_canary_role",
        "Schedule": {
            "Expression": "rate(10 minutes)",
            "DurationInSeconds": 0
        },
        "RunConfig": {
```

```
            "TimeoutInSeconds": 300,
            "MemoryInMB": 1000,
            "ActiveTracing": false
        },
        "SuccessRetentionPeriodInDays": 31,
        "FailureRetentionPeriodInDays": 31,
        "Status": {
            "State": "RUNNING"
        },
        "Timeline": {
            "Created": "2024-10-15T18:55:15.168000+05:30",
            "LastModified": "2024-10-15T18:55:40.540000+05:30",
            "LastStarted": "2024-10-15T18:55:40.540000+05:30"
        },
        "ArtifactS3Location": "cw-syn-results-123456789012-us-east-1/canary/us-
east-1/demo_canary-a12-a123bc456789",
        "EngineArn": "arn:aws:lambda:us-east-1:123456789012:function:cwsyn-
demo_canary-a1b2c3d4-5678-90ab-cdef-example111118:1",
        "RuntimeVersion": "syn-nodejs-puppeteer-9.1",
        "Tags": {
            "blueprint": "heartbeat"
        }
    }
}
```

자세한 내용은 Amazon CloudWatch 사용 설명서의 Synthetic 모니터링(카나리)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetCanary를 참조하세요.

get-group

다음 코드 예시는 get-group의 사용 방법을 보여 줍니다.

AWS CLI

하나의 그룹에 대한 정보를 반환하려면

다음 get-group 예제에서는 demo_group이라는 그룹에 대한 정보를 반환합니다.

```
aws synthetics get-group \
    --group-identifier demo_group
```

출력:

```
{
    "Group": {
        "Id": "example123",
        "Name": "demo_group",
        "Arn": "arn:aws:synthetics:us-east-1:123456789012:group:example123",
        "Tags": {},
        "CreatedTime": "2024-10-15T14:47:23.811000+05:30",
        "LastModifiedTime": "2024-10-15T14:47:23.811000+05:30"
    }
}
```

자세한 내용은 Amazon CloudWatch 사용 설명서의 Synthetic 모니터링(카나리)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetGroup을 참조하세요.

list-associated-groups

다음 코드 예시는 `list-associated-groups`의 사용 방법을 보여 줍니다.

AWS CLI

그룹 목록을 반환하려면

다음 `list-associated-groups` 예제에서는 demo_canary라는 카나리와 연결된 그룹의 목록을 반환합니다.

```
aws synthetics list-associated-groups \
    --resource-arn arn:aws:synthetics:us-east-1:123456789012:canary:demo_canary
```

출력:

```
{
    "Groups": [
        {
            "Id": "example123",
            "Name": "demo_group",
            "Arn": "arn:aws:synthetics:us-east-1:123456789012:group:example123"
        }
    ]
}
```

자세한 내용은 Amazon CloudWatch 사용 설명서의 Synthetic 모니터링(카나리)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListAssociatedGroups 섹션을 참조하세요.

list-group-resources

다음 코드 예시는 list-group-resources의 사용 방법을 보여 줍니다.

AWS CLI

지정된 그룹과 연결된 카나리의 ARN 목록을 반환하려면

다음 list-group-resources 예제에서는 demo_group이라는 그룹과 연결된 카나리의 ARN 목록을 반환합니다.

```
aws synthetics list-group-resources \
    --group-identifier demo_group
```

출력:

```
{
    "Resources": [
        "arn:aws:synthetics:us-east-1:123456789012:canary:demo_canary"
    ]
}
```

자세한 내용은 Amazon CloudWatch 사용 설명서의 Synthetic 모니터링(카나리)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListGroupResources를 참조하세요.

list-groups

다음 코드 예시는 list-groups의 사용 방법을 보여 줍니다.

AWS CLI

계정의 모든 그룹 목록을 반환하려면

다음 list-groups 예제에서는 계정의 모든 그룹 목록을 반환합니다.

```
aws synthetics list-groups
```

출력:

```
{
    "Groups": [
        {
            "Id": "example123",
            "Name": "demo_group",
            "Arn": "arn:aws:synthetics:us-east-1:123456789012:group:example123"
        }
    ]
}
```

자세한 내용은 Amazon CloudWatch 사용 설명서의 Synthetic 모니터링(카나리)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListGroups를 참조하세요.

list-tags-for-resource

다음 코드 예시는 list-tags-for-resource의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 카나리와 연결된 태그 표시

다음 list-tags-for-resource 예제에서는 demo_canary라는 카나리와 연결된 태그를 반환합니다.

```
aws synthetics list-tags-for-resource \
    --resource-arn arn:aws:synthetics:us-east-1:123456789012:canary:demo_canary
```

출력:

```
{
    "Tags": {
        "blueprint": "heartbeat"
    }
}
```

예제 2: 그룹과 연결된 태그 표시

다음 list-tags-for-resource 예제에서는 demo_group이라는 그룹과 연결된 태그를 반환합니다.

```
aws  synthetics list-tags-for-resource \
```

```
    --resource-arn arn:aws:synthetics:us-east-1:123456789012:group:example123
```

출력:

```
{
    "Tags": {
        "team": "Devops"
    }
}
```

자세한 내용은 Amazon CloudWatch 사용 설명서의 Synthetic 모니터링(카나리)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource를 참조하세요.

start-canary

다음 코드 예시는 start-canary의 사용 방법을 보여 줍니다.

AWS CLI

카나리를 실행하려면

다음 start-canary 예제에서는 demo_canary라는 카나리를 실행합니다.

```
aws synthetics start-canary \
    --name demo_canary
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon CloudWatch 사용 설명서의 Synthetic 모니터링(카나리)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartCanary 섹션을 참조하세요.

stop-canary

다음 코드 예시는 stop-canary의 사용 방법을 보여 줍니다.

AWS CLI

카나리를 중지하려면

다음 stop-canary 예제에서는 demo_canary라는 카나리를 중지합니다.

```
aws synthetics stop-canary \
    --name demo_canary
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon CloudWatch 사용 설명서의 Synthetic 모니터링(카나리)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StopCanary 섹션을 참조하세요.

tag-resource

다음 코드 예시는 `tag-resource`의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 카나리에 태그 할당

다음 `tag-resource` 예제에서는 `demo_canary`라는 카나리에 태그를 할당합니다.

```
aws synthetics tag-resource \
    --resource-arn arn:aws:synthetics:us-east-1:123456789012:canary:demo_canary \
    --tags blueprint=heartbeat
```

이 명령은 출력을 생성하지 않습니다.

예제 2: 그룹에 태그를 할당

다음 `tag-resource` 예제에서는 `demo_group`이라는 그룹에 태그를 할당합니다.

```
aws synthetics tag-resource \
    --resource-arn arn:aws:synthetics:us-east-1:123456789012:group:example123 \
    --tags team=Devops
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon CloudWatch 사용 설명서의 Synthetic 모니터링(카나리)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

untag-resource

다음 코드 예시는 `untag-resource`의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 카나리에서 태그 제거

다음 `untag-resource` 예제에서는 demo_canary라는 카나리에서 태그를 제거합니다.

```
aws synthetics untag-resource \
    --resource-arn arn:aws:synthetics:us-east-1:123456789012:canary:demo_canary \
    --tag-keys blueprint
```

이 명령은 출력을 생성하지 않습니다.

예제 2: 그룹에서 태그 제거

다음 `untag-resource` 예제에서는 demo_group이라는 그룹에서 태그를 제거합니다.

```
aws synthetics untag-resource \
    --resource-arn arn:aws:synthetics:us-east-1:123456789012:group:example123 \
    --tag-keys team
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon CloudWatch 사용 설명서의 Synthetic 모니터링(카나리)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

update-canary

다음 코드 예시는 `update-canary`의 사용 방법을 보여 줍니다.

AWS CLI

카나리를 업데이트하려면

다음 `update-canary` 예제에서는 demo_canary라는 카나리의 구성을 업데이트합니다.

```
aws synthetics update-canary \
    --name demo_canary \
    --schedule Expression="rate(15 minutes)"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon CloudWatch 사용 설명서의 Synthetic 모니터링(카나리)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateCanary를 참조하세요.

AWS CLI를 사용한 CodeArtifact 예제

다음 코드 예제에서는 CodeArtifact에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여 줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

associate-external-connection

다음 코드 예시는 `associate-external-connection`의 사용 방법을 보여 줍니다.

AWS CLI

리포지토리에 외부 연결을 추가하는 방법

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 CodeArtifac 사용 설명서의 외부 연결 추가를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AssociateExternalConnection 섹션을 참조하세요.

copy-package-versions

다음 코드 예시는 copy-package-versions의 사용 방법을 보여 줍니다.

AWS CLI

한 리포지토리에서 다른 리포지토리로 패키지 버전을 복사하는 방법

다음 copy-package-versions 명령은 test-package라는 패키지의 버전 4.0.0 및 5.0.0을 my-repo에서 test-repo로 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 CodeArtifac 사용 설명서의 리포지토리 간 패키지 복사를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CopyPackageVersions 섹션을 참조하세요.

create-domain

다음 코드 예시는 `create-domain`의 사용 방법을 보여 줍니다.

AWS CLI

도메인을 생성하려면

다음 `create-domain` 예제에서는 test-domain이라는 도메인을 만듭니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateDomain을 참조하세요.

`create-repository`

다음 코드 예시는 `create-repository`의 사용 방법을 보여 줍니다.

AWS CLI

리포지토리 생성

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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateRepository를 참조하세요.

delete-domain-permissions-policy

다음 코드 예시는 delete-domain-permissions-policy의 사용 방법을 보여 줍니다.

AWS CLI

도메인에서 권한 정책 문서를 삭제하는 방법

다음 delete-domain-permissions-policy 예제에서는 test-domain 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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteDomainPermissionsPolicy 섹션을 참조하세요.

delete-domain

다음 코드 예시는 delete-domain의 사용 방법을 보여 줍니다.

AWS CLI

도메인을 삭제하려면

다음 delete-domain 예제에서는 test-domain 도메인을 삭제합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteDomain을 참조하세요.

delete-package-versions

다음 코드 예시는 delete-package-versions의 사용 방법을 보여 줍니다.

AWS CLI

패키지 버전을 삭제하는 방법

다음 delete-package-versions 예제에서는 test-package 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.

- API 세부 정보는 AWS CLI 명령 참조의 DeletePackageVersions 섹션을 참조하세요.

delete-repository-permissions-policy

다음 코드 예시는 delete-repository-permissions-policy의 사용 방법을 보여 줍니다.

AWS CLI

리포지토리에서 권한 정책을 삭제하는 방법

다음 delete-repository-permissions-policy 예제에서는 test-repo 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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteRepositoryPermissionsPolicy 섹션을 참조하세요.

delete-repository

다음 코드 예시는 delete-repository의 사용 방법을 보여 줍니다.

AWS CLI

리포지토리 삭제

다음 delete-repository 예제에서는 test-domain 도메인의 test-repo 리포지토리를 삭제합니다.

```
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 AWSCodeArtifact 사용 안내서의 리포지토리 삭제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteRepository를 참조하세요.

`describe-domain`

다음 코드 예시는 `describe-domain`의 사용 방법을 보여 줍니다.

AWS CLI

도메인 정보를 가져오려면

다음 `describe-domain` 예제에서는 테스트 도메인이라는 도메인에 대한 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 개요을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDomain을 참조하세요.

describe-repository

다음 코드 예시는 describe-repository의 사용 방법을 보여 줍니다.

AWS CLI

리포지토리 관련 정보를 가져오기

다음 describe-repository 예제에서는 test-repo라는 리포지토리에 대한 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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeRepository 섹션을 참조하세요.

disassociate-external-connection

다음 코드 예시는 disassociate-external-connection의 사용 방법을 보여 줍니다.

AWS CLI

리포지토리에서 외부 연결을 제거하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 DisassociateExternalConnection 섹션을 참조하세요.

dispose-package-versions

다음 코드 예시는 dispose-package-versions의 사용 방법을 보여 줍니다.

AWS CLI

패키지 버전의 자산을 삭제하고 상태를 처리됨으로 설정하는 방법

다음 `dispose-package-versions` 예제에서는 test-package 버전 4.0.0의 애셋을 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 CodeArtifact에서의 패키지 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisposePackageVersions 섹션을 참조하세요.

get-authorization-token

다음 코드 예시는 `get-authorization-token`의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 GetAuthorizationToken 섹션을 참조하세요.

get-domain-permissions-policy

다음 코드 예시는 get-domain-permissions-policy의 사용 방법을 보여 줍니다.

AWS CLI

도메인에 대한 권한 정책 문서를 가져오는 방법

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 읽기]를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [GetDomainPermissionsPolicy] 섹션을 참조하세요.

get-package-version-asset

다음 코드 예시는 `get-package-version-asset`의 사용 방법을 보여 줍니다.

AWS CLI

패키지 버전에서 자산을 가져오는 방법

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 나열]을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [GetPackageVersionAsset] 섹션을 참조하세요.

get-package-version-readme

다음 코드 예시는 `get-package-version-readme`의 사용 방법을 보여 줍니다.

AWS CLI

패키지 버전의 Readme 파일을 가져오는 방법

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 readme 파일 보기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetPackageVersionReadme 섹션을 참조하세요.

get-repository-endpoint

다음 코드 예시는 `get-repository-endpoint`의 사용 방법을 보여 줍니다.

AWS CLI

리포지토리의 URL 엔드포인트를 가져오는 방법

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 CodeArtifac 사용 설명서의 [리포지토리에 연결](#)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [GetRepositoryEndpoint](#) 섹션을 참조하세요.

`get-repository-permissions-policy`

다음 코드 예시는 `get-repository-permissions-policy`의 사용 방법을 보여 줍니다.

AWS CLI

리포지토리에 대한 권한 정책 문서를 가져오는 방법

다음 `get-repository-permissions-policy` 예제에서는 test-repo라는 리포지토리에 연결된 권한 정책을 가져옵니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 GetRepositoryPermissionsPolicy 섹션을 참조하세요.

list-domains

다음 코드 예시는 list-domains의 사용 방법을 보여 줍니다.

AWS CLI

도메인 목록을 표시하려면

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 CodeArtifact에서의 도메인 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListDomains 섹션을 참조하세요.

list-package-version-assets

다음 코드 예시는 `list-package-version-assets`의 사용 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 나열](#)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ListPackageVersionAssets](#) 섹션을 참조하세요.

`list-package-version-dependencies`

다음 코드 예시는 `list-package-version-dependencies`의 사용 방법을 보여 줍니다.

AWS CLI

패키지 버전의 종속성을 보는 방법

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 종속성 보기 및 업데이트](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ListPackageVersionDependencies](#) 섹션을 참조하세요.

`list-package-versions`

다음 코드 예시는 `list-package-versions`의 사용 방법을 보여 줍니다.

AWS CLI

패키지의 패키지 버전을 나열하는 방법

다음 `list-package-versions` 예제에서는 `kind-of` 패키지의 패키지 버전 목록을 반환합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 ListPackageVersions를 참조하세요.

list-packages

다음 코드 예시는 list-packages의 사용 방법을 보여 줍니다.

AWS CLI

리포지토리의 패키지를 나열하는 방법

다음 list-packages 예제에서는 test-domain이라는 도메인의 test-repo라는 리포지토리에 있는 패키지를 나열합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 ListPackages 섹션을 참조하세요.

`list-repositories-in-domain`

다음 코드 예시는 `list-repositories-in-domain`의 사용 방법을 보여 줍니다.

AWS CLI

도메인에 있는 리포지토리 나열

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.

- API 세부 정보는 AWS CLI 명령 참조의 ListRepositoriesInDomain 섹션을 참조하세요.

list-repositories

다음 코드 예시는 `list-repositories`의 사용 방법을 보여 줍니다.

AWS CLI

리포지토리를 나열하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 ListRepositories를 참조하세요.

login

다음 코드 예시는 `login`의 사용 방법을 보여 줍니다.

AWS CLI

로그인 명령을 사용하여 리포지토리에 대한 인증을 구성하는 방법

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 AWS CLI 시작하기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 Login 섹션을 참조하세요.

put-domain-permissions-policy

다음 코드 예시는 put-domain-permissions-policy의 사용 방법을 보여 줍니다.

AWS CLI

도메인에 권한 정책을 연결하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 PutDomainPermissionsPolicy 섹션을 참조하세요.

put-repository-permissions-policy

다음 코드 예시는 put-repository-permissions-policy의 사용 방법을 보여 줍니다.

AWS CLI

리포지토리에 권한 정책을 연결하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 PutRepositoryPermissionsPolicy 섹션을 참조하세요.

update-package-versions-status

다음 코드 예시는 update-package-versions-status의 사용 방법을 보여 줍니다.

AWS CLI

패키지 버전 상태를 업데이트하는 방법

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 업데이트](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [UpdatePackageVersionsStatus](#) 섹션을 참조하세요.

`update-repository`

다음 코드 예시는 `update-repository`의 사용 방법을 보여 줍니다.

AWS CLI

리포지토리를 업데이트하는 방법

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 CodeArtifac 사용 설명서의 리포지토리 구성 보기 또는 수정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateRepository 섹션을 참조하세요.

AWS CLI를 사용한 CodeBuild 예시

다음 코드 예시에서는 CodeBuild에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`batch-delete-builds`

다음 코드 예시에서는 `batch-delete-builds` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS CodeBuild에서 빌드를 삭제하는 방법

다음 `batch-delete-builds` 예시에서는 ID가 지정된 CodeBuild의 빌드를 삭제합니다.

```
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 Delete Builds (AWS CLI)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 BatchDeleteBuilds 섹션을 참조하세요.

batch-get-build-batches

다음 코드 예시에서는 `batch-get-build-batches` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS CodeBuild에서 빌드의 세부 정보를 보는 방법

다음 `batch-get-build-batches` 예시에서는 ID가 지정된 CodeBuild의 빌드 배치에 대한 정보를 가져옵니다.

```
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 Batch builds in AWS CodeBuild<https://docs.aws.amazon.com/codebuild/latest/userguide/batch-build.html>)__를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [BatchGetBuildBatches](#) 섹션을 참조하세요.

batch-get-builds

다음 코드 예시에서는 batch-get-builds 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS CodeBuild에서 빌드의 세부 정보를 보는 방법

다음 batch-get-builds 예시에서는 ID가 지정된 CodeBuild의 빌드에 대한 정보를 가져옵니다.

```
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 View Build Details (AWS CLI)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 BatchGetBuilds 섹션을 참조하세요.

batch-get-projects

다음 코드 예시에서는 `batch-get-projects` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS CodeBuild 빌드 프로젝트 이름 목록 가져오기

다음 batch-get-projects 예시에서는 이름별로 지정된 CodeBuild 빌드 프로젝트 목록을 가져옵니다.

```
aws codebuild batch-get-projects --names codebuild-demo-project codebuild-demo-
project2 my-other-demo-project
```

다음 출력에서는 projectsNotFound 배열에 지정되었지만 찾을 수 없는 모든 빌드 프로젝트 이름이 나열됩니다. 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 View a Build Project's Details (AWS CLI)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 BatchGetProjects 섹션을 참조하세요.

batch-get-report-groups

다음 코드 예시에서는 `batch-get-report-groups` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS CodeBuild에서 하나 이상의 보고서 그룹에 대한 정보를 가져오는 방법

다음 `batch-get-report-groups` 예시에서는 ARN이 지정된 보고서 그룹에 대한 정보를 검색합니다.

```
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 Working with report groups를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 BatchGetReportGroups 섹션을 참조하세요.

batch-get-reports

다음 코드 예시에서는 batch-get-reports 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS CodeBuild에서 하나 이상의 보고서에 대한 정보를 가져오는 방법

다음 batch-get-reports 예시에서는 ARN이 지정된 보고서에 대한 정보를 검색합니다.

```
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 Working with reports를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 BatchGetReports 섹션을 참조하세요.

create-project

다음 코드 예시에서는 `create-project` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

예 1: AWS CodeBuild 빌드 프로젝트를 생성하는 방법

다음 `create-project` 예시에서는 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
    }
}
```

예 2: 파라미터에 대한 JSON 입력 파일을 사용하여 AWS CodeBuild 빌드 프로젝트를 생성하는 방법

다음 `create-project` 예시에서는 필요한 파라미터를 모두 JSON 입력 파일에 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 파일 `create-project.json`에는 다음 콘텐츠가 포함되어 있습니다.

```
{
    "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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateProject 섹션을 참조하세요.

`create-report-group`

다음 코드 예시에서는 `create-report-group` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS CodeBuild에서 보고서 그룹 생성

다음 `create-report-group` 예시에서는 새 보고서 그룹을 만듭니다.

```
aws codebuild create-report-group \
    --cli-input-json file://create-report-group-source.json
```

create-report-group-source.json의 콘텐츠:

```
{
    "name": "cli-created-report-group",
    "type": "TEST",
    "exportConfig": {
        "exportConfigType": "S3",
        "s3Destination": {
            "bucket": "amzn-s3-demo-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": "amzn-s3-demo-bucket",
                "path": "",
                "packaging": "ZIP",
                "encryptionDisabled": true
            }
        },
        "created": 1602020026.775,
        "lastModified": 1602020026.775
    }
}
```

자세한 내용은 AWS CodeBuild 사용 설명서의 Working with report groups를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateReportGroup 섹션을 참조하세요.

`create-webhook`

다음 코드 예시에서는 `create-webhook` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS CodeBuild 프로젝트에 대한 웹후크 필터를 생성하는 방법

다음 `create-webhook` 예시에서는 두 개의 필터 그룹이 있는 이름이 `my-project`인 CodeBuild 프로젝트의 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 `^refs/heads/myBranch$`와 일치하지 않는 Git 참조 이름을 가진 브랜치에 대한 푸시 요청을 지정합니다.

```
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 Filter GitHub Webhook Events (SDK)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateWebhook 섹션을 참조하세요.

delete-build-batch

다음 코드 예시에서는 delete-build-batch 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS CodeBuild.

다음 delete-build-batch 예시에서는 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 Batch builds in AWS CodeBuild를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteBuildBatch 섹션을 참조하세요.

delete-project

다음 코드 예시에서는 delete-project 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS CodeBuild 빌드 프로젝트를 삭제하는 방법

다음 delete-project 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteProject 섹션을 참조하세요.

delete-report-group

다음 코드 예시에서는 delete-report-group 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS CodeBuild 에서 보고서 그룹을 삭제하는 방법

다음 delete-report-group 예시에서는 ARN이 지정된 보고서 그룹을 삭제합니다.

```
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 Working with report groups를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteReportGroup 섹션을 참조하세요.

delete-report

다음 코드 예시에서는 delete-report 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS CodeBuild 에서 보고서를 삭제하는 방법

다음 `delete-report` 예시에서는 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 Working with reports를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteReport 섹션을 참조하세요.

`delete-source-credentials`

다음 코드 예시에서는 `delete-source-credentials` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

소스 공급자와의 연결 해제 및 해당 액세스 토큰 제거

다음 `delete-source-credentials` 예시에서는 소스 공급자와의 연결을 해제하고 토큰을 제거합니다. 소스 공급자에 연결하는 데 사용되는 소스 자격 증명의 ARN에 따라 어떤 소스 자격 증명을 사용할지 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 Connect Source Providers with Access Tokens (CLI)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteSourceCredentials 섹션을 참조하세요.

delete-webhook

다음 코드 예시에서는 delete-webhook 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS CodeBuild 프로젝트에서 웹후크 필터를 삭제하는 방법

다음 delete-webhook 예시에서는 지정된 CodeBuild 프로젝트에서 웹후크를 삭제합니다.

```
aws codebuild delete-webhook --project-name my-project
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS CodeBuild 사용 설명서의 Stop Running Builds Automatically (AWS CLI)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteWebhook 섹션을 참조하세요.

describe-code-coverages

다음 코드 예시에서는 describe-code-coverages 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS CodeBuild에서 코드 커버리지 테스트 결과에 대한 자세한 정보를 받는 방법

다음 describe-code-coverages 예시에서는 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 Code coverage reports를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeCodeCoverages 섹션을 참조하세요.

describe-test-cases

다음 코드 예시에서는 `describe-test-cases` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS CodeBuild 에서 테스트 사례에 대한 자세한 정보를 가져오는 방법

다음 `describe-test-cases` 예시에서는 지정된 보고서의 테스트 케이스에 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 Working with test reporting in AWS CodeBuild를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeTestCases 섹션을 참조하세요.

`import-source-credentials`

다음 코드 예시에서는 `import-source-credentials` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

소스 공급자에 대한 자격 증명을 가져와서 AWS CodeBuild 사용자를 소스 공급자에 연결.

다음 `import-source-credentials` 예시에서는 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 Connect Source Providers with Access Tokens (CLI)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ImportSourceCredentials 섹션을 참조하세요.

`invalidate-project-cache`

다음 코드 예시에서는 `invalidate-project-cache` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS CodeBuild 빌드 프로젝트의 캐시를 재설정하는 방법

다음 `invalidate-project-cache` 예시에서는 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 Build Caching in CodeBuild를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 InvalidateProjectCache 섹션을 참조하세요.

`list-build-batches-for-project`

다음 코드 예시에서는 `list-build-batches-for-project` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS CodeBuild에서 특정 빌드 프로젝트의 배치 빌드를 나열하는 방법

다음 `list-build-batches-for-project` 예시에서는 지정된 프로젝트에 대한 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 Batch builds in AWS CodeBuild를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListBuildBatchesForProject 섹션을 참조하세요.

list-build-batches

다음 코드 예시에서는 `list-build-batches` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS CodeBuild에서 배치 빌드를 나열하는 방법

다음 `list-build-batches` 예시에서는 현재 계정에 대한 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 Batch builds in AWS CodeBuild<https://docs.aws.amazon.com/codebuild/latest/userguide/batch-build.html>)__를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListBuildBatches 섹션을 참조하세요.

list-builds-for-project

다음 코드 예시에서는 list-builds-for-project 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS CodeBuild 빌드 프로젝트의 빌드 목록을 보는 방법

다음 list-builds-for-project 예시에서는 지정된 CodeBuild 빌드 프로젝트의 빌드 ID를 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 View a List of Build IDs for a Build Project (AWS CLI)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListBuildsForProject 섹션을 참조하세요.

list-builds

다음 코드 예시에서는 list-builds 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS CodeBuild 빌드 IDs.

다음 `list-builds` 예제에서는 오름차순으로 정렬된 CodeBuild ID 목록을 가져옵니다.

```
aws codebuild list-builds --sort-order ASCENDING
```

출력에는 사용 가능한 출력이 더 있음을 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 ID 목록 보기(AWS CLI)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListBuilds 섹션을 참조하세요.

list-curated-environment-images

다음 코드 예시에서는 `list-curated-environment-images` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

빌드에 사용할 수 있는 AWS CodeBuild에서 관리하는 Docker 이미지 목록 가져오기

다음 `list-curated-environment-images` 예시에서는 빌드에 사용할 수 있는 CodeBuild에서 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 Docker Images Provided by CodeBuild를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListCuratedEnvironmentImages 섹션을 참조하세요.

list-projects

다음 코드 예시에서는 `list-projects` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS CodeBuild 빌드 프로젝트 이름 목록 가져오기

다음 `list-projects` 예제에서는 이름을 기준으로 정렬된 CodeBuild 빌드 프로젝트 목록을 오름차순으로 가져옵니다.

```
aws codebuild list-projects --sort-by NAME --sort-order ASCENDING
```

출력에는 사용 가능한 출력이 더 있음을 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListProjects 섹션을 참조하세요.

`list-report-groups`

다음 코드 예시에서는 `list-report-groups` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS CodeBuild에서 보고서 그룹 ARNs 목록을 가져오는 방법

다음 `list-report-groups` 예시에서는 리전의 계정에 대한 보고서 그룹 ARN을 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 Working with report groups를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListReportGroups 섹션을 참조하세요.

list-reports-for-report-group

다음 코드 예시에서는 `list-reports-for-report-group` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS CodeBuild에서 보고서 그룹에서 보고서 목록을 가져오는 방법

다음 `list-report-for-report-groups` 예시에서는 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 Working with report groups를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListReportsForReportGroup 섹션을 참조하세요.

list-reports

다음 코드 예시에서는 `list-reports` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS CodeBuild에서 현재 계정에 대한 보고서 목록을 가져오는 방법

다음 `list-reports` 예시에서는 현재 계정에 대한 보고서의 ARN을 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 Working with reports를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListReports 섹션을 참조하세요.

`list-shared-projects`

다음 코드 예시에서는 `list-shared-projects` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS CodeBuild에서 공유 프로젝트를 나열하는 방법

다음 `list-shared-projects` 예시에서는 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 Working with shared projects를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListSharedProjects 섹션을 참조하세요.

`list-shared-report-groups`

다음 코드 예시에서는 `list-shared-report-groups` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS CodeBuild에서 공유 보고서 그룹 ARNs 목록을 가져오는 방법

다음 `list-shared-report-groups` 예시에서는 리전의 계정에 대한 보고서 그룹 ARN을 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 Working with report groups를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListSharedReportGroups 섹션을 참조하세요.

`list-source-credentials`

다음 코드 예시에서는 `list-source-credentials` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

sourceCredentialsObjects 목록을 보는 방법

다음 `list-source-credentials` 예시에서는 하나의 Bitbucket 계정과 하나의 GitHub 계정에 연결된 AWS 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 Connect Source Providers with Access Tokens (CLI)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListSourceCredentials 섹션을 참조하세요.

retry-build-batch

다음 코드 예시에서는 `retry-build-batch` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS CodeBuild에서 실패한 배치 빌드를 다시 시도하는 방법

다음 `retry-build-batch` 예시에서는 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 Batch builds in AWS CodeBuild를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RetryBuildBatch 섹션을 참조하세요.

retry-build

다음 코드 예시에서는 retry-build 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS CodeBuild에서 실패한 빌드를 다시 시도하는 방법

다음 retry-build 예시에서는 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 Batch builds in AWS CodeBuild를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RetryBuild 섹션을 참조하세요.

`start-build-batch`

다음 코드 예시에서는 `start-build-batch` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS CodeBuild에서 배치 빌드를 시작하는 방법

다음 `start-build-batch` 예시에서는 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 Batch builds in AWS CodeBuild를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartBuildBatch 섹션을 참조하세요.

start-build

다음 코드 예시에서는 `start-build` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS CodeBuild 빌드 프로젝트의 빌드 실행을 시작하는 방법

다음 `start-build` 예제에서는 지정된 CodeBuild 프로젝트의 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.

- API 세부 정보는 AWS CLI 명령 참조의 StartBuild 섹션을 참조하세요.

stop-build-batch

다음 코드 예시에서는 stop-build-batch 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS CodeBuild에서 진행 중인 배치 빌드를 중지하는 방법

다음 stop-build-batch 예시에서는 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 Batch builds in AWS CodeBuild를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StopBuildBatch 섹션을 참조하세요.

stop-build

다음 코드 예시에서는 `stop-build` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS CodeBuild 빌드 프로젝트의 빌드를 중지하는 방법

다음 `stop-build` 예시에서는 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 Stop a Build (AWS CLI)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StopBuild 섹션을 참조하세요.

`update-project`

다음 코드 예시에서는 `update-project` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS CodeBuild 빌드 프로젝트의 설정을 변경하는 방법

다음 `update-project` 예시에서는 my-demo-project라는 지정된 CodeBuild 빌드 프로젝트의 설정을 변경합니다.

```
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 Change a Build Project's Settings (AWS CLI)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateProject 섹션을 참조하세요.

update-report-group

다음 코드 예시에서는 update-report-group 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS CodeBuild에서 보고서 그룹을 업데이트하는 방법

다음 update-report-group 예시에서는 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 Working with report groups를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateReportGroup 섹션을 참조하세요.

update-webhook

다음 코드 예시에서는 update-webhook 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS CodeBuild 프로젝트의 웹후크를 업데이트하는 방법

다음 update-webhook 예시에서는 지정된 CodeBuild 프로젝트의 웹후크를 두 개의 필터 그룹으로 업데이트합니다. --rotate-secret 파라미터는 코드 변경이 빌드를 트리거할 때마다 GitHub가 프로젝트의 보안 키를 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 ^refs/heads/myBranch$와 일치하지 않는 Git 참조 이름을 가진 브랜치에 대한 푸시 요청을 지정합니다.

```
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 [Change a Build Project's Settings (AWS CLI)]를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateWebhook 섹션을 참조하세요.

AWS CLI를 사용한 CodeCommit 예시

다음 코드 예시에서는 CodeCommit에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

associate-approval-rule-template-with-repository

다음 코드 예시에서는 associate-approval-rule-template-with-repository의 사용 방법을 보여줍니다.

AWS CLI

승인 규칙 템플릿을 리포지토리와 연결

다음 associate-approval-rule-template-with-repository 예시는 지정된 승인 규칙 템플릿을 MyDemoRepo 리포지토리와 연결합니다.

```
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 [Associate an Approval Rule Template with a Repository](#) 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [AssociateApprovalRuleTemplateWithRepository](#) 섹션을 참조하세요.

batch-associate-approval-rule-template-with-repositories

다음 코드 예시에서는 batch-associate-approval-rule-template-with-repositories의 사용 방법을 보여줍니다.

AWS CLI

한 번의 작업으로 승인 규칙 템플릿을 여러 리포지토리와 연결

다음 batch-associate-approval-rule-template-with-repositories 예시에서는 지정된 승인 규칙 템플릿을 MyDemoRepo 및 MyOtherDemoRepo 리포지토리와 연결합니다.

참고: 승인 규칙 템플릿은 해당 템플릿이 생성된 AWS 리전에만 적용됩니다. 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 Associate an Approval Rule Template with a Repository 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 BatchAssociateApprovalRuleTemplateWithRepositories 섹션을 참조하세요.

batch-describe-merge-conflicts

다음 코드 예시에서는 `batch-describe-merge-conflicts`의 사용 방법을 보여줍니다.

AWS CLI

두 커밋 지정자 간의 병합에서 모든 파일 또는 파일 하위 세트의 병합 충돌에 대한 정보 가져오기

다음 `batch-describe-merge-conflicts` 예시에서는 `MyDemoRepo` 리포지토리에서 `THREE_WAY_MERGE` 전략을 사용하여 `feature-randomizationfeature` 소스 브랜치를 `main` 대상 브랜치와 병합할 때 병합 충돌을 확인합니다.

```
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 Resolve Conflicts in a Pull Request 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 BatchDescribeMergeConflicts 섹션을 참조하세요.

batch-disassociate-approval-rule-template-from-repositories

다음 코드 예시에서는 `batch-disassociate-approval-rule-template-from-repositories`의 사용 방법을 보여줍니다.

AWS CLI

한 번의 작업으로 여러 리포지토리에서 승인 규칙 템플릿 연결 해제

다음 `batch-disassociate-approval-rule-template-from-repositories` 예시는 지정된 승인 규칙 템플릿을 및 이라는 이름의 `MyDemoRepo` 및 `MyOtherDemoRepo` 리포지토리에서 연결 해제합니다.

```
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 Disassociate an Approval Rule Template 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 BatchDisassociateApprovalRuleTemplateFromRepositories 섹션을 참조하세요.

batch-get-commits

다음 코드 예시에서는 batch-get-commits의 사용 방법을 보여줍니다.

AWS CLI

여러 커밋에 대한 정보 보기

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 View Commit Details 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 BatchGetCommits 섹션을 참조하세요.

batch-get-repositories

다음 코드 예시에서는 `batch-get-repositories`의 사용 방법을 보여줍니다.

AWS CLI

여러 리포지토리에 대한 세부 정보 보기

이 예시에서는 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": []
}
```

- API 세부 정보는 AWS CLI 명령 참조의 BatchGetRepositories 섹션을 참조하세요.

create-approval-rule-template

다음 코드 예시에서는 `create-approval-rule-template`의 사용 방법을 보여줍니다.

AWS CLI

승인 규칙 템플릿 생성

다음 create-approval-rule-template 예시에서는 풀 요청을 main브랜치에 병합하기 전에 승인하도록 명명된 승인 규칙 2-approver-rule-for-main ``. The template requires two users who assume the role of ``CodeCommitReview 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 Create an Approval Rule Template 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateApprovalRuleTemplate 섹션을 참조하세요.

create-branch

다음 코드 예시에서는 create-branch의 사용 방법을 보여줍니다.

AWS CLI

브랜치 생성

이 예시에서는 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.
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateBranch 섹션을 참조하세요.

`create-commit`

다음 코드 예시에서는 `create-commit`의 사용 방법을 보여줍니다.

AWS CLI

커밋 생성

다음 `create-commit` 예시에서는 `main` 브랜치에 이름이 지정된 리포지토리에 `readme.md` 파일을 추가하는 `MyDemoRepo` 리포지토리에 대한 초기 커밋을 만드는 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 Create a Commit in AWS CodeCommit 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateCommit 섹션을 참조하세요.

`create-pull-request-approval-rule`

다음 코드 예시에서는 `create-pull-request-approval-rule`의 사용 방법을 보여줍니다.

AWS CLI

풀 요청에 대한 승인 규칙 생성

다음 `create-pull-request-approval-rule` 예시에서는 지정된 풀 리퀘스트에 대해 이름이 지정된 `Require two approved approvers` 승인 규칙을 만듭니다. 이 규칙은 하나의 승인 풀에서 두 개의 승인이 필요하도록 지정합니다. 이 풀에는 123456789012 AWS 계정에서 `CodeCommitReview` 역할을 수임하고 CodeCommit에 액세스할 수 있는 모든 사용자가 포함됩니다. 또한 동일한 AWS 계정에서 `Nikhil_Jayashankar` 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 Create an Approval Rule 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreatePullRequestApprovalRule 섹션을 참조하세요.

`create-pull-request`

다음 코드 예시에서는 `create-pull-request`의 사용 방법을 보여줍니다.

AWS CLI

풀 요청 생성

다음 `create-pull-request` 예시에서는 'jane-branch' 소스 브랜치를 대상으로 'Please review these changes by Tuesday'라는 설명과 함께 'Pronunciation difficulty analyzer'라는 풀 리퀘스트를 생성하고, 'MyDemoRepo'라는 CodeCommit 리포지토리의 기본 브랜치 'main'에 병합할 것입니다.

```
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
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [CreatePullRequest](#) 섹션을 참조하세요.

create-repository

다음 코드 예시에서는 `create-repository`의 사용 방법을 보여줍니다.

AWS CLI

리포지토리 생성

이 예시에서는 리포지토리를 생성하고 사용자의 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateRepository를 참조하세요.

`create-unreferenced-merge-commit`

다음 코드 예시에서는 `create-unreferenced-merge-commit`의 사용 방법을 보여줍니다.

AWS CLI

두 커밋 지정자를 병합한 결과를 나타내는 참조되지 않은 커밋

다음 `create-unreferenced-merge-commit` 예시에서는 `MyDemoRepo` 리포지토리에서 `bugfix-1234` 소스 브랜치와 `main` 대상 브랜치 간의 병합 결과를 나타내는 커밋을 생성한다.

```
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 Resolve Conflicts in a Pull Request 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateUnreferencedMergeCommit 섹션을 참조하세요.

`credential-helper`

다음 코드 예시에서는 `credential-helper`의 사용 방법을 보여줍니다.

AWS CLI

AWS CodeCommit을 사용하여 AWS CLI에 포함된 자격 증명 도우미를 설정하는 방법

`credential-helper` 유틸리티는 AWS CLI에서 직접 호출하도록 설계되지 않았습니다. 대신 로컬 컴퓨터를 설정하기 위한 `git config` 명령과 함께 파라미터로 사용하기 위한 것입니다. 이를 통해 Git은 CodeCommit 리포지토리와 상호 작용하기 위해 AWS로 인증해야 할 때마다 HTTPS와 암호로 서명된 버전의 IAM 사용자 자격 증명 또는 Amazon EC2 인스턴스 역할을 사용할 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 AWS 사용 설명서의 Setting up for CodeCommit Using Other Methods 섹션을 참조하세요. 콘텐츠를 주의 깊게 검토한 다음AWS CodeCommit 사용 설명서의 Linux, macOS 또는 Unix에서 HTTPS 연결의 경우 또는 Windows에서 HTTPS 연결의 경우 중 하나의 절차를 따르세요.

- API 세부 정보는 AWS CLI 명령 참조의 CredentialHelper 섹션을 참조하세요.

`delete-approval-rule-template`

다음 코드 예시에서는 `delete-approval-rule-template`의 사용 방법을 보여줍니다.

AWS CLI

승인 규칙 템플릿 삭제

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 Delete an Approval Rule Template 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteApprovalRuleTemplate 섹션을 참조하세요.

`delete-branch`

다음 코드 예시에서는 `delete-branch`의 사용 방법을 보여줍니다.

AWS CLI

브랜치 삭제

이 예시는 AWS CodeCommit 리포지토리에서 브랜치를 삭제하는 방법을 보여줍니다.

명령:

```
aws codecommit delete-branch --repository-name MyDemoRepo --branch-name MyNewBranch
```

출력:

```
{
  "branch": {
      "commitId": "317f8570EXAMPLE",
      "branchName": "MyNewBranch"
  }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteBranch 섹션을 참조하세요.

delete-comment-content

다음 코드 예시에서는 delete-comment-content의 사용 방법을 보여줍니다.

AWS CLI

커밋의 설명 내용 삭제

설명을 작성한 경우에만 설명의 콘텐츠를 삭제할 수 있습니다. 이 예시에서는 시스템 생성 ID가 ff30b348EXAMPLEb9aa670f인 설명의 내용을 삭제하는 방법을 보여줍니다.

```
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
            }
        }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteCommentContent 섹션을 참조하세요.

delete-file

다음 코드 예시에서는 delete-file의 사용 방법을 보여줍니다.

AWS CLI

파일 삭제

다음 delete-file 예시에서는 MyDemoRepo 리포지토리에서 가장 최근 커밋 ID가 c5709475EXAMPLE인 main 브랜치에서README.md 파일을 삭제하는 방법을 보여줍니다.

```
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 안내서의 Edit or Delete a File in AWS CodeCommit 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteFile 섹션을 참조하세요.

delete-pull-request-approval-rule

다음 코드 예시에서는 delete-pull-request-approval-rule의 사용 방법을 보여줍니다.

AWS CLI

풀 요청에 대한 승인 규칙 삭제

다음 `delete-pull-request-approval-rule` 예제에서는 지정된 풀 요청에 대해 `My Approval Rule` 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 [Edit or Delete an Approval Rule](#) 섹션을 참조하세요.

• API 세부 정보는 AWS CLI 명령 참조의 [DeletePullRequestApprovalRule](#) 섹션을 참조하세요.

`delete-repository`

다음 코드 예시에서는 `delete-repository`의 사용 방법을 보여줍니다.

AWS CLI

리포지토리 삭제

이 예시에서는 AWS CodeCommit 리포지토리를 삭제하는 방법을 보여줍니다.

명령:

```
aws codecommit delete-repository --repository-name MyDemoRepo
```

출력:

```
{
  "repositoryId": "f7579e13-b83e-4027-aaef-650c0EXAMPLE"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteRepository]를 참조하세요.

`describe-merge-conflicts`

다음 코드 예시에서는 `describe-merge-conflicts`의 사용 방법을 보여줍니다.

AWS CLI

병합 충돌에 대한 자세한 정보를 얻으려면

다음 `describe-merge-conflicts` 예시에서는 지정된 소스 브랜치와 대상 브랜치에 있는 `readme.md` 파일에 대한 병합 충돌을 THREE_WAY_MERGE 전략을 사용하여 확인합니다.

```
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 Resolve Conflicts in a Pull Request 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeMergeConflicts 섹션을 참조하세요.

`describe-pull-request-events`

다음 코드 예시에서는 `describe-pull-request-events`의 사용 방법을 보여줍니다.

AWS CLI

풀 요청에서 이벤트를 보는 방법

다음 describe-pull-request-events 예시에서는 ID가 '8'인 풀 리퀘스트에 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribePullRequestEvents 섹션을 참조하세요.

disassociate-approval-rule-template-from-repository

다음 코드 예시에서는 disassociate-approval-rule-template-from-repository의 사용 방법을 보여줍니다.

AWS CLI

리포지토리에서 승인 규칙 템플릿 연결 해제

다음 disassociate-approval-rule-template-from-repository 예시는 MyDemoRepo 리포지토리에서 지정된 승인 규칙 템플릿을 연결 해제합니다.

```
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 Disassociate an Approval Rule Template 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisassociateApprovalRuleTemplateFromRepository 섹션을 참조하세요.

evaluate-pull-request-approval-rules

다음 코드 예시에서는 evaluate-pull-request-approval-rules의 사용 방법을 보여줍니다.

AWS CLI

풀 요청에 모든 승인 규칙이 충족되었는지 평가하는 방법

다음 evaluate-pull-request-approval-rules 예시는 지정된 풀 리퀘스트에 대한 승인 규칙의 상태를 평가합니다. 이 예시에서는 풀 리퀘스트에 대한 승인 규칙이 충족되지 않았으므로 명령의 출력에 false의 값 approved가 표시됩니다.

```
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 Merge a Pull Request 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 EvaluatePullRequestApprovalRules 섹션을 참조하세요.

get-approval-rule-template

다음 코드 예시에서는 get-approval-rule-template의 사용 방법을 보여줍니다.

AWS CLI

승인 규칙 템플릿의 내용 가져오기

다음 get-approval-rule-template 예시는 승인 규칙 템플릿 1-approver-rule-for-all-pull-requests의 콘텐츠를 가져옵니다.

```
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 Manage Approval Rule Templates 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetApprovalRuleTemplate 섹션을 참조하세요.

get-blob

다음 코드 예시에서는 get-blob의 사용 방법을 보여줍니다.

AWS CLI

Git BLOB 객체에 대한 정보 보기

다음 `get-blob` 예시에서는 AWS CodeCommit 리포지토리 'MyDemoRepo'에 있는 ID가 '2eb4af3bEXAMPLE'인 Git 블롭에 대한 정보를 검색합니다.

```
aws codecommit get-blob  --repository-name MyDemoRepo  --blob-id 2eb4af3bEXAMPLE
```

출력:

```
{
    "content": "QSBCaW5hcnkgTGFyToEXAMPLE="
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetBlob 섹션을 참조하세요.

`get-branch`

다음 코드 예시에서는 `get-branch`의 사용 방법을 보여줍니다.

AWS CLI

커밋에 대한 정보 가져오기

이 예시에서는 AWS CodeCommit 리포지토리의 브랜치에 대한 정보를 가져옵니다.

명령:

```
aws codecommit get-branch --repository-name MyDemoRepo --branch-name MyNewBranch
```

출력:

```
{
  "BranchInfo": {
        "commitID": "317f8570EXAMPLE",
                "branchName": "MyNewBranch"
  }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetBranch 섹션을 참조하세요.

get-comment-reactions

다음 코드 예시에서는 get-comment-reactions의 사용 방법을 보여줍니다.

AWS CLI

설명에 대한 이모티콘 반응 보기

다음 get-comment-reactions 예시에서는 ID가 abcd1234EXAMPLEb5678efgh인 설명에 대한 모든 이모티콘 반응을 나열합니다. 셸의 글꼴이 이모지 버전 1.0 표시를 지원하는 경우 emoji의 출력에 이모지가 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 Comment on a commit in AWS CodeCommit 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetCommentReactions 섹션을 참조하세요.

get-comment

다음 코드 예시에서는 get-comment의 사용 방법을 보여줍니다.

AWS CLI

설명에 대한 세부 정보 보기

이 예시에서는 시스템에서 생성된 설명 ID가 ff30b348EXAMPLEb9aa670f인 설명의 세부 정보를 보는 방법을 설명합니다.

```
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
            }
        }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetComment 섹션을 참조하세요.

get-comments-for-compared-commit

다음 코드 예시에서는 get-comments-for-compared-commit의 사용 방법을 보여줍니다.

AWS CLI

커밋에 대한 설명 보기

예를 들어 MyDemoRepo 리포지토리의 두 커밋 간의 비교에 대해 작성된 설명을 보려면 다음을 수행합니다.

```
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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetCommentsForComparedCommit 섹션을 참조하세요.

get-comments-for-pull-request

다음 코드 예시에서는 `get-comments-for-pull-request`의 사용 방법을 보여줍니다.

AWS CLI

풀 리퀘스트에 남긴 설명 보기

이 예시에서는 `MyDemoRepo` 리포지토리에서 풀 리퀘스트에 대한 설명을 보는 방법을 보여줍니다.

```
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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetCommentsForPullRequest 섹션을 참조하세요.

get-commit

다음 코드 예시에서는 get-commit의 사용 방법을 보여줍니다.

AWS CLI

리포지토리의 커밋에 대한 정보 보기

이 예시에서는 'MyDemoRepo'라는 이름의 CodeCommit 리포지토리에 있는 시스템 생성 ID가 '7e9fd3091thisisanexamplethisisanexample1'인 커밋에 대한 세부 정보를 보여줍니다.

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"
  }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetCommit 섹션을 참조하세요.

`get-differences`

다음 코드 예시에서는 `get-differences`의 사용 방법을 보여줍니다.

AWS CLI

리포지토리의 커밋 지정자의 차이점에 대한 정보를 가져오는 방법

이 예시에서는 AWS CodeCommit 리포지토리의 이름이 변경된 폴더에서 MyDemoRepo라는 이름의 두 커밋 지정자(브랜치, 태그, HEAD 또는 기타 정규화된 참조(커밋 ID 등)) 간의 변경 사항에 대한 보기 메타데이터 정보를 보여줍니다. 이 예시에서는 이러한 옵션을 사용하여 결과를 제한하는 방법을 보다 완벽하게 설명하기 위해 --before-commit-specifier, --before-path, --after-path 등 필수 옵션이 아닌 몇 가지 옵션을 포함합니다. 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"
          }
      }
  ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [GetDifferences](https://docs.aws.amazon.com/goto/aws-cli/codecommit-2015-04-13/GetDifferences) 섹션을 참조하세요.

get-file

다음 코드 예시에서는 `get-file`의 사용 방법을 보여줍니다.

AWS CLI

AWS CodeCommit 리포지토리에서 파일의 base-64 인코딩 콘텐츠를 가져오는 방법

다음 `get-file` 예시에서는 `MyDemoRepo` 리포지토리에 `main` 브랜치에서 `README.md` 파일의 base-64로 인코딩된 내용을 가져오는 방법을 보여줍니다.

```
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 안내서의 GetFile을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetFile 섹션을 참조하세요.

get-folder

다음 코드 예시에서는 `get-folder`의 사용 방법을 보여줍니다.

AWS CLI

AWS CodeCommit 리포지토리의 폴더 콘텐츠를 가져오는 방법

다음 `get-folder` 예시에서는 `MyDemoRepo` 리포지토리에서 최상위 폴더의 콘텐츠를 가져오는 방법을 보여줍니다.

```
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 안내서의 GetFolder 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetFolder 섹션을 참조하세요.

get-merge-commit

다음 코드 예시에서는 get-merge-commit의 사용 방법을 보여줍니다.

AWS CLI

병합 커밋에 대한 자세한 정보 얻기

다음 get-merge-commit 예제에서는 MyDemoRepo 리포지토리에서 main 대상 브랜치와 bugfix-bug1234 소스 브랜치에 대한 병합 커밋에 관한 세부 정보를 표시합니다.

```
aws codecommit get-merge-commit \
    --source-commit-specifier bugfix-bug1234 \
    --destination-commit-specifier main \
    --repository-name MyDemoRepo
```

출력:

```
{
    "sourceCommitId": "c5709475EXAMPLE",
    "destinationCommitId": "317f8570EXAMPLE",
    "baseCommitId": "fb12a539EXAMPLE",
    "mergeCommitId": "ffc4d608eEXAMPLE"
}
```

자세한 내용은 AWS CodeCommit 사용 설명서의 View Commit Details 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetMergeCommit 섹션을 참조하세요.

get-merge-conflicts

다음 코드 예시에서는 get-merge-conflicts의 사용 방법을 보여줍니다.

AWS CLI

풀 리퀘스트에 병합 충돌이 있는지 확인

다음 get-merge-conflicts 예시에서는 MyDemoRepo 리포지토리에서 feature-randomizationfeature 소스 브랜치 끝과 'main' 대상 브랜치 사이에 병합 충돌이 있는지 표시합니다.

```
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"
            }
```

```
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetMergeConflicts 섹션을 참조하세요.

get-merge-options

다음 코드 예시에서는 get-merge-options의 사용 방법을 보여줍니다.

AWS CLI

두 개의 지정된 브랜치를 병합하는 데 사용할 수 있는 병합 옵션에 대한 정보를 가져오는 방법

다음 get-merge-options 예시에서는 bugfix-bug1234 소스 브랜치를 MyDemoRepo 리포지토리에 있는 main 대상 브랜치와 병합하는 데 사용할 수 있는 병합 옵션을 결정합니다.

```
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 Resolve Conflicts in a Pull Request 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetMergeOptions 섹션을 참조하세요.

get-pull-request-approval-states

다음 코드 예시에서는 get-pull-request-approval-states의 사용 방법을 보여줍니다.

AWS CLI

풀 요청에 대한 승인 보기

다음 get-pull-request-approval-states 예시에서는 지정된 풀 리퀘스트에 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 View Pull Requests 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetPullRequestApprovalStates 섹션을 참조하세요.

get-pull-request-override-state

다음 코드 예시에서는 get-pull-request-override-state의 사용 방법을 보여줍니다.

AWS CLI

풀 리퀘스트의 재정의 상태에 대한 정보 가져오기

다음 get-pull-request-override-state 예시에서는 지정된 풀 리퀘스트에 대한 재정의 상태를 반환합니다. 이 예시에서는 Mary Major라는 사용자가 풀 리퀘스트에 대한 승인 규칙을 재정의했기 때문에 출력은 true라는 값을 반환합니다.

```
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 Override Approval Rules on a Pull Request 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetPullRequestOverrideState 섹션을 참조하세요.

get-pull-request

다음 코드 예시에서는 `get-pull-request`의 사용 방법을 보여줍니다.

AWS CLI

풀 요청의 세부 정보 보기

이 예시에서는 ID가 27인 풀 요청에 대한 정보를 보는 방법을 보여줍니다.

```
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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetPullRequest 섹션을 참조하세요.

get-repository-triggers

다음 코드 예시에서는 get-repository-triggers의 사용 방법을 보여줍니다.

AWS CLI

리포지토리의 트리거에 대한 정보를 가져오는 방법

이 예시에서는 MyDemoRepo라는 AWS CodeCommit 리포지토리에 대해 구성된 트리거에 대한 세부 정보를 보여줍니다.

```
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"
            ]
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetRepositoryTriggers 섹션을 참조하세요.

get-repository

다음 코드 예시에서는 `get-repository`의 사용 방법을 보여줍니다.

AWS CLI

리포지토리 관련 정보를 가져오기

이 예시에서는 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [GetRepository](https://docs.aws.amazon.com/goto/aws-cli/codecommit-2015-04-13/GetRepository) 섹션을 참조하세요.

`list-approval-rule-templates`

다음 코드 예시에서는 `list-approval-rule-templates`의 사용 방법을 보여줍니다.

AWS CLI

AWS 리전의 모든 승인 규칙 템플릿 나열

다음 `list-approval-rule-templates` 예시에서는 지정된 리전의 모든 승인 규칙 템플릿을 나열합니다. AWS 리전이 파라미터로 지정되지 않은 경우 명령은 명령을 실행하는 데 사용되는 AWS CLI 프로파일에 지정된 리전에 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 [Manage Approval Rule Templates](#) 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ListApprovalRuleTemplates](#) 섹션을 참조하세요.

`list-associated-approval-rule-templates-for-repository`

다음 코드 예시에서는 `list-associated-approval-rule-templates-for-repository`의 사용 방법을 보여줍니다.

AWS CLI

리포지토리와 연결된 모든 템플릿 나열

다음 `list-associated-approval-rule-templates-for-repository` 예시에서는 `MyDemoRepo` 리포지토리와 연결된 모든 승인 규칙 템플릿을 나열합니다.

```
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 [Manage Approval Rule Templates](#) 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ListAssociatedApprovalRuleTemplatesForRepository](#) 섹션을 참조하세요.

`list-branches`

다음 코드 예시에서는 `list-branches`의 사용 방법을 보여줍니다.

AWS CLI

브랜치 이름 목록 보기

이 예시에서는 AWS CodeCommit 리포지토리의 모든 브랜치 이름을 나열합니다.

```
aws codecommit list-branches \
    --repository-name MyDemoRepo
```

출력:

```
{
    "branches": [
        "MyNewBranch",
        "main"
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListBranches 섹션을 참조하세요.

`list-pull-requests`

다음 코드 예시에서는 `list-pull-requests`의 사용 방법을 보여줍니다.

AWS CLI

리포지토리의 풀 요청 보기

이 예시에서는 ARN이 'arn:aws:iam::111111111111:user/Li_Juan'이고 상태가 'CLOSED'인 IAM 사용자가 만든 풀 리퀘스트를 'MyDemoRepo'라는 AWS CodeCommit 리포지토리에 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"]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [ListPullRequests](#) 섹션을 참조하세요.

list-repositories-for-approval-rule-template

다음 코드 예시에서는 `list-repositories-for-approval-rule-template`의 사용 방법을 보여줍니다.

AWS CLI

템플릿과 연결된 모든 리포지토리 나열

다음 `list-repositories-for-approval-rule-template` 예시에서는 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 [Manage Approval Rule Templates](#) 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ListRepositoriesForApprovalRuleTemplate](#) 섹션을 참조하세요.

list-repositories

다음 코드 예시에서는 `list-repositories`의 사용 방법을 보여줍니다.

AWS CLI

리포지토리의 목록 보기

이 예시에서는 사용자의 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"
      }
  ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListRepositories 섹션을 참조하세요.

list-tags-for-resource

다음 코드 예시에서는 `list-tags-for-resource`의 사용 방법을 보여줍니다.

AWS CLI

리포지토리의 AWS 태그 보기

다음 `list-tags-for-resource` 예시에서는 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 View Tags for a Repository 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource를 참조하세요.

merge-branches-by-fast-forward

다음 코드 예시에서는 merge-branches-by-fast-forward의 사용 방법을 보여줍니다.

AWS CLI

빨리 감기 병합 전략을 사용하여 두 브랜치 병합

다음 merge-branches-by-fast-forward 예시에서는 지정된 소스 브랜치를 MyDemoRepo 리포지토리의 지정된 대상 브랜치와 병합합니다.

```
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 Compare and Merge Branches 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 MergeBranchesByFastForward 섹션을 참조하세요.

merge-branches-by-squash

다음 코드 예시에서는 merge-branches-by-squash의 사용 방법을 보여줍니다.

AWS CLI

스쿼시 병합 전략을 사용하여 두 브랜치 병합

다음 `merge-branches-by-squash` 예시에서는 지정된 소스 브랜치를 `MyDemoRepo` 리포지토리의 지정된 대상 브랜치와 병합합니다.

```
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 Compare and Merge Branches 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 MergeBranchesBySquash 섹션을 참조하세요.

`merge-branches-by-three-way`

다음 코드 예시에서는 `merge-branches-by-three-way`의 사용 방법을 보여줍니다.

AWS CLI

3방향 병합 전략을 사용하여 두 브랜치 병합

다음 `merge-branches-by-three-way` 예시에서는 지정된 소스 브랜치를 `MyDemoRepo` 리포지토리의 지정된 대상 브랜치와 병합합니다.

```
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 Compare and Merge Branches 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 MergeBranchesByThreeWay 섹션을 참조하세요.

merge-pull-request-by-fast-forward

다음 코드 예시에서는 `merge-pull-request-by-fast-forward`의 사용 방법을 보여줍니다.

AWS CLI

풀 요청을 병합하고 닫으려면

이 예시에서는 MyDemoRepo 리포지토리에서 ID가 '47'이고 소스 커밋 ID가 '99132ab0EXAMPLE'인 풀 리퀘스트를 병합하고 닫는 방법을 보여 줍니다.

```
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 Merge a Pull Request 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 MergePullRequestByFastForward 섹션을 참조하세요.

merge-pull-request-by-squash

다음 코드 예시에서는 `merge-pull-request-by-squash`의 사용 방법을 보여줍니다.

AWS CLI

스쿼시 병합 전략을 사용하여 풀 요청을 병합하는 방법

다음 `merge-pull-request-by-squash` 예시에서는 `MyDemoRepo` 리포지토리에서 ACCEPT_SOURCE의 충돌 해결 전략을 사용하여 지정된 풀 요청을 병합하고 닫습니다.

```
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 Merge a Pull Request 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 MergePullRequestBySquash 섹션을 참조하세요.

merge-pull-request-by-three-way

다음 코드 예시에서는 `merge-pull-request-by-three-way`의 사용 방법을 보여줍니다.

AWS CLI

3방향 병합 전략을 사용하여 풀 요청을 병합하는 방법

다음 `merge-pull-request-by-three-way` 예시에서는 `MyDemoRepo` 리포지토리에서 충돌 세부 정보 및 충돌 해결 전략에 대한 기본 옵션을 사용하여 지정된 풀 요청을 병합하고 닫습니다.

```
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 Merge a Pull Request 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 MergePullRequestByThreeWay 섹션을 참조하세요.

override-pull-request-approval-rules

다음 코드 예시에서는 `override-pull-request-approval-rules`의 사용 방법을 보여줍니다.

AWS CLI

풀 요청에 대한 승인 규칙 요구 사항 재정의

다음 `override-pull-request-approval-rules` 예시에서는 지정된 풀 요청에 대한 승인 규칙을 재정의합니다. 대신 재정의를 취소하는 방법 `--override-status` 파라미터 값을 로 설정합니다`REVOKE`.

```
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 Override Approval Rules on a Pull Request 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 OverridePullRequestApprovalRules 섹션을 참조하세요.

post-comment-for-compared-commit

다음 코드 예시에서는 `post-comment-for-compared-commit`의 사용 방법을 보여줍니다.

AWS CLI

커밋에 설명 작성

이 예시에서는 `MyDemoRepo` 리포지토리에 있는 두 커밋을 비교하여 `cl_sample.js` 파일에 변경 사항에 대한 설명 `"Can you add a test case for this?"`을 추가하는 방법을 보여줍니다.

```
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"
    }
}
```


- API 세부 정보는 AWS CLI 명령 참조의 PostCommentForComparedCommit 섹션을 참조하세요.

post-comment-for-pull-request

다음 코드 예시에서는 post-comment-for-pull-request의 사용 방법을 보여줍니다.

AWS CLI

풀 리퀘스트에 설명 추가

다음 `post-comment-for-pull-request` 예시에서는 'These don't appear to be used anywhere Can we remove them?'라는 설명을 `MyDemoRepo` 리포지토리에 있는 풀 리퀘스트의 ID가 47인 `ahs_count.py` 파일에 대한 변경 사항에 추가합니다.

```
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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [PostCommentForPullRequest](https://docs.aws.amazon.com/goto/aws-cli/codecommit-2015-04-13/PostCommentForPullRequest) 섹션을 참조하세요.

`post-comment-reply`

다음 코드 예시에서는 `post-comment-reply`의 사용 방법을 보여줍니다.

AWS CLI

커밋 또는 풀 리퀘스트의 설명에 답장

이 예시에서는 시스템 생성 ID가`abcd1234EXAMPLEb5678efgh`인 설명에 대한 응답 `"Good catch. I'll remove them."`을 추가하는 방법을 보여줍니다.

```
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": []
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [PostCommentReply](https://docs.aws.amazon.com/goto/aws-cli/codecommit-2015-04-13/PostCommentReply) 섹션을 참조하세요.

`put-comment-reaction`

다음 코드 예시에서는 `put-comment-reaction`의 사용 방법을 보여줍니다.

AWS CLI

이모티콘으로 커밋에 대한 설명에 답글 달기

다음 `put-comment-reaction` 예시에서는 ID가 `:thumbsup:`이고 이모티콘 반응 값이 `abcd1234EXAMPLEb5678efgh`인 설명에 응답합니다.

```
aws codecommit put-comment-reaction \
    --comment-id abcd1234EXAMPLEb5678efgh \
    --reaction-value :thumbsup:
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS CodeCommit 사용 설명서의 Comment on a commit in AWS CodeCommit 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutCommentReaction 섹션을 참조하세요.

put-file

다음 코드 예시에서는 `put-file`의 사용 방법을 보여줍니다.

AWS CLI

리포지토리에 파일 추가

다음 `put-file` 예시에서는 가장 최근 커밋의 ID가 '4c925148EXAMPLE'인 'feature-randomizationfeature'라는 브랜치에 'ExampleSolution.py'라는 파일을 'MyDemoRepo'라는 리포지토리에 추가합니다.

```
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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 PutFile 섹션을 참조하세요.

put-repository-triggers

다음 코드 예시에서는 `put-repository-triggers`의 사용 방법을 보여줍니다.

AWS CLI

리포지토리에서 트리거를 추가하거나 업데이트하는 방법

이 예시에서는 MyDemoRepo라는 리포지토리에 대한 모든 트리거의 구조가 포함된 이미 생성된 JSON 파일(여기서는 MyTriggers.json)을 사용하여 'MyFirstTrigger' 및 'MySecondTrigger'라는 트리거를 업데이트하는 방법을 보여줍니다. 기존 트리거에 대한 JSON을 가져오는 방법을 알아보려면 get-repository-triggers 명령을 참조하세요.

```
aws codecommit put-repository-triggers \
    --repository-name MyDemoRepo file://MyTriggers.json
```

`MyTriggers.json`의 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 PutRepositoryTriggers 섹션을 참조하세요.

tag-resource

다음 코드 예시에서는 tag-resource의 사용 방법을 보여줍니다.

AWS CLI

기존 리포지토리에 AWS 태그 추가

다음 tag-resource 예시에서는 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 Add a Tag to a Repository 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

`test-repository-triggers`

다음 코드 예시에서는 `test-repository-triggers`의 사용 방법을 보여줍니다.

AWS CLI

리포지토리에서 트리거를 테스트하는 방법

이 예시에서는 MyDemoRepo라는 AWS CodeCommit 리포지토리에서 'MyFirstTrigger'라는 트리거를 테스트하는 방법을 보여줍니다. 이 예시에서는 리포지토리의 이벤트가 Amazon Simple Notification Service(Amazon SNS) 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": []
}
```

- API 세부 정보는 AWS CLI 명령 참조의 TestRepositoryTriggers 섹션을 참조하세요.

`untag-resource`

다음 코드 예시에서는 `untag-resource`의 사용 방법을 보여줍니다.

AWS CLI

리포지토리에서 AWS 태그 제거

다음 `untag-resource` 예시에서는 `MyDemoRepo` 리포지토리에서 지정된 키가 있는 태그를 제거합니다.

```
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 Remove a Tag from a Repository 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

update-approval-rule-template-content

다음 코드 예시에서는 update-approval-rule-template-content의 사용 방법을 보여줍니다.

AWS CLI

승인 규칙 템플릿의 내용 업데이트

다음 update-approval-rule-template-content 예시에서는 지정된 승인 규칙 템플릿의 내용을 변경하여 승인 풀을 CodeCommitReview 역할을 맡는 사용자로 재정의합니다.

```
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 [Manage Approval Rule Templates](#) 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [UpdateApprovalRuleTemplateContent](#) 섹션을 참조하세요.

`update-approval-rule-template-description`

다음 코드 예시에서는 `update-approval-rule-template-description`의 사용 방법을 보여줍니다.

AWS CLI

승인 규칙 템플릿의 설명 업데이트

다음 `update-approval-rule-template-description` 예시에서는 지정된 승인 규칙 템플릿의 설명을`Requires 1 approval for all pull requests from the CodeCommitReview pool`로 변경합니다.

```
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 Manage Approval Rule Templates 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateApprovalRuleTemplateDescription 섹션을 참조하세요.

update-approval-rule-template-name

다음 코드 예시에서는 update-approval-rule-template-name의 사용 방법을 보여줍니다.

AWS CLI

승인 규칙 템플릿의 이름 업데이트

다음 update-approval-rule-template-name 예시에서는 승인 규칙 템플릿의 이름을 1-approver-rule에서 1-approvers-rule-for-all-pull-requests`로 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 [Manage Approval Rule Templates](#) 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [UpdateApprovalRuleTemplateName](#) 섹션을 참조하세요.

update-comment

다음 코드 예시에서는 `update-comment`의 사용 방법을 보여줍니다.

AWS CLI

커밋에 대한 설명 업데이트

이 예시에서는 ID가 `442b498bEXAMPLE5756813`인 설명에 `"Fixed as requested. I'll update the pull request."` 콘텐츠를 추가하는 방법을 보여줍니다.

```
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
            }
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [UpdateComment](#) 섹션을 참조하세요.

update-default-branch

다음 코드 예시에서는 `update-default-branch`의 사용 방법을 보여줍니다.

AWS CLI

리포지토리의 기본 브랜치 변경

이 예시에서는 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.
```

- API 세부 정보는 AWS CLI 명령 참조의 [UpdateDefaultBranch](#) 섹션을 참조하세요.

update-pull-request-approval-rule-content

다음 코드 예시에서는 `update-pull-request-approval-rule-content`의 사용 방법을 보여줍니다.

AWS CLI

풀 요청에 대한 승인 규칙 편집

다음 `update-pull-request-approval-rule-content` 예시에서는 123456789012 AWS 계정의 모든 IAM 사용자를 포함하는 승인 풀에서 한 명의 사용자 승인을 요구하도록 승인 규칙을 지정하여 업데이트합니다.

```
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 Edit or Delete an Approval Rule 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdatePullRequestApprovalRuleContent 섹션을 참조하세요.

`update-pull-request-approval-state`

다음 코드 예시에서는 `update-pull-request-approval-state`의 사용 방법을 보여줍니다.

AWS CLI

풀 요청에 대한 승인 또는 승인 취소

다음 `update-pull-request-approval-state` ID가 27이고 개정 ID가 `9f29d167EXAMPLE`인 풀 리퀘스트를 승인합니다. 대신 승인을 취소하려면 `--approval-state` 파라미터 값을 `REVOKE`로 설정합니다.

```
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 Review a Pull Request 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdatePullRequestApprovalState 섹션을 참조하세요.

update-pull-request-description

다음 코드 예시에서는 update-pull-request-description의 사용 방법을 보여줍니다.

AWS CLI

폴 요청에 대한 설명을 변경하는 방법

이 예시에서는 ID가 47인 풀 리퀘스트에 대한 설명을 변경하는 방법을 보여줍니다.

```
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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdatePullRequestDescription 섹션을 참조하세요.

update-pull-request-status

다음 코드 예시에서는 update-pull-request-status의 사용 방법을 보여줍니다.

AWS CLI

풀 요청의 상태 변경

이 예시에서는 MyDemoRepo라는 이름의 AWS CodeCommit 리포지토리에서 ID가 42인 풀 리퀘스트의 상태를 CLOSED로 변경하는 방법을 보여줍니다.

```
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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdatePullRequestStatus 섹션을 참조하세요.

update-pull-request-title

다음 코드 예시에서는 `update-pull-request-title`의 사용 방법을 보여줍니다.

AWS CLI

풀 요청의 제목을 변경하는 방법

이 예시에서는 ID가 47인 풀 리퀘스트의 제목을 변경하는 방법을 보여줍니다.

```
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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [UpdatePullRequestTitle](#) 섹션을 참조하세요.

update-repository-description

다음 코드 예시에서는 update-repository-description의 사용 방법을 보여줍니다.

AWS CLI

리포지토리의 설명 변경

이 예시에서는 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.
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateDocumentDefaultVersion 섹션을 참조하세요.

update-repository-name

다음 코드 예시에서는 update-repository-name의 사용 방법을 보여줍니다.

AWS CLI

리포지토리의 이름 변경

이 예시에서는 AWS CodeCommit 리포지토리의 이름을 변경합니다. 이 명령은 오류가 있는 경우에만 출력을 생성합니다. AWS CodeCommit 리포지토리의 이름을 변경하면 사용자가 리포지토리에 연결하는 데 필요한 SSH 및 HTTPS URL이 변경됩니다. 사용자는 연결 설정을 업데이트할 때까지 이 리포지토리에 연결할 수 없습니다. 또한 리포지토리의 ARN이 변경되므로, 리포지토리 이름을 변경하면 이 리포지토리의 ARN을 사용하는 모든 IAM 사용자 정책이 무효화됩니다.

명령:

```
aws codecommit update-repository-name --old-name MyDemoRepo --new-
name MyRenamedDemoRepo
```

출력:

```
None.
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateRepositoryName 섹션을 참조하세요.

AWS CLI를 사용한 CodeDeploy 예시

다음 코드 예시에서는 CodeDeploy에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

add-tags-to-on-premises-instances

다음 코드 예시에서는 add-tags-to-on-premises-instances 코드를 사용하는 방법을 보여줍니다.

AWS CLI

온프레미스 인스턴스에 태그 추가

다음 add-tags-to-on-premises-instances 예시에서는 AWS CodeDeploy 동일한 온프레미스 인스턴스 태그를 두 개의 온프레미스 인스턴스에 연결합니다. 온프레미스 인스턴스를 AWS CodeDeploy에 등록하지 않습니다.

```
aws deploy add-tags-to-on-premises-instances \
    --instance-names AssetTag12010298EX AssetTag23121309EX \
    --tags Key=Name,Value=CodeDeployDemo-OnPrem
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 AddTagsToOnPremisesInstances 섹션을 참조하세요.

`batch-get-application-revisions`

다음 코드 예시에서는 `batch-get-application-revisions` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

애플리케이션 개정에 대한 정보를 검색하는 방법

다음 `batch-get-application-revisions` 예시에서는 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 AWS CodeDeploy API 참조의 BatchGetApplicationRevisions를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 BatchGetApplicationRevisions 섹션을 참조하세요.

batch-get-applications

다음 코드 예시에서는 `batch-get-applications` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

여러 애플리케이션에 대한 정보를 가져오는 방법

다음 `batch-get-applications` 예시에서는 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
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 BatchGetApplications 섹션을 참조하세요.

batch-get-deployment-groups

다음 코드 예시에서는 `batch-get-deployment-groups` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

하나 이상의 배포 그룹에 대한 정보 검색

다음 `batch-get-deployment-groups` 예시에서는 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 AWS CodeDeploy API 참조의 BatchGetDeploymentGroups를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 BatchGetDeploymentGroups 섹션을 참조하세요.

batch-get-deployment-targets

다음 코드 예시에서는 `batch-get-deployment-targets` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

배포와 연결된 대상을 검색하는 방법

다음 `batch-get-deployment-targets` 예시에서는 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 AWS CodeDeploy API 참조의 BatchGetDeploymentTargets를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 BatchGetDeploymentTargets 섹션을 참조하세요.

batch-get-deployments

다음 코드 예시에서는 batch-get-deployments 코드를 사용하는 방법을 보여줍니다.

AWS CLI

여러 배포에 대한 정보를 가져오는 방법

다음 `batch-get-deployments` 예시에서는 사용자 AWS 계정과 연결된 여러 배포에 대한 정보를 표시합니다.

```
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": "amzn-s3-demo-bucket",
                "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": "amzn-s3-demo-bucket",
                "key": "MyOtherApp.zip"
                }
            },
            "deploymentId": "d-A1B2C3222",
            "createTime": 1409764576.589,
            "completeTime": 1409764596.101
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 BatchGetDeployments 섹션을 참조하세요.

batch-get-on-premises-instances

다음 코드 예시에서는 `batch-get-on-premises-instances` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

하나 이상의 온프레미스 인스턴스에 대한 정보 가져오기

다음 `batch-get-on-premises-instances` 예시에서는 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"
        }
    ]
}
```


- API 세부 정보는 AWS CLI 명령 참조의 [BatchGetOnPremisesInstances](#) 섹션을 참조하세요.

continue-deployment

다음 코드 예시에서는 continue-deployment 코드를 사용하는 방법을 보여줍니다.

AWS CLI

지정된 대기 시간이 경과할 때까지 기다리지 않고 트래픽 경로 변경을 시작하는 방법

다음 continue-deployment 예시에서는 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 AWS CodeDeploy API 참조의 ContinueDeployment를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ContinueDeployment 섹션을 참조하세요.

create-application

다음 코드 예시에서는 create-application 코드를 사용하는 방법을 보여줍니다.

AWS CLI

애플리케이션 생성

다음 create-application 예시에서는 애플리케이션을 생성하고 이를 사용자 AWS 계정과 연결합니다.

```
aws deploy create-application --application-name MyOther_App
```

출력:

```
{
    "applicationId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateApplication 섹션을 참조하세요.

create-deployment-config

다음 코드 예시에서는 create-deployment-config 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용자 지정 배포 구성 생성

다음 create-deployment-config 예시에서는 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateDeploymentConfig 섹션을 참조하세요.

create-deployment-group

다음 코드 예시에서는 create-deployment-group 코드를 사용하는 방법을 보여줍니다.

AWS CLI

배포 그룹 생성

다음 create-deployment-group 예시에서는 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateDeploymentGroup 섹션을 참조하세요.

`create-deployment`

다음 코드 예시에서는 `create-deployment` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

예시 1: EC2/온프레미스 컴퓨팅 플랫폼을 사용하여 CodeDeploy 배포 생성

다음 `create-deployment` 예시에서는 배포를 생성하고 이를 사용자의 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=amzn-s3-demo-
bucket,bundleType=zip,eTag=dd56cfdEXAMPLE8e768f9d77fEXAMPLE,key=WordPressApp.zip
```

출력:

```
{
    "deploymentId": "d-A1B2C3111"
}
```

예시 2: Amazon ECS 컴퓨팅 플랫폼을 사용하여 CodeDeploy 배포 생성

다음 `create-deployment` 예시에서는 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 `ecs-deployment-bucket`이라는 S3 버킷에서 다음 파일 `appspec.yaml`을 검색합니다.

```
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 AWS CodeDeploy API 참조의 CreateDeployment를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateDeployment를 참조하세요.

delete-application

다음 코드 예시에서는 delete-application 코드를 사용하는 방법을 보여줍니다.

AWS CLI

애플리케이션 삭제

다음 delete-application 예시에서는 사용자의 AWS 계정과 연결된 지정된 애플리케이션을 삭제합니다.

```
aws deploy delete-application --application-name WordPress_App
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteApplication 섹션을 참조하세요.

delete-deployment-config

다음 코드 예시에서는 delete-deployment-config 코드를 사용하는 방법을 보여줍니다.

AWS CLI

배포 구성 삭제

다음 delete-deployment-config 예시에서는 사용자의 AWS 계정과 연결된 사용자 지정 배포 구성을 삭제합니다.

```
aws deploy delete-deployment-config --deployment-config-name ThreeQuartersHealthy
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteDeploymentConfig 섹션을 참조하세요.

delete-deployment-group

다음 코드 예시에서는 delete-deployment-group 코드를 사용하는 방법을 보여줍니다.

AWS CLI

배포 그룹 삭제

다음 delete-deployment-group 예시에서는 지정된 애플리케이션과 연결된 배포 그룹을 삭제합니다.

```
aws deploy delete-deployment-group \
    --application-name WordPress_App \
    --deployment-group-name WordPress_DG
```

출력:

```
{
    "hooksNotCleanedUp": []
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteDeploymentGroup 섹션을 참조하세요.

delete-git-hub-account-token

다음 코드 예시에서는 delete-git-hub-account-token 코드를 사용하는 방법을 보여줍니다.

AWS CLI

GitHub 계정 연결 삭제

다음 delete-git-hub-account-token 예시에서는 지정된 GitHub 계정의 연결을 삭제합니다.

```
aws deploy delete-git-hub-account-token --token-name my-github-account
```

출력:

```
{
    "tokenName": "my-github-account"
}
```

자세한 내용은 AWS CodeDeploy API 참조의 DeleteGitHubAccountToken을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteGitHubAccountToken 섹션을 참조하세요.

deregister-on-premises-instance

다음 코드 예시에서는 deregister-on-premises-instance 코드를 사용하는 방법을 보여줍니다.

AWS CLI

온프레미스 인스턴스 등록 취소

다음 `deregister-on-premises-instance` 예시에서는 AWS CodeDeploy 에 온프레미스 인스턴스를 등록 취소하지만 인스턴스와 연결된 IAM 사용자를 삭제하지 않으며 AWS CodeDeploy에서 온프레미스 인스턴스 태그의 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.

- API 세부 정보는 AWS CLI 명령 참조의 DeregisterOnPremisesInstance 섹션을 참조하세요.

deregister

다음 코드 예시에서는 `deregister` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

온프레미스 인스턴스 등록 취소

다음 `deregister` 예시에서는 온프레미스 인스턴스를 AWS CodeDeploy에 등록 취소합니다. 인스턴스와 연결된 IAM 사용자는 삭제되지 않습니다. AWS CodeDeploy에서 인스턴스에서 온프레미스 태그의 연결을 해제합니다. 인스턴스에서 AWS CodeDeploy Agent를 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 세부 정보는 AWS CLI 명령 참조의 Deregister 섹션을 참조하세요.

get-application-revision

다음 코드 예시에서는 get-application-revision 코드를 사용하는 방법을 보여줍니다.

AWS CLI

애플리케이션 개정에 대한 정보를 가져오는 방법

다음 get-application-revision 예시에서는 지정된 애플리케이션과 연결된 애플리케이션 개정에 대한 정보를 표시합니다.

```
aws deploy get-application-revision \
    --application-name WordPress_App \
    --s3-location bucket=amzn-s3-demo-
bucket,bundleType=zip,eTag=dd56cfdEXAMPLE8e768f9d77fEXAMPLE,key=WordPressApp.zip
```

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": "amzn-s3-demo-bucket",
            "key": "WordPressApp.zip"
        }
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetApplicationRevision 섹션을 참조하세요.

get-application

다음 코드 예시에서는 get-application 코드를 사용하는 방법을 보여줍니다.

AWS CLI

애플리케이션에 대한 정보를 가져오는 방법

다음 get-application 예시에서는 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
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetApplication 섹션을 참조하세요.

get-deployment-config

다음 코드 예시에서는 get-deployment-config 코드를 사용하는 방법을 보여줍니다.

AWS CLI

배포 구성에 대한 정보 가져오기

다음 get-deployment-config 예시에서는 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetDeploymentConfig 섹션을 참조하세요.

get-deployment-group

다음 코드 예시에서는 `get-deployment-group` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

배포 그룹에 대한 정보를 보는 방법

다음 `get-deployment-group` 예시에서는 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetDeploymentGroup 섹션을 참조하세요.

get-deployment-instance

다음 코드 예시에서는 get-deployment-instance 코드를 사용하는 방법을 보여줍니다.

AWS CLI

배포 인스턴스에 대한 정보를 가져오는 방법

다음 get-deployment-instance 예시에서는 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetDeploymentInstance 섹션을 참조하세요.

get-deployment-target

다음 코드 예시에서는 `get-deployment-target` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

배포 대상에 대한 정보 반환

다음 `get-deployment-target` 예시에서는 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 AWS CodeDeploy API 참조의 GetDeploymentTarget을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetDeploymentTarget 섹션을 참조하세요.

get-deployment

다음 코드 예시에서는 `get-deployment` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

배포에 대한 정보 가져오기

다음 `get-deployment` 예시에서는 사용자의 AWS 계정과 연결된 배포에 대한 정보를 표시합니다.

```
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": "amzn-s3-demo-bucket",
            "key": "WordPressApp.zip"
            }
        },
        "deploymentId": "d-A1B2C3123",
        "deploymentGroupName": "WordPress_DG",
        "createTime": 1409764576.589,
        "completeTime": 1409764596.101,
        "ignoreApplicationStopFailures": false
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetDeployment 섹션을 참조하세요.

`get-on-premises-instance`

다음 코드 예시에서는 `get-on-premises-instance` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

단일 온프레미스 인스턴스에 대한 정보 가져오기

다음 get-on-premises-instance 예시에서는 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetOnPremisesInstance 섹션을 참조하세요.

install

다음 코드 예시에서는 install 코드를 사용하는 방법을 보여줍니다.

AWS CLI

온프레미스 인스턴스 설치

다음 install 예시에서는 온프레미스 구성 파일을 인스턴스의 지정된 위치에서 AWS CodeDeploy 에이전트가 찾을 것으로 예상되는 인스턴스의 위치로 복사합니다. 또한 인스턴스에 AWS CodeDeploy 에이전트를 설치합니다. IAM 사용자를 생성하지 않으며, 온프레미스 인스턴스를 AWS CodeDeploy에 등록하거나 인스턴스에 대한 온프레미스 인스턴스 태그를 AWS CodeDeploy에 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
```

- API 세부 정보는 AWS CLI 명령 참조의 Install 섹션을 참조하세요.

list-application-revisions

다음 코드 예시에서는 `list-application-revisions` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

애플리케이션 개정에 대한 정보 가져오기

다음 `list-application-revisions` 예시에서는 지정된 애플리케이션과 관련된 모든 애플리케이션 수정본에 대한 정보를 표시합니다.

```
aws deploy list-application-revisions \
    --application-name WordPress_App \
    --s-3-bucket amzn-s3-demo-bucket \
    --deployed exclude \
    --s-3-key-prefix WordPress_ \
    --sort-by lastUsedTime \
    --sort-order descending
```

출력:

```
{
    "revisions": [
        {
            "revisionType": "S3",
            "s3Location": {
                "version": "uTecLusvCB_JqHFXtfUcyfV8bEXAMPLE",
                "bucket": "amzn-s3-demo-bucket",
```

```
                "key": "WordPress_App.zip",
                "bundleType": "zip"
            }
        },
        {
            "revisionType": "S3",
            "s3Location": {
                "version": "tMk.UxgDpMEVb7V187ZM6wVAWEXAMPLE",
                "bucket": "amzn-s3-demo-bucket",
                "key": "WordPress_App_2-0.zip",
                "bundleType": "zip"
            }
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListApplicationRevisions 섹션을 참조하세요.

list-applications

다음 코드 예시에서는 list-applications 코드를 사용하는 방법을 보여줍니다.

AWS CLI

애플리케이션에 대한 정보를 가져오는 방법

다음 list-applications 예시에서는 사용자의 AWS 계정과 연결된 모든 애플리케이션에 대한 정보를 표시합니다.

```
aws deploy list-applications
```

출력:

```
{
    "applications": [
        "WordPress_App",
        "MyOther_App"
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListApplications 섹션을 참조하세요.

list-deployment-configs

다음 코드 예시에서는 list-deployment-configs 코드를 사용하는 방법을 보여줍니다.

AWS CLI

배포 구성에 대한 정보 가져오기

다음 list-deployment-configs 예시에서는 사용자의 AWS 계정과 연결된 모든 배포 구성에 대한 정보를 표시합니다.

```
aws deploy list-deployment-configs
```

출력:

```
{
    "deploymentConfigsList": [
        "ThreeQuartersHealthy",
        "CodeDeployDefault.AllAtOnce",
        "CodeDeployDefault.HalfAtATime",
        "CodeDeployDefault.OneAtATime"
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListDeploymentConfigs 섹션을 참조하세요.

list-deployment-groups

다음 코드 예시에서는 list-deployment-groups 코드를 사용하는 방법을 보여줍니다.

AWS CLI

배포 그룹에 대한 정보 가져오기

다음 list-deployment-groups 예시에서는 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"
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListDeploymentGroups 섹션을 참조하세요.

list-deployment-instances

다음 코드 예시에서는 `list-deployment-instances` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

배포 인스턴스에 대한 정보를 가져오는 방법

다음 `list-deployment-instances` 예시에서는 지정된 배포와 관련된 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"
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListDeploymentInstances 섹션을 참조하세요.

list-deployment-targets

다음 코드 예시에서는 `list-deployment-targets` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

배포와 연결된 대상 IDs 목록을 검색하는 방법

다음 `list-deployment-targets` 예시에서는 상태가 'Failed' 또는 'InProgress'인 배포와 연결된 대상 ID의 목록을 검색합니다.

```
aws deploy list-deployment-targets \
    --deployment-id "d-A1B2C3111" \
    --target-filters "{\"TargetStatus\":[\"Failed\",\"InProgress\"]}"
```

출력:

```
{
    "targetIds": [
        "i-0f1558aaf90e5f1f9"
    ]
}
```

자세한 내용은 AWS CodeDeploy API 참조의 ListDeploymentTargets를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListDeploymentTargets 섹션을 참조하세요.

`list-deployments`

다음 코드 예시에서는 `list-deployments` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

배포에 대한 정보를 가져오는 방법

다음 `list-deployments` 예시에서는 지정된 애플리케이션 및 배포 그룹과 관련된 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"
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListDeployments를 참조하세요.

list-git-hub-account-token-names

다음 코드 예시에서는 `list-git-hub-account-token-names` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

GitHub 계정에 저장된 연결의 이름 나열

다음 `list-git-hub-account-token-names` 예시에서는 현재 AWS 사용자에 대한 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 AWS CodeDeploy API 참조의 ListGitHubAccountTokenNames를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListGitHubAccountTokenNames 섹션을 참조하세요.

list-on-premises-instances

다음 코드 예시에서는 `list-on-premises-instances` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

하나 이상의 온프레미스 인스턴스에 대한 정보 가져오기

다음 `list-on-premises-instances` 예시에서는 AWS CodeDeploy에 등록된 인스턴스에 대해 사용 가능한 온프레미스 인스턴스 이름 목록을 검색하고 또한 지정된 온프레미스 인스턴스 태그가 AWS CodeDeploy에서 인스턴스와 연결되어 있습니다.

```
aws deploy list-on-premises-instances \
    --registration-status Registered \
    --tag-filters Key=Name,Value=CodeDeployDemo-OnPrem,Type=KEY_AND_VALUE
```

출력:

```
{
    "instanceNames": [
        "AssetTag12010298EX"
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListOnPremisesInstances 섹션을 참조하세요.

`list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

리소스에 대한 태그를 나열하는 방법(애플리케이션)

다음 `list-tags-for-resource` 예시에서는 CodeDeploy의 testApp이라는 애플리케이션에 적용된 태그를 나열합니다.

```
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 Tagging instances for deployment groups in CodeDeploy를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource를 참조하세요.

push

다음 코드 예시에서는 push 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Amazon S3에 AWS CodeDeploy 호환 애플리케이션 개정을 번들링하고 배포하는 방법

다음 push 예시에서는 Amazon S3에 애플리케이션 개정을 번들링하고 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://amzn-s3-demo-bucket/WordPressApp.zip \
    --source /tmp/MyLocalDeploymentFolder/
```

출력은 create-deployment 명령을 사용하여 업로드된 애플리케이션 리비전을 사용하는 배포를 만드는 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=amzn-s3-demo-
bucket,key=WordPressApp.zip,bundleType=zip,eTag="cecc9b8EXAMPLE50a6e71fdb88EXAMPLE",version=
```

- API 세부 정보는 AWS CLI 명령 참조의 Push 섹션을 참조하세요.

register-application-revision

다음 코드 예시에서는 register-application-revision 코드를 사용하는 방법을 보여줍니다.

AWS CLI

이미 업로드된 애플리케이션 개정에 대한 정보를 등록하는 방법

다음 register-application-revision 예시에서는 AWS CodeDeploy 를 사용하여 Amazon S3에 저장된 이미 업로드된 애플리케이션 개정에 대한 정보를 등록합니다.

```
aws deploy register-application-revision \
    --application-name WordPress_App \
    --description "Revised WordPress application" \
    --s3-location bucket=amzn-s3-demo-
bucket,key=RevisedWordPressApp.zip,bundleType=zip,eTag=cecc9b8a08eac650a6e71fdb88EXAMPLE
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 RegisterApplicationRevision 섹션을 참조하세요.

register-on-premises-instance

다음 코드 예시에서는 register-on-premises-instance 코드를 사용하는 방법을 보여줍니다.

AWS CLI

온프레미스 인스턴스 등록

다음 register-on-premises-instance 예시에서는 온프레미스 인스턴스를 AWS CodeDeploy에 등록합니다. 지정된 IAM 사용자를 생성하지 않으며, 등록된 인스턴스에 대한 온프레미스 인스턴스 태그를 AWS CodeDeploy에서 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.

- API 세부 정보는 AWS CLI 명령 참조의 RegisterOnPremisesInstance 섹션을 참조하세요.

register

다음 코드 예시에서는 `register` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

온프레미스 인스턴스 등록

다음 `register` 예시에서는 AWS CodeDeploy에 온프레미스 인스턴스를 등록하고, AWS CodeDeploy에서 지정된 온프레미스 인스턴스 태그를 등록된 인스턴스와 연결하고, 인스턴스에 복사할 수 있는 온프레미스 구성 파일을 생성합니다. IAM 사용자를 생성하지 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 세부 정보는 AWS CLI 명령 참조의 Register 섹션을 참조하세요.

remove-tags-from-on-premises-instances

다음 코드 예시에서는 `remove-tags-from-on-premises-instances` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

하나 이상의 온프레미스 인스턴스에서 태그 제거

다음 `remove-tags-from-on-premises-instances` 예시에서는 AWS CodeDeploy에서 지정된 온프레미스 태그를 온프레미스 인스턴스에서 연결 해제합니다. AWS CodeDeploy에서 온프레

미스 인스턴스의 등록을 취소하거나 인스턴스에서 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.

- API 세부 정보는 AWS CLI 명령 참조의 RemoveTagsFromOnPremisesInstances 섹션을 참조하세요.

stop-deployment

다음 코드 예시에서는 `stop-deployment` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

배포 중지를 시도하는 방법

다음 `stop-deployment` 예시에서는 사용자의 AWS 계정과 연결된 진행 중인 배포를 중지하려고 시도합니다.

aws deploy stop-deployment --deployment-id d-A1B2C3111

출력:

```
{
    "status": "Succeeded",
    "statusMessage": "No more commands will be scheduled for execution in the
 deployment instances"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 StopDeployment 섹션을 참조하세요.

tag-resource

다음 코드 예시에서는 `tag-resource` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

리소스에 태그를 지정하는 방법(애플리케이션)

다음 `tag-resource` 예시에서는 두 개의 태그에 Name 및 Type 키와 testName 및 testType 값을 추가하여 CodeDeploy에서 testApp이라는 애플리케이션에 추가합니다.

```
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 Tagging instances for deployment groups in CodeDeploy를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

`uninstall`

다음 코드 예시에서는 `uninstall` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

온프레미스 인스턴스를 제거하는 방법

다음 `uninstall` 예시에서는 온프레미스 인스턴스에서 AWS CodeDeploy 에이전트를 제거하고 인스턴스에서 온프레미스 구성 파일을 제거합니다. AWS CodeDeploy에서 인스턴스를 등록 취소하거나, 인스턴스에서 AWS CodeDeploy의 온프레미스 인스턴스 태그를 연결 해제하거나, 인스턴스와 연결된 IAM 사용자를 삭제하지 않습니다.

```
aws deploy uninstall
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 Uninstall 섹션을 참조하세요.

`untag-resource`

다음 코드 예시에서는 `untag-resource` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

리소스에서 태그를 제거하는 방법(애플리케이션)

다음 untag-resource 예시에서는 CodeDeploy의 testApp이라는 애플리케이션에서 Name 및 Type 키가 있는 두 개의 태그를 제거합니다.

```
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 Tagging instances for deployment groups in CodeDeploy를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

update-application

다음 코드 예시에서는 update-application 코드를 사용하는 방법을 보여줍니다.

AWS CLI

애플리케이션의 세부 정보를 변경하는 방법

다음 update-application 예시에서는 사용자의 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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateApplication 섹션을 참조하세요.

update-deployment-group

다음 코드 예시에서는 update-deployment-group 코드를 사용하는 방법을 보여줍니다.

AWS CLI

배포 그룹에 대한 정보를 변경하는 방법

다음 update-deployment-group 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateDeploymentGroup 섹션을 참조하세요.

CodeGuru Reviewer examples using AWS CLI

다음 코드 예제에서는 CodeGuru Reviewer에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여 줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`associate-repository`

다음 코드 예시는 associate-repository의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: Bitbucket 리포지토리 연결을 만드는 방법

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 CodeGuru Reviewer에서 Bitbucket 리포지토리 연결 생성을 참조하세요.

예제 2: GitHub Enterprise 리포지토리 연결을 만드는 방법

다음 `associate-repository` 예제에서는 기존 GitHub Enterprise 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 Reviewer 사용 설명서의 Amazon CodeGuru Reviewer에서 GitHub Enterprise Server 리포지토리 연결 생성을 참조하세요.

예제 3: AWS CodeCommit 리포지토리 연결을 생성하는 방법

다음 `associate-repository` 예제에서는 기존 AWS CodeCommit 리포지토리를 사용하여 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 CodeGuru Reviewer에서 AWS CodeCommit 리포지토리 연결 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AssociateRepository 섹션을 참조하세요.

create-code-review

다음 코드 예시는 `create-code-review`의 사용 방법을 보여 줍니다.

AWS CLI

코드 검토를 생성하는 방법

다음 `create-code-review` 코드는 이름이 `my-repository-name`인 AWS CodeCommit 리포지토리의 `mainline` 브랜치에 코드 검토를 생성합니다.

```
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 CodeGuru Reviewer에서 코드 검토 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateCodeReview 섹션을 참조하세요.

`describe-code-review`

다음 코드 예시는 `describe-code-review`의 사용 방법을 보여 줍니다.

AWS CLI

코드 검토에 대한 세부 정보를 나열합니다.

다음 `describe-code-review` 코드는 'my-repo-name'이라는 이름이 붙은 AWS CodeCommit 리포지토리의 'mainline' 브랜치의 코드 검토에 대한 정보를 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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeCodeReview 섹션을 참조하세요.

describe-recommendation-feedback

다음 코드 예시는 describe-recommendation-feedback의 사용 방법을 보여 줍니다.

AWS CLI

권장 사항에 대한 피드백 정보를 보는 방법

다음 describe-recommendation-feedback 코드는 권장 사항에 대한 피드백에 대한 정보를 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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeRecommendationFeedback 섹션을 참조하세요.

`describe-repository-association`

다음 코드 예시는 `describe-repository-association`의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: GitHub 리포지토리 연결에 대한 정보를 반환하는 방법

다음 `describe-repository-association` 예제에서는 GitHub Enterprise 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 CodeGuru Reviewer에서 GitHub Enterprise Server 리포지토리 연결 생성을 참조하세요.

예제 2: 실패한 리포지토리 연결에 대한 정보를 반환하는 방법

다음 `describe-repository-association` 예제에서는 GitHub Enterprise 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 CodeGuru Reviewer에서 GitHub Enterprise Server 리포지토리 연결 생성을 참조하세요.

예제 3: 연결 해제 리포지토리 연결에 대한 정보를 반환하는 방법

다음 `describe-repository-association` 예제에서는 GitHub Enterprise 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 CodeGuru Reviewer에서 GitHub Enterprise Server 리포지토리 연결 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeRepositoryAssociation 섹션을 참조하세요.

`disassociate-repository`

다음 코드 예시는 `disassociate-repository`의 사용 방법을 보여 줍니다.

AWS CLI

리포지토리 연결을 해제하는 방법

다음 `disassociate-repository`는 AWS CodeCommit 리포지토리를 사용하는 리포지토리 연결을 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 CodeGuru Reviewer에서 리포지토리 연결 해제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisassociateRepository 섹션을 참조하세요.

list-code-reviews

다음 코드 예시는 `list-code-reviews`의 사용 방법을 보여 줍니다.

AWS CLI

지난 90일 동안 AWS 계정에 생성된 코드 검토를 나열하는 방법.

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.

- API 세부 정보는 AWS CLI 명령 참조의 ListCodeReviews 섹션 섹션을 참조하세요.

`list-recommendation-feedback`

다음 코드 예시는 `list-recommendation-feedback`의 사용 방법을 보여 줍니다.

AWS CLI

연결된 리포지토리의 권장 사항에 대한 고객 권장 사항 피드백을 나열하는 방법

다음 `list-recommendation-feedback` 코드는 코드 검토에 대한 모든 권장 사항에 대한 고객 피드백을 나열합니다. 이 코드 검토에는 고객의 피드백인 “ThumbsUp”이 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListRecommendationFeedback 섹션을 참조하세요.

`list-recommendations`

다음 코드 예시는 `list-recommendations`의 사용 방법을 보여 줍니다.

AWS CLI

완료된 코드 검토에 대한 권장 사항을 나열하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 ListRecommendations를 참조하세요.

`list-repository-associations`

다음 코드 예시는 `list-repository-associations`의 사용 방법을 보여 줍니다.

AWS CLI

AWS 계정의 리포지토리 연결을 나열하는 방법

다음 `list-repository-associations` 예제에서는 계정의 리포지토리 연결 요약 객체 목록을 반환합니다. `ProviderType`, `Name`, `State` 및 `Owner`를 기준으로 반환된 목록을 필터링할 수 있습니다.

```
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 CodeGuru Reviewer에서 모든 리포지토리 연결 보기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListRepositoryAssociations 섹션을 참조하세요.

`list-tags-for-resource`

다음 코드 예시는 `list-tags-for-resource`의 사용 방법을 보여 줍니다.

AWS CLI

연결된 리포지토리의 태그를 나열하는 방법

다음 `list-tags-for-resource` 코드는 리포지토리와 연결된 태그를 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 CodeGuru Reviewer와 연결된 리포지토리에 대한 태그 보기(AWS CLI)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource를 참조하세요.

put-recommendation-feedback

다음 코드 예시는 put-recommendation-feedback의 사용 방법을 보여 줍니다.

AWS CLI

코드 검토에 권장 사항을 추가하는 방법

다음 put-recommendation-feedback 코드는 코드 검토에 ThumbsUp 권장 사항을 표시합니다.

```
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 CodeGuru Reviewer 사용 설명서의 4단계: 피드백 제공을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutRecommendationFeedback 섹션을 참조하세요.

tag-resource

다음 코드 예시는 tag-resource의 사용 방법을 보여 줍니다.

AWS CLI

연결된 리포지토리에 태그를 추가하는 방법

다음 tag-resource 코드는 연결된 리포지토리에 두 개의 태그를 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 CodeGuru Reviewer 연결 리포지토리에 태그 추가(AWS CLI) 및 CodeGuru Reviewer 연결 리포지토리에 대한 태그 추가 또는 업데이트(AWS CLI)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

`untag-resource`

다음 코드 예시는 `untag-resource`의 사용 방법을 보여 줍니다.

AWS CLI

연결된 리포지토리의 태그를 해제하는 방법

다음 `untag-resource` 코드는 연결된 리포지토리에서 'Secret' 및 'Team' 키가 있는 태그 2개를 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 CodeGuru Reviewer와 연결된 리포지토리에서 태그 제거(AWS CLI)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

AWS CLI를 사용한 CodePipeline 예제

다음 코드 예제에서는 CodePipeline에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

acknowledge-job

다음 코드 예시는 `acknowledge-job`의 사용 방법을 보여 줍니다.

AWS CLI

지정된 작업에 대한 정보를 검색하는 방법

이 예제에서는 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 poll-for-jobs를 사용합니다.

명령:

```
aws codepipeline acknowledge-job --job-id f4f4ff82-2d11-EXAMPLE --nonce 3
```

출력:

```
{
  "status": "InProgress"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 AcknowledgeJob을 참조하세요.

create-custom-action-type

다음 코드 예시는 `create-custom-action-type`의 사용 방법을 보여 줍니다.

AWS CLI

사용자 지정 작업을 만들려면

이 예제에서는 사용자 지정 작업의 구조가 포함된 이미 생성된 JSON 파일(여기서는 MyCustomAction.json으로 이름 지정)을 사용하여 AWS CodePipeline에 대한 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 `MyCustomAction.json`의 콘텐츠:

```
{
    "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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateCustomActionType을 참조하세요.

create-pipeline

다음 코드 예시는 create-pipeline의 사용 방법을 보여 줍니다.

AWS CLI

파이프라인을 만들려면

이 예제에서는 파이프라인의 구조를 포함하는 이미 생성된 JSON 파일(여기서는 MySecondPipeline.json으로 이름 지정)을 사용하여 AWS CodePipeline에 파이프라인을 생성합니다. 파일 구조를 포함하여 파이프라인을 생성하기 위한 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.
```

- API 세부 정보는 AWS CLI 명령 참조의 CreatePipeline을 참조하세요.

delete-custom-action-type

다음 코드 예시는 `delete-custom-action-type`의 사용 방법을 보여 줍니다.

AWS CLI

사용자 지정 작업을 삭제하려면

이 예제에서는 삭제할 작업의 작업 유형, 제공업체 이름 및 버전 번호가 포함된 이미 생성된 JSON 파일(여기서는 DeleteMyCustomAction.json으로 이름 지정)을 사용하여 AWS CodePipeline에서 사용자 지정 작업을 삭제합니다. list-action-types 명령을 사용하여 범주, 버전 및 제공업체에 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.
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteCustomActionType을 참조하세요.

delete-pipeline

다음 코드 예시는 delete-pipeline의 사용 방법을 보여 줍니다.

AWS CLI

파이프라인을 삭제하려면

이 예제에서는 AWS CodePipeline에서 MySecondPipeline 파이프라인을 삭제합니다. list-pipelines 명령을 사용하여 AWS 계정과 연결된 파이프라인 목록을 봅니다.

명령:

```
aws codepipeline delete-pipeline --name MySecondPipeline
```

출력:

```
None.
```

- API 세부 정보는 AWS CLI 명령 참조의 DeletePipeline을 참조하세요.

delete-webhook

다음 코드 예시는 delete-webhook의 사용 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 Delete the webhook for your GitHub source를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteWebhook](#)를 참조하세요.

deregister-webhook-with-third-party

다음 코드 예시는 `deregister-webhook-with-third-party`의 사용 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 [Delete the webhook for your GitHub source](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DeregisterWebhookWithThirdParty](#)를 참조하세요.

disable-stage-transition

다음 코드 예시는 `disable-stage-transition`의 사용 방법을 보여 줍니다.

AWS CLI

파이프라인에서 스테이지로의 전환을 비활성화하려면

이 예제에서는 AWS CodePipeline에서 MyFirstPipeline 파이프라인의 베타 단계로의 전환을 비활성화합니다.

명령:

```
aws codepipeline disable-stage-transition --pipeline-name MyFirstPipeline --stage-name Beta  --transition-type Inbound
```

출력:

```
None.
```

- API 세부 정보는 AWS CLI 명령 참조의 [DisableStageTransition](https://docs.aws.amazon.com/goto/aws-cli/codepipeline-2015-07-09/DisableStageTransition)을 참조하세요.

`enable-stage-transition`

다음 코드 예시는 `enable-stage-transition`의 사용 방법을 보여 줍니다.

AWS CLI

파이프라인에서 스테이지로의 전환을 활성화하려면

이 예제에서는 AWS CodePipeline에서 MyFirstPipeline 파이프라인의 베타 단계로의 전환을 활성화합니다.

명령:

```
aws codepipeline enable-stage-transition --pipeline-name MyFirstPipeline --stage-name Beta  --transition-type Inbound
```

출력:

```
None.
```

- API 세부 정보는 AWS CLI 명령 참조의 [EnableStageTransition](https://docs.aws.amazon.com/goto/aws-cli/codepipeline-2015-07-09/EnableStageTransition)을 참조하세요.

`get-job-details`

다음 코드 예시는 `get-job-details`의 사용 방법을 보여 줍니다.

AWS CLI

작업의 세부 정보를 가져오려면

이 예제에서는 ID가 f4f4ff82-2d11-EXAMPLE로 표시되는 작업에 대한 세부 정보를 반환합니다. 이 명령은 사용자 지정 작업에만 사용됩니다. 이 명령이 직접 호출되면 AWS CodePipeline은 사용자 지정 작업에 필요한 경우 파이프라인의 아티팩트를 저장하는 데 사용되는 Amazon S3 버킷에 대한 임시 자격 증명을 반환합니다. 이 명령은 작업에 대해 정의된 보안 암호 값이 정의된 경우에도 해당 값을 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"
 }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetJobDetails를 참조하세요.

get-pipeline-state

다음 코드 예시는 get-pipeline-state의 사용 방법을 보여 줍니다.

AWS CLI

파이프라인 상태에 대한 정보를 가져오려면

이 예제에서는 MyFirstPipeline이라는 파이프라인의 최신 상태를 반환합니다.

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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetPipelineState를 참조하세요.

get-pipeline

다음 코드 예시는 `get-pipeline`의 사용 방법을 보여 줍니다.

AWS CLI

파이프라인의 구조를 보려면

이 예제에서는 MyFirstPipeline이라는 파이프라인의 구조를 반환합니다.

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
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetPipeline을 참조하세요.

list-action-executions

다음 코드 예시는 `list-action-executions`의 사용 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 볼 수 있습니다.

```
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": "amzn-s3-demo-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 View action executions (CLI)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListActionExecutions를 참조하세요.

list-action-types

다음 코드 예시는 `list-action-types`의 사용 방법을 보여 줍니다.

AWS CLI

사용 가능한 작업 유형을 보려면

단독으로 사용되는 list-action-types 명령은 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}"
            }
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListActionTypes를 참조하세요.

`list-pipeline-executions`

다음 코드 예시는 `list-pipeline-executions`의 사용 방법을 보여 줍니다.

AWS CLI

파이프라인 실행 내역을 보려면

다음 `list-pipeline-executions` 예제는 AWS 계정의 파이프라인에 대한 파이프라인 실행 내역을 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 View execution history를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListPipelineExecutions를 참조하세요.

list-pipelines

다음 코드 예시는 `list-pipelines`의 사용 방법을 보여 줍니다.

AWS CLI

파이프라인의 목록을 보려면

이 예제에서는 사용자의 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
      }
      ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListPipelines를 참조하세요.

`list-tags-for-resource`

다음 코드 예시는 `list-tags-for-resource`의 사용 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 View tags for a pipeline (CLI)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource를 참조하세요.

list-webhooks

다음 코드 예시는 `list-webhooks`의 사용 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/
trigger111111111EXAMPLE11111111111111111": {
                "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 List webhooks in your account를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListWebhooks를 참조하세요.

poll-for-jobs

다음 코드 예시는 poll-for-jobs의 사용 방법을 보여 줍니다.

AWS CLI

사용 가능한 작업을 보려면

이 예제에서는 작업 작업자가 수행할 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 직접 호출되면 AWS CodePipeline은 파이프라인의 아티팩트를 저장하는 데 사용되는 Amazon S3 버킷에 대한 임시 자격 증명을 반환합니다. 이 명령은 작업에 대해 정의된 보안 암호 값이 정의된 경우에도 해당 값을 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"
    }
  ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 PollForJobs를 참조하세요.

put-webhook

다음 코드 예시는 put-webhook의 사용 방법을 보여 줍니다.

AWS CLI

웹후크를 만들려면

다음 put-webhook 예제에서는 GitHub 버전 1 소스 작업에 대한 웹후크를 생성합니다. 웹후크를 만든 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 Create a webhook for a GitHub source를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutWebhook를 참조하세요.

retry-stage-execution

다음 코드 예시는 retry-stage-execution의 사용 방법을 보여 줍니다.

AWS CLI

실패한 작업을 재시도하려면

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 Retry failed actions (CLI)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RetryStageExecution을 참조하세요.

start-pipeline-execution

다음 코드 예시는 start-pipeline-execution의 사용 방법을 보여 줍니다.

AWS CLI

파이프라인을 통해 최신 개정을 실행하려면

이 예제에서는 'MyFirstPipeline'이라는 파이프라인을 통해 파이프라인의 소스 단계에 있는 최신 개정을 실행합니다.

명령:

```
aws codepipeline start-pipeline-execution --name MyFirstPipeline
```

출력:

```
{
  "pipelineExecutionId": "3137f7cb-7cf7-EXAMPLE"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 StartPipelineExecution을 참조하세요.

stop-pipeline-execution

다음 코드 예시는 stop-pipeline-execution의 사용 방법을 보여 줍니다.

AWS CLI

파이프라인 실행을 중지하려면

다음 stop-pipeline-execution 예제에서는 기본적으로 진행 중인 작업이 완료될 때까지 기다렸다가 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 Stop a pipeline execution (CLI)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StopPipelineExecution을 참조하세요.

tag-resource

다음 코드 예시는 tag-resource의 사용 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 Add tags to a pipeline (CLI)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

untag-resource

다음 코드 예시는 untag-resource의 사용 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 Remove tags from a pipeline (CLI)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

update-pipeline

다음 코드 예시는 update-pipeline의 사용 방법을 보여 줍니다.

AWS CLI

파이프라인 구조를 업데이트하려면

이 예제에서는 update-pipeline 명령을 --cli-input-json 인수와 함께 사용합니다. 이 예제에서는 미리 정의된 JSON 파일(MyFirstPipeline.json)을 사용하여 파이프라인 구조를 업데이트합니다. AWS CodePipeline은 JSON 파일에 포함된 파이프라인 이름을 인식한 다음, 파이프라인 구조의 수정된 필드의 변경 사항을 적용하여 파이프라인을 업데이트합니다.

미리 정의된 JSON 파일을 만들 때는 다음 지침을 따르세요.

get-pipeline 명령을 사용하여 검색된 파이프라인 구조로 작업하는 경우 JSON 파일의 파이프라인 구조에서 메타데이터 섹션을 제거해야 합니다('메타데이터': { } 라인 및 '생성됨', 'pipelineARN', '업데이트됨' 필드). 파이프라인 이름은 변경할 수 없습니다.

명령:

```
aws codepipeline update-pipeline --cli-input-json file://MyFirstPipeline.json
```

샘플 JSON 파일 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-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
    }
  }
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdatePipeline을 참조하세요.

AWS CodeStar를 사용한 Notifications 예제AWS CLI

다음 코드 예제에서는 AWS CodeStar Notifications에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하고 개별 서비스 작업을 수행하는 방법을 보여 줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`create-notification-rule`

다음 코드 예시는 `create-notification-rule`의 사용 방법을 보여 줍니다.

AWS CLI

알림 규칙을 생성하는 방법

다음 `create-notification-rule` 예제에서는 `rule.json`이라는 JSON 파일을 사용하여 지정된 AWS 계정에 이름이 지정된 리포지토리 `MyDemoRepo`에 대한 `MyNotificationRule`이라는 알림 규칙을 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 Create a Notification rule을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateNotificationRule을 참조하세요.

delete-notification-rule

다음 코드 예시는 delete-notification-rule의 사용 방법을 보여 줍니다.

AWS CLI

알림 규칙을 삭제하는 방법

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 Delete a Notification Rule를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteNotificationRule을 참조하세요.

delete-target

다음 코드 예시는 delete-target의 사용 방법을 보여 줍니다.

AWS CLI

알림 규칙 대상을 삭제하는 방법

다음 delete-target 예제에서는 지정된 대상을 대상으로 사용하도록 구성된 모든 알림 규칙에서 지정된 대상을 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 Delete a Notification Rule Target를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteTarget을 참조하세요.

describe-notification-rule

다음 코드 예시는 describe-notification-rule의 사용 방법을 보여 줍니다.

AWS CLI

알림 규칙의 세부 정보를 검색하는 방법

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 View Notification Rules를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeNotificationRule을 참조하세요.

`list-event-types`

다음 코드 예시는 `list-event-types`의 사용 방법을 보여 줍니다.

AWS CLI

알림 규칙의 이벤트 유형을 나열하는 방법

다음 `list-event-types` 예제에서는 CodeDeploy 애플리케이션에 사용 가능한 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 Create a Notification Rule을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListEventTypes를 참조하세요.

list-notification-rules

다음 코드 예시는 `list-notification-rules`의 사용 방법을 보여 줍니다.

AWS CLI

알림 규칙 목록을 검색하는 방법

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 View Notification Rules를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListNotificationRules를 참조하세요.

list-tags-for-resource

다음 코드 예시는 list-tags-for-resource의 사용 방법을 보여 줍니다.

AWS CLI

알림 규칙에 연결된 태그 목록을 가져오는 방법

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 Create a Notification Rule을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource를 참조하세요.

list-targets

다음 코드 예시는 list-targets의 사용 방법을 보여 줍니다.

AWS CLI

알림 규칙 대상 목록을 검색하는 방법

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 View Notification Rule Targets를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTargets를 참조하세요.

subscribe

다음 코드 예시는 subscribe의 사용 방법을 보여 줍니다.

AWS CLI

알림 규칙에 대상을 추가하는 방법

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 Add or Remove an Amazon SNS Topic as a Target for a Notification Rule을 참조하세요.

- API 세부 정보는 AWS CLI명령 참조의 Subscribe를 참조하세요.

tag-resource

다음 코드 예시는 tag-resource의 사용 방법을 보여 줍니다.

AWS CLI

알림 규칙에 태그를 추가하는 방법

다음 tag-resource 예제에서는 지정된 알림 규칙에 Team의 키 이름 및 값인 Li_Juan을 갖는 태그를 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 Create a Notification Rule을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

unsubscribe

다음 코드 예시는 unsubscribe의 사용 방법을 보여 줍니다.

AWS CLI

알림 규칙에서 대상을 제거하는 방법

다음 `unsubscribe` 예제에서는 지정된 알림 규칙의 대상으로 Amazon SNS 주제를 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 Add or Remove an Amazon SNS Topic as a Target for a Notification Rule을 참조하세요.

- API 세부 정보는 AWS CLI명령 참조의 Unsubscribe를 참조하세요.

`untag-resource`

다음 코드 예시는 `untag-resource`의 사용 방법을 보여 줍니다.

AWS CLI

알림 규칙에서 태그를 제거하는 방법

다음 `untag-resource` 예제에서는 지정된 알림 규칙에서 키 이름이 `Team`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 Edit a Notification Rule을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

update-notification-rule

다음 코드 예시는 update-notification-rule의 사용 방법을 보여 줍니다.

AWS CLI

알림 규칙을 업데이트하는 방법

다음 update-notification-rule 예제에서는 update.json이라는 JSON 파일을 사용하여 AWS 계정 123456789012에 이름이 MyNotificationRule인 알림 규칙을 업데이트합니다.

```
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 Edit a Notification Rule을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateNotificationRule을 참조하세요.

AWS CLI를 사용한 CodeConnections 예제

다음 코드 예제에서는 CodeConnections에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여 줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`create-connection`

다음 코드 예시는 `create-connection`의 사용 방법을 보여 줍니다.

AWS CLI

연결을 생성하는 방법

다음 `create-connection` 예제에서는 타사 리포지토리에 대한 연결을 생성하는 방법을 보여줍니다. 이 예제에서는 타사 공급자가 Bitbucket인 연결을 만듭니다.

AWS CLI 또는 AWS CloudFormation을 통해 생성된 연결은 기본적으로 Pending 상태입니다. CLI 또는 AWS CloudFormation으로 연결을 만든 후 콘솔을 사용하여 연결을 편집하여 Available 상태로 설정합니다.

```
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 Create a connection을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateConnection을 참조하세요.

create-host

다음 코드 예시는 create-host의 사용 방법을 보여 줍니다.

AWS CLI

호스트를 생성하는 방법

다음 create-host 예제에서는 타사 제공업체가 설치된 인프라의 엔드포인트를 나타내는 호스트를 만드는 방법을 보여 줍니다. 이 예제에서는 타사에 설치된 공급자가 GitHub Enterprise Server인 호스트를 생성합니다.

AWS CLI를 통해 생성된 호스트는 기본적으로 Pending 상태입니다. CLI를 사용하여 호스트를 생성한 후 콘솔 또는 CLI를 통해 호스트를 설정하여 호스트를 상태를 Available로 전환합니다.

```
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 Create a host (CLI)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateHost를 참조하세요.

delete-connection

다음 코드 예시는 delete-connection의 사용 방법을 보여 줍니다.

AWS CLI

연결을 삭제하려면

다음 delete-connection 예제는 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 Delete a connection (CLI)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteConnection을 참조하세요.

delete-host

다음 코드 예시는 delete-host의 사용 방법을 보여 줍니다.

AWS CLI

호스트를 삭제하는 방법

다음 delete-host 예제는 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 Delete a host (CLI)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteHost를 참조하세요.

get-connection

다음 코드 예시는 get-connection의 사용 방법을 보여 줍니다.

AWS CLI

연결에 대한 정보를 가져오는 방법

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 View connection details를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetConnection을 참조하세요.

get-host

다음 코드 예시는 get-host의 사용 방법을 보여 줍니다.

AWS CLI

호스트에 대한 정보를 가져오는 방법

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 View host details (CLI)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetHost를 참조하세요.

list-connections

다음 코드 예시는 `list-connections`의 사용 방법을 보여 줍니다.

AWS CLI

연결을 나열하는 방법

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 List connections (CLI)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListConnections를 참조하세요.

list-hosts

다음 코드 예시는 list-hosts의 사용 방법을 보여 줍니다.

AWS CLI

호스트를 나열하는 방법

다음 list-hosts 예제에서는 계정에서 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 List hosts (CLI)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListHosts를 참조하세요.

list-tags-for-resource

다음 코드 예시는 `list-tags-for-resource`의 사용 방법을 보여 줍니다.

AWS CLI

태그를 나열하는 방법

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 View tags for a connections resource를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource를 참조하세요.

`tag-resource`

다음 코드 예시에서는 `tag-resource`의 사용 방법을 보여줍니다.

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 Add tags to a connections resource를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

`untag-resource`

다음 코드 예시는 `untag-resource`의 사용 방법을 보여 줍니다.

AWS CLI

연결 리소스에서 AWS 태그를 제거하는 방법

다음 `untag-resource`에서는 지정된 리소스에서 태그를 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 Remove tags from a connections resource를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

AWS CLI를 사용한 Amazon Cognito ID 예제

다음 코드 예제에서는 Amazon Cognito 자격 증명에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여 줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`create-identity-pool`

다음 코드 예시는 `create-identity-pool`의 사용 방법을 보여 줍니다.

AWS CLI

Cognito 자격 증명 풀 공급자를 사용하여 자격 증명 풀 생성

이 예시에서는 MyIdentityPool이라는 자격 증명 풀을 생성합니다. 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
      }
  ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateIdentityPool을 참조합니다.

delete-identities

다음 코드 예시는 delete-identities의 사용 방법을 보여 줍니다.

AWS CLI

ID 풀 삭제

이 예제에서는 ID 풀을 삭제합니다.

명령:

```
aws cognito-identity delete-identity-pool --identity-ids-to-delete "us-west-2:11111111-1111-1111-1111-111111111111"
```

출력:

```
{
  "UnprocessedIdentityIds": []
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteIdentities를 참조하세요.

delete-identity-pool

다음 코드 예시는 delete-identity-pool의 사용 방법을 보여 줍니다.

AWS CLI

ID 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.

- API 세부 정보는 AWS CLI 명령 참조에서 [DeleteIdentityPool](#)을 참조하세요.

`describe-identity-pool`

다음 코드 예시는 `describe-identity-pool`의 사용 방법을 보여 줍니다.

AWS CLI

자격 증명 풀을 설명하는 방법

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
      }
  ]
```

```
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeIdentityPool](https://docs.aws.amazon.com/goto/aws-cli/cognito-identity-2014-06-30/DescribeIdentityPool) 섹션을 참조하세요.

get-identity-pool-roles

다음 코드 예시는 `get-identity-pool-roles`의 사용 방법을 보여 줍니다.

AWS CLI

자격 증명 풀 역할을 가져오는 방법

이 예제에서는 자격 증명 풀이 나열되어 있습니다.

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"
  }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [GetIdentityPoolRoles](https://docs.aws.amazon.com/goto/aws-cli/cognito-identity-2014-06-30/GetIdentityPoolRoles) 섹션을 참조하세요.

list-identity-pools

다음 코드 예시는 `list-identity-pools`의 사용 방법을 보여 줍니다.

AWS CLI

ID 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"
      }
  ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListIdentityPools 섹션을 참조하세요.

set-identity-pool-roles

다음 코드 예시는 set-identity-pool-roles의 사용 방법을 보여 줍니다.

AWS CLI

자격 증명 풀 역할을 설정하는 방법

다음 set-identity-pool-roles 예제에서는 자격 증명 풀의 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"
```

- API 세부 정보는 AWS CLI 명령 참조의 SetIdentityPoolRoles 섹션을 참조하세요.

`update-identity-pool`

다음 코드 예시는 `update-identity-pool`의 사용 방법을 보여 줍니다.

AWS CLI

자격 증명 풀을 생성하는 방법

이 예제에서는 자격 증명 풀을 업데이트합니다. 이름을 MyIdentityPool로 설정합니다. 자격 증명 공급자로 Cognito를 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
      }
  ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateIdentityPool 섹션을 참조하세요.

AWS CLI를 사용한 Amazon Cognito 자격 증명 공급자 예시

다음 코드 예는 Amazon Cognito 자격 증명 공급자와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

add-custom-attributes

다음 코드 예시에서는 add-custom-attributes 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용자 지정 속성 추가

이 예시에서는 사용자 지정 속성 CustomAttr1을 사용자 풀에 추가합니다. 문자열 유형이며 최소 1자에서 최대 15자까지 입력해야 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
```

- API 세부 정보는 AWS CLI 명령 참조의 AddCustomAttributes 섹션을 참조하세요.

admin-add-user-to-group

다음 코드 예시는 admin-add-user-to-group의 사용 방법을 보여 줍니다.

AWS CLI

그룹에 사용자 추가

이 예시에서는 사용자 Jane을 MyGroup 그룹에 추가합니다.

명령:

```
aws cognito-idp admin-add-user-to-group --user-pool-id us-west-2_aaaaaaaaa --
username Jane --group-name MyGroup
```

- API 세부 정보는 AWS CLI 명령 참조의 AdminAddUserToGroup 섹션을 참조하세요.

admin-confirm-sign-up

다음 코드 예시에서는 admin-confirm-sign-up 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용자 등록을 확인하는 방법

이 예시에서는 사용자 jane@example.com을 확인합니다.

명령:

```
aws cognito-idp admin-confirm-sign-up --user-pool-id us-west-2_aaaaaaaaa --
username jane@example.com
```

- API 세부 정보는 AWS CLI 명령 참조의 AdminConfirmSignUp 섹션을 참조하세요.

admin-create-user

다음 코드 예시에서는 admin-create-user 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용자 생성

다음 admin-create-user 예시에서는 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 AdminCreateUser 섹션을 참조하세요.

admin-delete-user-attributes

다음 코드 예시에서는 admin-delete-user-attributes 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용자 속성을 삭제하는 방법

이 예시에서는 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"
```

- API 세부 정보는 AWS CLI 명령 참조의 AdminDeleteUserAttributes 섹션을 참조하세요.

admin-delete-user

다음 코드 예시에서는 admin-delete-user 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용자 삭제

이 예제에서는 사용자를 삭제합니다.

명령:

```
aws cognito-idp admin-delete-user --user-pool-id us-west-2_aaaaaaaaa --
username diego@example.com
```

- API 세부 정보는 AWS CLI 명령 참조의 AdminDeleteUser 섹션을 참조하세요.

admin-disable-provider-for-user

다음 코드 예시는 admin-disable-provider-for-user의 사용 방법을 보여 줍니다.

AWS CLI

로컬 사용자 프로파일에서 페더레이션 사용자를 연결 해제하려면

다음 admin-disable-provider-for-user 예제에서는 연결된 로컬 프로파일에서 Google 사용자의 연결을 해제합니다.

```
aws cognito-idp admin-disable-provider-for-user \
    --user-pool-id us-west-2_EXAMPLE \
    --
user ProviderAttributeName=Cognito_Subject,ProviderAttributeValue=0000000000000000,ProviderN
```

자세한 내용은 Amazon Cognito 개발자 안내서의 페더레이션 사용자를 기존 사용자 프로파일에 연결을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AdminDisableProviderForUser 섹션을 참조하세요.

admin-disable-user

다음 코드 예시는 admin-disable-user의 사용 방법을 보여 줍니다.

AWS CLI

사용자의 로그인을 방지하려면

다음 `admin-disable-user` 예제에서는 `diego@example.com` 사용자의 로그인을 방지합니다.

```
aws cognito-idp admin-disable-user \
    --user-pool-id us-west-2_EXAMPLE \
    --username diego@example.com
```

자세한 정보는 Amazon Cognito 개발자 안내서의 사용자 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AdminDisableUser를 참조하세요.

admin-enable-user

다음 코드 예시는 `admin-enable-user`의 사용 방법을 보여 줍니다.

AWS CLI

사용자의 로그인을 활성화하려면

다음 `admin-enable-user` 예제에서는 사용자 diego@example.com의 로그인을 활성화합니다.

```
aws cognito-idp admin-enable-user \
    --user-pool-id us-west-2_EXAMPLE \
    --username diego@example.com
```

자세한 정보는 Amazon Cognito 개발자 안내서의 사용자 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AdminEnableUser를 참조하세요.

admin-forget-device

다음 코드 예시는 `admin-forget-device`의 사용 방법을 보여 줍니다.

AWS CLI

디바이스 지우기

이 예시에서는 사용자 이름 jane@example.com의의 디바이스를 잊어버립니다.

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
```

- API 세부 정보는 AWS CLI 명령 참조의 [AdminForgetDevice](#) 섹션을 참조하세요.

admin-get-device

다음 코드 예시에서는 admin-get-device 코드를 사용하는 방법을 보여줍니다.

AWS CLI

디바이스를 가져오는 방법

다음 admin-get-device 예제에서는 diego 사용자에 대한 디바이스 하나를 표시합니다.

```
aws cognito-idp admin-get-device \
    --user-pool-id us-west-2_EXAMPLE \
    --username diego \
    --device-key us-west-2_a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

출력:

```
{
    "Device": {
        "DeviceKey": "us-west-2_a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "DeviceAttributes": [
            {
                "Name": "device_status",
                "Value": "valid"
            },
            {
                "Name": "device_name",
                "Value": "MyDevice"
            },
            {
                "Name": "dev:device_arn",
                "Value": "arn:aws:cognito-idp:us-west-2:123456789012:owner/diego.us-
west-2_EXAMPLE/device/us-west-2_a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
            },
            {
                "Name": "dev:device_owner",
                "Value": "diego.us-west-2_EXAMPLE"
```

```
            },
            {
                "Name": "last_ip_used",
                "Value": "192.0.2.1"
            },
            {
                "Name": "dev:device_remembered_status",
                "Value": "remembered"
            },
            {
                "Name": "dev:device_sdk",
                "Value": "aws-sdk"
            }
        ],
        "DeviceCreateDate": 1715100742.022,
        "DeviceLastModifiedDate": 1723233651.167,
        "DeviceLastAuthenticatedDate": 1715100742.0
    }
}
```

자세한 내용은 Amazon Cognito 개발자 안내서의 사용자 풀에서 사용자 디바이스 작업 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AdminGetDevice 섹션을 참조하세요.

admin-get-user

다음 코드 예시에서는 admin-get-user 코드를 사용하는 방법을 보여줍니다.

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
}
```

- API 세부 정보는 AWS CLI 명령 참조에서 AdminGetUser 섹션을 참조하세요.

`admin-initiate-auth`

다음 코드 예시는 `admin-initiate-auth`의 사용 방법을 보여 줍니다.

AWS CLI

관리 사용자로 로그인하려면

다음 `admin-initiate-auth` 예제에서는 사용자 diego@example.com에 로그인합니다. 이 예제에는 위협 방지를 위한 메타데이터와 Lambda 트리거를 위한 ClientMetadata도 포함되어 있습니다. 사용자는 TOTP MFA에 대해 구성되어 있으며 인증을 완료하기 전에 인증 앱에서 코드를 제공하라는 질문을 받습니다.

```
aws cognito-idp admin-initiate-auth \
    --user-pool-id us-west-2_EXAMPLE \
    --client-id 1example23456789 \
    --auth-flow ADMIN_USER_PASSWORD_AUTH \
    --auth-parameters USERNAME=diego@example.com,PASSWORD="My@Example
$Password3!",SECRET_HASH=ExampleEncodedClientIdSecretAndUsername= \
    --context-data="{\"EncodedData\":\"abc123example\",\"HttpHeaders\":
[{\"headerName\":\"UserAgent\",\"headerValue\":\"Mozilla/5.0 (Windows NT 6.1;
 Win64; x64; rv:47.0) Gecko/20100101 Firefox/47.0\"}],\"IpAddress\":\"192.0.2.1\",
\"ServerName\":\"example.com\",\"ServerPath\":\"/login\"}" \
    --client-metadata="{\"MyExampleKey\": \"MyExampleValue\"}"
```

출력:

```
{
    "ChallengeName": "SOFTWARE_TOKEN_MFA",
    "Session": "AYABeExample...",
    "ChallengeParameters": {
        "FRIENDLY_DEVICE_NAME": "MyAuthenticatorApp",
        "USER_ID_FOR_SRP": "diego@example.com"
    }
}
```

자세한 내용은 Amazon Cognito 개발자 안내서의 관리자 인증 흐름 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조에서 AdminInitiateAuth 섹션을 참조하세요.

admin-link-provider-for-user

다음 코드 예시는 `admin-link-provider-for-user`의 사용 방법을 보여 줍니다.

AWS CLI

로컬 사용자를 페더레이션 사용자에 연결하려면

다음 `admin-link-provider-for-user` 예제에서는 로컬 사용자 diego를 Google을 사용하여 페더레이션 로그인을 수행할 사용자에게 연결합니다.

```
aws cognito-idp admin-link-provider-for-user \
    --user-pool-id us-west-2_EXAMPLE \
    --destination-user ProviderName=Cognito,ProviderAttributeValue=diego \
```

```
    --source-
user ProviderAttributeName=Cognito_Subject,ProviderAttributeValue=0000000000000000,ProviderN
```

자세한 내용은 Amazon Cognito 개발자 안내서의 페더레이션 사용자를 기존 사용자 프로파일에 연결을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AdminLinkProviderForUser 섹션을 참조하세요.

admin-list-devices

다음 코드 예시는 `admin-list-devices`의 사용 방법을 보여 줍니다.

AWS CLI

사용자의 디바이스를 나열하는 방법

다음 `admin-list-devices` 예제에서는 사용자 diego의 디바이스를 나열합니다.

```
aws cognito-idp admin-list-devices \
    --user-pool-id us-west-2_EXAMPLE \
    --username diego \
    --limit 1
```

출력:

```
{
    "Devices": [
        {
            "DeviceKey": "us-west-2_a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "DeviceAttributes": [
                {
                    "Name": "device_status",
                    "Value": "valid"
                },
                {
                    "Name": "device_name",
                    "Value": "MyDevice"
                },
                {
                    "Name": "dev:device_arn",
                    "Value": "arn:aws:cognito-idp:us-west-2:123456789012:owner/
diego.us-west-2_EXAMPLE/device/us-west-2_a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
                },
```

```
                {
                    "Name": "dev:device_owner",
                    "Value": "diego.us-west-2_EXAMPLE"
                },
                {
                    "Name": "last_ip_used",
                    "Value": "192.0.2.1"
                },
                {
                    "Name": "dev:device_remembered_status",
                    "Value": "remembered"
                },
                {
                    "Name": "dev:device_sdk",
                    "Value": "aws-sdk"
                }
            ],
            "DeviceCreateDate": 1715100742.022,
            "DeviceLastModifiedDate": 1723233651.167,
            "DeviceLastAuthenticatedDate": 1715100742.0
        }
    ]
}
```

자세한 내용은 Amazon Cognito 개발자 안내서의 사용자 풀에서 사용자 디바이스 작업 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AdminListDevices 섹션을 참조하세요.

admin-list-groups-for-user

다음 코드 예시에서는 admin-list-groups-for-user 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용자가 속한 그룹 나열

이 예시에서는 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
      }
  ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 AdminListGroupsForUser 섹션을 참조하세요.

admin-list-user-auth-events

다음 코드 예시에서는 admin-list-user-auth-events 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용자에 대한 권한 부여 이벤트를 나열하는 방법

다음 admin-list-user-auth-events 예제에서는 사용자 diego에 대한 최신 사용자 활동 로그 이벤트를 나열합니다.

```
aws cognito-idp admin-list-user-auth-events \
    --user-pool-id us-west-2_ywDJHlIfU \
    --username brcotter+050123 \
    --max-results 1
```

출력:

```
{
    "AuthEvents": [
        {
            "EventId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "EventType": "SignIn",
```

```
            "CreationDate": 1726694203.495,
            "EventResponse": "InProgress",
            "EventRisk": {
                "RiskDecision": "AccountTakeover",
                "RiskLevel": "Medium",
                "CompromisedCredentialsDetected": false
            },
            "ChallengeResponses": [
                {
                    "ChallengeName": "Password",
                    "ChallengeResponse": "Success"
                }
            ],
            "EventContextData": {
                "IpAddress": "192.0.2.1",
                "City": "Seattle",
                "Country": "United States"
            }
        }
    ],
    "NextToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222#2024-09-18T21:16:43.495Z"
}
```

자세한 내용은 Amazon Cognito 개발자 안내서의 사용자 이벤트 기록 보기 및 내보내기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AdminListUserAuthEvents 섹션을 참조하세요.

`admin-remove-user-from-group`

다음 코드 예시에서는 `admin-remove-user-from-group` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

그룹에서 사용자 제거

이 예시에서는 SampleGroup에서 jane@example.com을 제거합니다.

명령:

```
aws cognito-idp admin-remove-user-from-group --user-pool-id us-west-2_aaaaaaaaa --
username jane@example.com --group-name SampleGroup
```

- API 세부 정보는 AWS CLI 명령 참조의 AdminRemoveUserFromGroup 섹션을 참조하세요.

admin-reset-user-password

다음 코드 예시에서는 admin-reset-user-password 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용자 암호 재설정

이 예시에서는 diego@example.com의 암호를 재설정합니다.

명령:

```
aws cognito-idp admin-reset-user-password --user-pool-id us-west-2_aaaaaaaaa --
username diego@example.com
```

- API 세부 정보는 AWS CLI 명령 참조의 AdminResetUserPassword 섹션을 참조하세요.

admin-respond-to-auth-challenge

다음 코드 예시는 admin-respond-to-auth-challenge의 사용 방법을 보여 줍니다.

AWS CLI

인증 질문에 응답하려면

인증 흐름, 사용자 풀 구성 및 사용자 설정에 따라 다양한 인증 질문에 대응할 수 있는 다양한 방법이 있습니다. 다음 admin-respond-to-auth-challenge 예제에서는 diego@example.com에 대한 TOTP MFA 코드를 제공하고 로그인을 완료합니다. 이 사용자 풀에는 디바이스 기억 기능이 켜져 있으므로 인증 결과도 새 디바이스 키를 반환합니다.

```
aws cognito-idp admin-respond-to-auth-challenge \
    --user-pool-id us-west-2_EXAMPLE \
    --client-id 1example23456789 \
    --challenge-name SOFTWARE_TOKEN_MFA \
    --challenge-responses USERNAME=diego@example.com,SOFTWARE_TOKEN_MFA_CODE=000000
 \
    --session AYABeExample...
```

출력:

```
{
    "ChallengeParameters": {},
    "AuthenticationResult": {
        "AccessToken": "eyJra456defEXAMPLE",
        "ExpiresIn": 3600,
        "TokenType": "Bearer",
        "RefreshToken": "eyJra123abcEXAMPLE",
        "IdToken": "eyJra789ghiEXAMPLE",
        "NewDeviceMetadata": {
            "DeviceKey": "us-west-2_a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "DeviceGroupKey": "-ExAmPlE1"
        }
    }
}
```

자세한 내용은 Amazon Cognito 개발자 안내서의 관리자 인증 흐름 섹션을 참조하세요.

- API 세부 정보는 AWS CLI API 참조의 AdminRespondToAuthChallenge를 참조하세요.

admin-set-user-mfa-preference

다음 코드 예시는 admin-set-user-mfa-preference의 사용 방법을 보여 줍니다.

AWS CLI

사용자 MFA 기본 설정을 지정하는 방법

이 예시에서는 사용자 이름 diego@example.com에 대한 SMS MFA 기본 설정을 설정합니다.

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
```

- API 세부 정보는 AWS CLI 명령 참조의 AdminSetUserMfaPreference 섹션을 참조하세요.

admin-set-user-password

다음 코드 예시는 admin-set-user-password의 사용 방법을 보여 줍니다.

AWS CLI

사용자 암호를 관리자로 설정하려면

다음 `admin-set-user-password` 예제에서는 diego@example.com의 암호를 영구적으로 설정합니다.

```
aws cognito-idp admin-set-user-password \
    --user-pool-id us-west-2_EXAMPLE \
    --username diego@example.com \
    --password MyExamplePassword1! \
    --permanent
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Cognito 개발자 안내서의 암호, 암호 복구 및 암호 정책을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AdminSetUserPassword를 참조하세요.

admin-set-user-settings

다음 코드 예시는 `admin-set-user-settings`의 사용 방법을 보여 줍니다.

AWS CLI

사용자 설정 구성

이 예시에서는 사용자 이름 diego@example.com에 대한 MFA 전송 기본 설정을 EMAIL로 설정합니다.

명령:

```
aws cognito-idp admin-set-user-settings --user-pool-id us-west-2_aaaaaaaaa --
username diego@example.com --mfa-options DeliveryMedium=EMAIL
```

- API 세부 정보는 AWS CLI 명령 참조의 AdminSetUserSettings 섹션을 참조하세요.

admin-update-auth-event-feedback

다음 코드 예시에서는 `admin-update-auth-event-feedback` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

권한 부여 이벤트에 대한 피드백을 제공하는 방법

이 예시에서는 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
```

- API 세부 정보는 AWS CLI 명령 참조의 AdminUpdateAuthEventFeedback 섹션을 참조하세요.

admin-update-device-status

다음 코드 예시에서는 admin-update-device-status 코드를 사용하는 방법을 보여줍니다.

AWS CLI

디바이스 상태 업데이트

이 예시에서는 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
```

- API 세부 정보는 AWS CLI 명령 참조의 AdminUpdateDeviceStatus 섹션을 참조하세요.

admin-update-user-attributes

다음 코드 예시에서는 admin-update-user-attributes 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용자 속성을 업데이트하는 방법

이 예시에서는 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"
```

- API 세부 정보는 AWS CLI 명령 참조의 AdminUpdateUserAttributes 섹션을 참조하세요.

admin-user-global-sign-out

다음 코드 예시는 admin-user-global-sign-out의 사용 방법을 보여 줍니다.

AWS CLI

사용자를 관리자로 로그아웃하려면

다음 admin-user-global-sign-out 예제에서는 사용자 diego@example.com을 로그아웃합니다.

```
aws cognito-idp admin-user-global-sign-out \
    --user-pool-id us-west-2_EXAMPLE \
    --username diego@example.com
```

자세한 내용은 Amazon Cognito 개발자 안내서의 사용자 풀을 통한 인증 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AdminUserGlobalSignOut을 참조하세요.

associate-software-token

다음 코드 예시는 associate-software-token의 사용 방법을 보여 줍니다.

AWS CLI

MFA 인증 앱의 보안 키를 생성하려면

다음 associate-software-token 예제에서는 로그인하고 액세스 토큰을 받은 사용자를 위해 TOTP 프라이빗 키를 생성합니다. 생성된 프라이빗 키는 인증 앱에 수동으로 입력하거나 애플리케이션이 사용자가 스캔할 수 있는 QR 코드로 렌더링할 수 있습니다.

```
aws cognito-idp associate-software-token \
    --access-token eyJra456defEXAMPLE
```

출력:

```
{
    "SecretCode": "QWERTYUIOP123456EXAMPLE"
}
```

자세한 내용은 Amazon Cognito 개발자 안내서의 TOTP 소프트웨어 토큰 MFA를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AssociateSoftwareToken을 참조하세요.

change-password

다음 코드 예시는 change-password의 사용 방법을 보여 줍니다.

AWS CLI

암호 변경

이 예시에서는 암호를 변경합니다.

명령:

```
aws cognito-idp change-password --previous-password OldPassword --proposed-
password NewPassword --access-token ACCESS_TOKEN
```

- API 세부 정보는 AWS CLI 명령 참조의 ChangePassword 섹션을 참조하세요.

confirm-device

다음 코드 예시는 confirm-device의 사용 방법을 보여 줍니다.

AWS CLI

사용자 디바이스를 확인하려면

다음 confirm-device 예제에서는 현재 사용자에 대해 기억된 새 디바이스를 추가합니다.

```
aws cognito-idp confirm-device \
```

```
    --access-token eyJra456defEXAMPLE \
    --device-key us-west-2_a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --device-secret-verifier-
config PasswordVerifier=TXlWZXJpZmllclN0cmluZw,Salt=TXlTUlBTYWx0
```

출력:

```
{
     "UserConfirmationNecessary": false
}
```

자세한 내용은 Amazon Cognito 개발자 안내서의 [사용자 풀에서 사용자 디바이스 작업] 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조에서 [ConfirmDevice]를 참조하세요.

confirm-forgot-password

다음 코드 예시는 confirm-forgot-password의 사용 방법을 보여 줍니다.

AWS CLI

암호를 잊어버렸는지 확인하는 방법

이 예시에서는 사용자 이름 diego@example.com의 잊어버린 암호를 확인합니다.

명령:

```
aws cognito-idp confirm-forgot-password --client-id 3n4b5urk1ft4fl3mg5e62d9ado --
username=diego@example.com --password PASSWORD --confirmation-code CONF_CODE
```

- API 세부 정보는 AWS CLI 명령 참조의 [ConfirmForgotPassword] 섹션을 참조하세요.

confirm-sign-up

다음 코드 예시에서는 confirm-sign-up 코드를 사용하는 방법을 보여줍니다.

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
```

- API 세부 정보는 AWS CLI 명령 참조에서 ConfirmSignUp을 참조하세요.

create-group

다음 코드 예시에서는 create-group 코드를 사용하는 방법을 보여줍니다.

AWS CLI

그룹 생성

이 예시에서는 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 생성하는 방법

이 예시에서는 설명이 포함된 그룹을 생성합니다. 또한 역할과 우선 순위도 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
  }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateGroup을 참조하세요.

`create-identity-provider`

다음 코드 예시는 `create-identity-provider`의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 메타데이터 URL을 사용하여 사용자 풀 SAML ID 제공업체(idP) 생성

다음 `create-identity-provider` 예제에서는 퍼블릭 URL의 메타데이터, 속성 매핑 및 두 식별자를 사용하여 새 SAML IdP를 생성합니다.

```
aws cognito-idp create-identity-provider \
    --user-pool-id us-west-2_EXAMPLE \
    --provider-name MySAML \
    --provider-type SAML \
    --provider-
details IDPInit=true,IDPSignout=true,EncryptedResponses=true,MetadataURL=https://
auth.example.com/sso/saml/metadata,RequestSigningAlgorithm=rsa-sha256 \
    --attribute-mapping email=emailaddress,phone_number=phone,custom:111=department
 \
    --idp-identifiers CorpSAML WestSAML
```

출력:

```
{
    "IdentityProvider": {
        "UserPoolId": "us-west-2_EXAMPLE",
        "ProviderName": "MySAML",
        "ProviderType": "SAML",
        "ProviderDetails": {
            "ActiveEncryptionCertificate": "MIICvTCCAaEXAMPLE",
            "EncryptedResponses": "true",
            "IDPInit": "true",
            "IDPSignout": "true",
            "MetadataURL": "https://auth.example.com/sso/saml/metadata",
            "RequestSigningAlgorithm": "rsa-sha256",
            "SLORedirectBindingURI": "https://auth.example.com/slo/saml",
            "SSORedirectBindingURI": "https://auth.example.com/sso/saml"
        },
        "AttributeMapping": {
            "custom:111": "department",
            "emailaddress": "email",
            "phone": "phone_number"
        },
        "IdpIdentifiers": [
            "CorpSAML",
            "WestSAML"
        ],
        "LastModifiedDate": 1726853833.977,
        "CreationDate": 1726853833.977
    }
}
```

자세한 내용은 Amazon Cognito 개발자 안내서의 [제3자를 통해 사용자 풀 로그인 추가](#) 섹션을 참조하세요.

예제 2: 메타데이터 파일을 사용하여 사용자 풀 SAML ID 제공업체(idP) 생성

다음 `create-identity-provider` 예제에서는 파일, 속성 매핑 및 두 식별자의 메타데이터를 사용하여 새 SAML IdP를 생성합니다. 파일 구문은 `--provider-details` 파라미터의 운영 체제마다 다를 수 있습니다. 이 작업에 대한 JSON 입력 파일을 생성하는 것이 가장 쉽습니다.

```
aws cognito-idp create-identity-provider \
    --cli-input-json file://.\SAML-identity-provider.json
```

SAML-identity-provider.json의 콘텐츠:

```
{
    "AttributeMapping": {
        "email" : "idp_email",
        "email_verified" : "idp_email_verified"
    },
    "IdpIdentifiers": [ "platform" ],
    "ProviderDetails": {
        "MetadataFile": "<md:EntityDescriptor xmlns:md=
\"urn:oasis:names:tc:SAML:2.0:metadata\" entityID=\"http://www.example.com/
sso\"><md:IDPSSODescriptor WantAuthnRequestsSigned=\"false\"
 protocolSupportEnumeration=\"urn:oasis:names:tc:SAML:2.0:protocol
\"><md:KeyDescriptor use=\"signing\"><ds:KeyInfo xmlns:ds=\"http://
www.w3.org/2000/09/xmldsig#
\"><ds:X509Data><ds:X509Certificate>[IDP_CERTIFICATE_DATA]</ds:X509Certificate></
ds:X509Data></ds:KeyInfo></md:KeyDescriptor><md:SingleLogoutService
 Binding=\"urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST\" Location=
\"https://www.example.com/slo/saml\"/><md:SingleLogoutService
 Binding=\"urn:oasis:names:tc:SAML:2.0:bindings:HTTP-
Redirect\" Location=\"https://www.example.com/slo/saml\"/
><md:NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified</
md:NameIDFormat><md:NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-
format:emailAddress</md:NameIDFormat><md:SingleSignOnService
 Binding=\"urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST\" Location=
\"https://www.example.com/sso/saml\"/><md:SingleSignOnService Binding=
\"urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect\" Location=\"https://
www.example.com/sso/saml\"/></md:IDPSSODescriptor></md:EntityDescriptor>",
        "IDPSignout" : "true",
        "RequestSigningAlgorithm" : "rsa-sha256",
        "EncryptedResponses" : "true",
        "IDPInit" : "true"
    },
    "ProviderName": "MySAML2",
    "ProviderType": "SAML",
    "UserPoolId": "us-west-2_EXAMPLE"
}
```

출력:

```
{
    "IdentityProvider": {
        "UserPoolId": "us-west-2_EXAMPLE",
```

```
        "ProviderName": "MySAML2",
        "ProviderType": "SAML",
        "ProviderDetails": {
            "ActiveEncryptionCertificate":
 "[USER_POOL_ENCRYPTION_CERTIFICATE_DATA]",
            "EncryptedResponses": "true",
            "IDPInit": "true",
            "IDPSignout": "true",
            "MetadataFile": "<md:EntityDescriptor xmlns:md=
\"urn:oasis:names:tc:SAML:2.0:metadata\" entityID=\"http://www.example.com/
sso\"><md:IDPSSODescriptor WantAuthnRequestsSigned=\"false\"
 protocolSupportEnumeration=\"urn:oasis:names:tc:SAML:2.0:protocol
\"><md:KeyDescriptor use=\"signing\"><ds:KeyInfo xmlns:ds=\"http://
www.w3.org/2000/09/xmldsig#
\"><ds:X509Data><ds:X509Certificate>[IDP_CERTIFICATE_DATA]</ds:X509Certificate></
ds:X509Data></ds:KeyInfo></md:KeyDescriptor><md:SingleLogoutService
 Binding=\"urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST\" Location=
\"https://www.example.com/slo/saml\"/><md:SingleLogoutService
 Binding=\"urn:oasis:names:tc:SAML:2.0:bindings:HTTP-
Redirect\" Location=\"https://www.example.com/slo/saml\"/
><md:NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified</
md:NameIDFormat><md:NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-
format:emailAddress</md:NameIDFormat><md:SingleSignOnService
 Binding=\"urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST\" Location=
\"https://www.example.com/sso/saml\"/><md:SingleSignOnService Binding=
\"urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect\" Location=\"https://
www.example.com/sso/saml\"/></md:IDPSSODescriptor></md:EntityDescriptor>",
            "RequestSigningAlgorithm": "rsa-sha256",
            "SLORedirectBindingURI": "https://www.example.com/slo/saml",
            "SSORedirectBindingURI": "https://www.example.com/sso/saml"
        },
        "AttributeMapping": {
            "email": "idp_email",
            "email_verified": "idp_email_verified"
        },
        "IdpIdentifiers": [
            "platform"
        ],
        "LastModifiedDate": 1726855290.731,
        "CreationDate": 1726855290.731
    }
}
```

자세한 내용은 Amazon Cognito 개발자 안내서의 제3자를 통해 사용자 풀 로그인 추가 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateIdentityProvider 섹션을 참조하세요.

create-resource-server

다음 코드 예시는 `create-resource-server`의 사용 방법을 보여 줍니다.

AWS CLI

사용자 풀 클라이언트 생성

다음 `create-resource-server` 예제에서는 사용자 지정 범위를 사용하여 새 리소스 서버를 생성합니다.

```
aws cognito-idp create-resource-server \
    --user-pool-id us-west-2_EXAMPLE \
    --identifier solar-system-data \
    --name "Solar system object tracker" \
    --scopes ScopeName=sunproximity.read,ScopeDescription="Distance in AU from Sol"
 ScopeName=asteroids.add,ScopeDescription="Enter a new asteroid"
```

출력:

```
{
    "ResourceServer": {
        "UserPoolId": "us-west-2_EXAMPLE",
        "Identifier": "solar-system-data",
        "Name": "Solar system object tracker",
        "Scopes": [
            {
                "ScopeName": "sunproximity.read",
                "ScopeDescription": "Distance in AU from Sol"
            },
            {
                "ScopeName": "asteroids.add",
                "ScopeDescription": "Enter a new asteroid"
            }
        ]
    }
}
```

자세한 내용은 Amazon Cognito 개발자 안내서의 리소스 서버가 있는 범위, M2M 및 API를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateResourceServer를 참조하세요.

create-user-import-job

다음 코드 예시는 `create-user-import-job`의 사용 방법을 보여 줍니다.

AWS CLI

사용자 가져오기 작업 생성

이 예시에서는 MyImportJob이라는 사용자 가져오기 작업을 만듭니다.

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
  }
}
```

사전 서명된 URL을 사용하여 curl을 사용하여 .csv 파일을 업로드합니다.

명령:

```
curl -v -T "PATH_TO_CSV_FILE" -H "x-amz-server-side-encryption:aws:kms"
 "PRE_SIGNED_URL"
```

• API 세부 정보는 AWS CLI 명령 참조의 CreateUserImportJob 섹션을 참조하세요.

create-user-pool-client

다음 코드 예시에서는 create-user-pool-client 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용자 풀 클라이언트 생성

다음 create-user-pool-client 예제에서는 클라이언트 보안 암호, 명시적 읽기 및 쓰기 속성, 사용자 이름 암호 및 SRP 흐름으로 로그인, IdP 3개로 로그인, OAuth 범위의 하위 집합에 대한 액세스, PinPoint 분석 및 확장된 인증 세션 유효성이 있는 새 사용자 풀 클라이언트를 생성합니다.

```
aws cognito-idp create-user-pool-client \
    --user-pool-id us-west-2_EXAMPLE \
    --client-name MyTestClient \
    --generate-secret \
    --refresh-token-validity 10 \
    --access-token-validity 60 \
    --id-token-validity 60 \
    --token-validity-units AccessToken=minutes,IdToken=minutes,RefreshToken=days \
    --read-attributes email phone_number email_verified phone_number_verified \
    --write-attributes email phone_number \
    --explicit-auth-
flows ALLOW_USER_PASSWORD_AUTH ALLOW_USER_SRP_AUTH ALLOW_REFRESH_TOKEN_AUTH \
    --supported-identity-providers Google Facebook MyOIDC \
    --callback-urls https://www.amazon.com https://example.com http://
localhost:8001 myapp://example \
    --allowed-o-auth-flows code implicit \
    --allowed-o-auth-scopes openid profile aws.cognito.signin.user.admin solar-
system-data/asteroids.add \
    --allowed-o-auth-flows-user-pool-client \
    --analytics-configuration ApplicationArn=arn:aws:mobiletargeting:us-
west-2:767671399759:apps/thisisanexamplepinpointapplicationid,UserDataShared=TRUE \
    --prevent-user-existence-errors ENABLED \
    --enable-token-revocation \
```

```
    --enable-propagate-additional-user-context-data \
    --auth-session-validity 4
```

출력:


```
{
    "UserPoolClient": {
        "UserPoolId": "us-west-2_EXAMPLE",
        "ClientName": "MyTestClient",
        "ClientId": "123abc456defEXAMPLE",
        "ClientSecret": "this1234is5678my91011example1213client1415secret",
        "LastModifiedDate": 1726788459.464,
        "CreationDate": 1726788459.464,
        "RefreshTokenValidity": 10,
        "AccessTokenValidity": 60,
        "IdTokenValidity": 60,
        "TokenValidityUnits": {
            "AccessToken": "minutes",
            "IdToken": "minutes",
            "RefreshToken": "days"
        },
        "ReadAttributes": [
            "email_verified",
            "phone_number_verified",
            "phone_number",
            "email"
        ],
        "WriteAttributes": [
            "phone_number",
            "email"
        ],
        "ExplicitAuthFlows": [
            "ALLOW_USER_PASSWORD_AUTH",
            "ALLOW_USER_SRP_AUTH",
            "ALLOW_REFRESH_TOKEN_AUTH"
        ],
        "SupportedIdentityProviders": [
            "Google",
            "MyOIDC",
            "Facebook"
        ],
        "CallbackURLs": [
            "https://example.com",
```

```
            "https://www.amazon.com",
            "myapp://example",
            "http://localhost:8001"
        ],
        "AllowedOAuthFlows": [
            "implicit",
            "code"
        ],
        "AllowedOAuthScopes": [
            "aws.cognito.signin.user.admin",
            "openid",
            "profile",
            "solar-system-data/asteroids.add"
        ],
        "AllowedOAuthFlowsUserPoolClient": true,
        "AnalyticsConfiguration": {
            "ApplicationArn": "arn:aws:mobiletargeting:us-west-2:123456789012:apps/
thisisanexamplepinpointapplicationid",
            "RoleArn": "arn:aws:iam::123456789012:role/aws-service-role/cognito-
idp.amazonaws.com/AWSServiceRoleForAmazonCognitoIdp",
            "UserDataShared": true
        },
        "PreventUserExistenceErrors": "ENABLED",
        "EnableTokenRevocation": true,
        "EnablePropagateAdditionalUserContextData": true,
        "AuthSessionValidity": 4
    }
}
```

자세한 내용은 Amazon Cognito 개발자 안내서의 앱 클라이언트를 사용한 애플리케이션별 설정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateUserPoolClient 섹션을 참조하세요.

`create-user-pool-domain`

다음 코드 예시는 `create-user-pool-domain`의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 사용자 풀 도메인 생성

다음 `create-user-pool-domain` 예제에서는 새로운 사용자 지정 도메인을 생성합니다.

```
aws cognito-idp create-user-pool-domain \
    --user-pool-id us-west-2_EXAMPLE \
    --domain auth.example.com \
    --custom-domain-config CertificateArn=arn:aws:acm:us-
east-1:123456789012:certificate/a1b2c3d4-5678-90ab-cdef-EXAMPLE22222
```

출력:

```
{
    "CloudFrontDomain": "example1domain.cloudfront.net"
}
```

자세한 내용은 Amazon Cognito 개발자 안내서의 사용자 풀 도메인 구성을 참조하세요.

예제 2: 사용자 풀 도메인 생성

다음 `create-user-pool-domain` 예제에서는 서비스 소유 접두사가 있는 새 도메인을 생성합니다.

```
aws cognito-idp create-user-pool-domain \
    --user-pool-id us-west-2_EXAMPLE2 \
    --domain mydomainprefix
```

자세한 내용은 Amazon Cognito 개발자 안내서의 사용자 풀 도메인 구성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateUserPoolDomain 섹션을 참조하세요.

`create-user-pool`

다음 코드 예시에서는 `create-user-pool` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

최소 구성 사용자 풀 생성

이 예시에서는 기본값을 사용하여 MyUserPool이라는 사용자 풀을 만듭니다. 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 예시에서는 사용자 풀 MyUserPool을 생성합니다. 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": {}
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateUserPool을 참조하세요.

delete-group

다음 코드 예시에서는 delete-group 코드를 사용하는 방법을 보여줍니다.

AWS CLI

그룹 삭제

이 예시에서는 그룹을 삭제합니다.

명령:

```
aws cognito-idp delete-group --user-pool-id us-west-2_aaaaaaaaa  --group-name MyGroupName
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteGroup을 참조하세요.

delete-identity-provider

다음 코드 예시에서는 delete-identity-provider 코드를 사용하는 방법을 보여줍니다.

AWS CLI

자격 증명 공급자 삭제

이 예시에서는 자격 증명 공급자를 삭제합니다.

명령:

```
aws cognito-idp delete-identity-provider --user-pool-id us-west-2_aaaaaaaaa --
provider-name Facebook
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteIdentityProvider 섹션을 참조하세요.

delete-resource-server

다음 코드 예시에서는 delete-resource-server 코드를 사용하는 방법을 보여줍니다.

AWS CLI

리소스 서버 삭제

이 예시에서는 weather.example.com 리소스 서버를 삭제합니다.

명령:

```
aws cognito-idp delete-resource-server --user-pool-id us-west-2_aaaaaaaaa --
identifier weather.example.com
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteResourceServer 섹션을 참조하세요.

delete-user-attributes

다음 코드 예시는 delete-user-attributes의 사용 방법을 보여 줍니다.

AWS CLI

사용자 속성을 삭제하는 방법

다음 delete-user-attributes 예제에서는 현재 로그인한 사용자로부터 사용자 지정 속성 "custom:attribute"를 삭제합니다.

```
aws cognito-idp delete-user-attributes \
```

```
    --access-token ACCESS_TOKEN \
    --user-attribute-names "custom:department"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Cognito 개발자 안내서의 사용자 속성 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteUserAttributes 섹션을 참조하세요.

delete-user-pool-client

다음 코드 예시에서는 `delete-user-pool-client` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용자 풀 클라이언트 삭제

이 예시에서는 사용자 풀 클라이언트를 삭제합니다.

명령:

```
aws cognito-idp delete-user-pool-client --user-pool-id us-west-2_aaaaaaaaa --client-id 38fjsnc484p94kpqsnet7mpld0
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteUserPoolClient 섹션을 참조하세요.

delete-user-pool-domain

다음 코드 예시에서는 `delete-user-pool-domain` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용자 풀 도메인 삭제

다음 `delete-user-pool-domain` 예시에서는 `my-domain` 사용자 풀 도메인을 삭제합니다.

```
aws cognito-idp delete-user-pool-domain \
    --user-pool-id us-west-2_aaaaaaaaa \
    --domain my-domain
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteUserPoolDomain 섹션을 참조하세요.

delete-user-pool

다음 코드 예시에서는 delete-user-pool 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용자 풀 삭제

이 예시에서는 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
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteUserPool 섹션을 참조하세요.

delete-user

다음 코드 예시에서는 delete-user 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용자 삭제

이 예제에서는 사용자를 삭제합니다.

명령:

```
aws cognito-idp delete-user --access-token ACCESS_TOKEN
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteUser를 참조하세요.

describe-identity-provider

다음 코드 예시에서는 describe-identity-provider 코드를 사용하는 방법을 보여줍니다.

AWS CLI

자격 증명 공급자를 설명하는 방법

이 예시에서는 Facebook이라는 ID 제공업체를 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
  }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeIdentityProvider 섹션을 참조하세요.

describe-resource-server

다음 코드 예시에서는 `describe-resource-server` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

리소스 서버 설명

이 예시에서는 리소스 서버 weather.example.com을 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"
          }
      ]
  }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeResourceServer 섹션을 참조하세요.

describe-risk-configuration

다음 코드 예시에서는 `describe-risk-configuration` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

리스크 구성 설명

이 예시에서는 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"
             }
         }
     }
  }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeRiskConfiguration 섹션을 참조하세요.

describe-user-import-job

다음 코드 예시에서는 describe-user-import-job 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용자 가져오기 작업 설명

이 예시에서는 사용자 입력 작업을 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
  }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeUserImportJob 섹션을 참조하세요.

describe-user-pool-client

다음 코드 예시에서는 describe-user-pool-client 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용자 풀 클라이언트를 설명하는 방법

이 예시에서는 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
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeUserPoolClient 섹션을 참조하세요.

describe-user-pool-domain

다음 코드 예시에서는 `describe-user-pool-domain` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용자 풀 클라이언트를 설명하는 방법

이 예시에서는 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": {}
  }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeUserPoolDomain 섹션을 참조하세요.

`describe-user-pool`

다음 코드 예시에서는 `describe-user-pool` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용자 풀을 설명하는 방법

다음 예제에서는 사용자 풀 ID us-west-2_EXAMPLE이 있는 사용자 풀을 설명합니다.

```
aws cognito-idp describe-user-pool \
    --user-pool-id us-west-2_EXAMPLE
```

출력:

```
{
    "UserPool": {
        "Id": "us-west-2_EXAMPLE",
        "Name": "MyUserPool",
        "Policies": {
            "PasswordPolicy": {
                "MinimumLength": 8,
                "RequireUppercase": true,
```

```
                "RequireLowercase": true,
                "RequireNumbers": true,
                "RequireSymbols": true,
                "TemporaryPasswordValidityDays": 1
            }
        },
        "DeletionProtection": "ACTIVE",
        "LambdaConfig": {
            "PreSignUp": "arn:aws:lambda:us-
west-2:123456789012:function:MyPreSignUpFunction",
            "CustomMessage": "arn:aws:lambda:us-
west-2:123456789012:function:MyCustomMessageFunction",
            "PostConfirmation": "arn:aws:lambda:us-
west-2:123456789012:function:MyPostConfirmationFunction",
            "PreAuthentication": "arn:aws:lambda:us-
west-2:123456789012:function:MyPreAuthenticationFunction",
            "PostAuthentication": "arn:aws:lambda:us-
west-2:123456789012:function:MyPostAuthenticationFunction",
            "DefineAuthChallenge": "arn:aws:lambda:us-
west-2:123456789012:function:MyDefineAuthChallengeFunction",
            "CreateAuthChallenge": "arn:aws:lambda:us-
west-2:123456789012:function:MyCreateAuthChallengeFunction",
            "VerifyAuthChallengeResponse": "arn:aws:lambda:us-
west-2:123456789012:function:MyVerifyAuthChallengeFunction",
            "PreTokenGeneration": "arn:aws:lambda:us-
west-2:123456789012:function:MyPreTokenGenerationFunction",
            "UserMigration": "arn:aws:lambda:us-
west-2:123456789012:function:MyMigrateUserFunction",
            "PreTokenGenerationConfig": {
                "LambdaVersion": "V2_0",
                "LambdaArn": "arn:aws:lambda:us-
west-2:123456789012:function:MyPreTokenGenerationFunction"
            },
            "CustomSMSSender": {
                "LambdaVersion": "V1_0",
                "LambdaArn": "arn:aws:lambda:us-
west-2:123456789012:function:MyCustomSMSSenderFunction"
            },
            "CustomEmailSender": {
                "LambdaVersion": "V1_0",
                "LambdaArn": "arn:aws:lambda:us-
west-2:123456789012:function:MyCustomEmailSenderFunction"
            },
```

```
            "KMSKeyID": "arn:aws:kms:us-west-2:123456789012:key/a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"
        },
        "LastModifiedDate": 1726784814.598,
        "CreationDate": 1602103465.273,
        "SchemaAttributes": [
            {
                "Name": "sub",
                "AttributeDataType": "String",
                "DeveloperOnlyAttribute": false,
                "Mutable": false,
                "Required": true,
                "StringAttributeConstraints": {
                    "MinLength": "1",
                    "MaxLength": "2048"
                }
            },
            {
                "Name": "name",
                "AttributeDataType": "String",
                "DeveloperOnlyAttribute": false,
                "Mutable": true,
                "Required": false,
                "StringAttributeConstraints": {
                    "MinLength": "0",
                    "MaxLength": "2048"
                }
            },
            {
                "Name": "given_name",
                "AttributeDataType": "String",
                "DeveloperOnlyAttribute": false,
                "Mutable": true,
                "Required": false,
                "StringAttributeConstraints": {
                    "MinLength": "0",
                    "MaxLength": "2048"
                }
            },
            {
                "Name": "family_name",
                "AttributeDataType": "String",
                "DeveloperOnlyAttribute": false,
                "Mutable": true,
```

```
                    "Required": false,
                    "StringAttributeConstraints": {
                        "MinLength": "0",
                        "MaxLength": "2048"
                    }
                },
                {
                    "Name": "middle_name",
                    "AttributeDataType": "String",
                    "DeveloperOnlyAttribute": false,
                    "Mutable": true,
                    "Required": false,
                    "StringAttributeConstraints": {
                        "MinLength": "0",
                        "MaxLength": "2048"
                    }
                },
                {
                    "Name": "nickname",
                    "AttributeDataType": "String",
                    "DeveloperOnlyAttribute": false,
                    "Mutable": true,
                    "Required": false,
                    "StringAttributeConstraints": {
                        "MinLength": "0",
                        "MaxLength": "2048"
                    }
                },
                {
                    "Name": "preferred_username",
                    "AttributeDataType": "String",
                    "DeveloperOnlyAttribute": false,
                    "Mutable": true,
                    "Required": false,
                    "StringAttributeConstraints": {
                        "MinLength": "0",
                        "MaxLength": "2048"
                    }
                },
                {
                    "Name": "profile",
                    "AttributeDataType": "String",
                    "DeveloperOnlyAttribute": false,
                    "Mutable": true,
```

```
                "Required": false,
                "StringAttributeConstraints": {
                    "MinLength": "0",
                    "MaxLength": "2048"
                }
            },
            {
                "Name": "picture",
                "AttributeDataType": "String",
                "DeveloperOnlyAttribute": false,
                "Mutable": true,
                "Required": false,
                "StringAttributeConstraints": {
                    "MinLength": "0",
                    "MaxLength": "2048"
                }
            },
            {
                "Name": "website",
                "AttributeDataType": "String",
                "DeveloperOnlyAttribute": false,
                "Mutable": true,
                "Required": false,
                "StringAttributeConstraints": {
                    "MinLength": "0",
                    "MaxLength": "2048"
                }
            },
            {
                "Name": "email",
                "AttributeDataType": "String",
                "DeveloperOnlyAttribute": false,
                "Mutable": true,
                "Required": true,
                "StringAttributeConstraints": {
                    "MinLength": "0",
                    "MaxLength": "2048"
                }
            },
            {
                "Name": "email_verified",
                "AttributeDataType": "Boolean",
                "DeveloperOnlyAttribute": false,
                "Mutable": true,
```

```
                "Required": false
            },
            {
                "Name": "gender",
                "AttributeDataType": "String",
                "DeveloperOnlyAttribute": false,
                "Mutable": true,
                "Required": false,
                "StringAttributeConstraints": {
                    "MinLength": "0",
                    "MaxLength": "2048"
                }
            },
            {
                "Name": "birthdate",
                "AttributeDataType": "String",
                "DeveloperOnlyAttribute": false,
                "Mutable": true,
                "Required": false,
                "StringAttributeConstraints": {
                    "MinLength": "10",
                    "MaxLength": "10"
                }
            },
            {
                "Name": "zoneinfo",
                "AttributeDataType": "String",
                "DeveloperOnlyAttribute": false,
                "Mutable": true,
                "Required": false,
                "StringAttributeConstraints": {
                    "MinLength": "0",
                    "MaxLength": "2048"
                }
            },
            {
                "Name": "locale",
                "AttributeDataType": "String",
                "DeveloperOnlyAttribute": false,
                "Mutable": true,
                "Required": false,
                "StringAttributeConstraints": {
                    "MinLength": "0",
                    "MaxLength": "2048"
```

```
            }
        },
        {
            "Name": "phone_number",
            "AttributeDataType": "String",
            "DeveloperOnlyAttribute": false,
            "Mutable": true,
            "Required": false,
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            }
        },
        {
            "Name": "phone_number_verified",
            "AttributeDataType": "Boolean",
            "DeveloperOnlyAttribute": false,
            "Mutable": true,
            "Required": false
        },
        {
            "Name": "address",
            "AttributeDataType": "String",
            "DeveloperOnlyAttribute": false,
            "Mutable": true,
            "Required": false,
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            }
        },
        {
            "Name": "updated_at",
            "AttributeDataType": "Number",
            "DeveloperOnlyAttribute": false,
            "Mutable": true,
            "Required": false,
            "NumberAttributeConstraints": {
                "MinValue": "0"
            }
        },
        {
            "Name": "identities",
            "AttributeDataType": "String",
```

```
                "DeveloperOnlyAttribute": false,
                "Mutable": true,
                "Required": false,
                "StringAttributeConstraints": {}
            },
            {
                "Name": "custom:111",
                "AttributeDataType": "String",
                "DeveloperOnlyAttribute": false,
                "Mutable": true,
                "Required": false,
                "StringAttributeConstraints": {
                    "MinLength": "1",
                    "MaxLength": "256"
                }
            },
            {
                "Name": "dev:custom:222",
                "AttributeDataType": "String",
                "DeveloperOnlyAttribute": true,
                "Mutable": true,
                "Required": false,
                "StringAttributeConstraints": {
                    "MinLength": "1",
                    "MaxLength": "421"
                }
            },
            {
                "Name": "custom:accesstoken",
                "AttributeDataType": "String",
                "DeveloperOnlyAttribute": false,
                "Mutable": true,
                "Required": false,
                "StringAttributeConstraints": {
                    "MaxLength": "2048"
                }
            },
            {
                "Name": "custom:idtoken",
                "AttributeDataType": "String",
                "DeveloperOnlyAttribute": false,
                "Mutable": true,
                "Required": false,
                "StringAttributeConstraints": {
```

```
                    "MaxLength": "2048"
                }
            }
        ],
        "AutoVerifiedAttributes": [
            "email"
        ],
        "SmsVerificationMessage": "Your verification code is {####}. ",
        "EmailVerificationMessage": "Your verification code is {####}. ",
        "EmailVerificationSubject": "Your verification code",
        "VerificationMessageTemplate": {
            "SmsMessage": "Your verification code is {####}. ",
            "EmailMessage": "Your verification code is {####}. ",
            "EmailSubject": "Your verification code",
            "EmailMessageByLink": "Please click the link below to verify your email
 address. <b>{##Verify Your Email##}</b>\n this is from us-west-2_ywDJHlIfU",
            "EmailSubjectByLink": "Your verification link",
            "DefaultEmailOption": "CONFIRM_WITH_LINK"
        },
        "SmsAuthenticationMessage": "Your verification code is {####}. ",
        "UserAttributeUpdateSettings": {
            "AttributesRequireVerificationBeforeUpdate": []
        },
        "MfaConfiguration": "OPTIONAL",
        "DeviceConfiguration": {
            "ChallengeRequiredOnNewDevice": true,
            "DeviceOnlyRememberedOnUserPrompt": false
        },
        "EstimatedNumberOfUsers": 166,
        "EmailConfiguration": {
            "SourceArn": "arn:aws:ses:us-west-2:123456789012:identity/
admin@example.com",
            "EmailSendingAccount": "DEVELOPER"
        },
        "SmsConfiguration": {
            "SnsCallerArn": "arn:aws:iam::123456789012:role/service-role/userpool-
SMS-Role",
            "ExternalId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "SnsRegion": "us-west-2"
        },
        "UserPoolTags": {},
        "Domain": "myCustomDomain",
        "CustomDomain": "auth.example.com",
        "AdminCreateUserConfig": {
```

```
            "AllowAdminCreateUserOnly": false,
            "UnusedAccountValidityDays": 1,
            "InviteMessageTemplate": {
                "SMSMessage": "Your username is {username} and temporary password is
 {####}. ",
                "EmailMessage": "Your username is {username} and temporary password
 is {####}. ",
                "EmailSubject": "Your temporary password"
            }
        },
        "UserPoolAddOns": {
            "AdvancedSecurityMode": "ENFORCED",
            "AdvancedSecurityAdditionalFlows": {}
        },
        "Arn": "arn:aws:cognito-idp:us-west-2:123456789012:userpool/us-
west-2_EXAMPLE",
        "AccountRecoverySetting": {
            "RecoveryMechanisms": [
                {
                    "Priority": 1,
                    "Name": "verified_email"
                }
            ]
        }
    }
}
```

자세한 내용은 Amazon Cognito 개발자 안내서의 Amazon Cognito 사용자 풀을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeUserPool 섹션을 참조하세요.

forget-device

다음 코드 예시에서는 `forget-device` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

디바이스 지우기

이 예시에서는 디바이스를 잊어버립니다.

명령:

```
aws cognito-idp forget-device --device-key us-west-2_abcd_1234-5678
```

- API 세부 정보는 AWS CLI 명령 참조의 [ForgetDevice](#) 섹션을 참조하세요.

`forgot-password`

다음 코드 예시에서는 `forgot-password` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

암호를 강제로 변경하는 방법

다음 `forgot-password` 예시에서는 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [ForgotPassword](#) 섹션을 참조하세요.

`get-csv-header`

다음 코드 예시에서는 `get-csv-header` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

csv 헤더를 생성하는 방법

이 예시에서는 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

- API 세부 정보는 AWS CLI 명령 참조의 GetCsvHeader 섹션을 참조하세요.

get-device

다음 코드 예시는 get-device의 사용 방법을 보여 줍니다.

AWS CLI

디바이스를 가져오는 방법

다음 `get-device` 예제에서는 현재 로그인한 사용자를 위한 디바이스 하나를 표시합니다.

```
aws cognito-idp get-device \
    --access-token eyJra456defEXAMPLE \
    --device-key us-west-2_a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

출력:

```
{
    "Device": {
        "DeviceKey": "us-west-2_a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "DeviceAttributes": [
            {
                "Name": "device_status",
                "Value": "valid"
            },
            {
                "Name": "device_name",
                "Value": "MyDevice"
            },
            {
                "Name": "dev:device_arn",
                "Value": "arn:aws:cognito-idp:us-west-2:123456789012:owner/diego.us-
west-2_EXAMPLE/device/us-west-2_a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
            },
            {
                "Name": "dev:device_owner",
                "Value": "diego.us-west-2_EXAMPLE"
            },
            {
                "Name": "last_ip_used",
                "Value": "192.0.2.1"
            },
            {
                "Name": "dev:device_remembered_status",
                "Value": "remembered"
            },
            {
                "Name": "dev:device_sdk",
```

```
                "Value": "aws-sdk"
            }
        ],
        "DeviceCreateDate": 1715100742.022,
        "DeviceLastModifiedDate": 1723233651.167,
        "DeviceLastAuthenticatedDate": 1715100742.0
    }
}
```

자세한 내용은 Amazon Cognito 개발자 안내서의 사용자 풀에서 사용자 디바이스 작업 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetDevice 섹션을 참조하세요.

get-group

다음 코드 예시는 get-group의 사용 방법을 보여 줍니다.

AWS CLI

그룹에 대한 정보 가져오기

다음 get-group 예제에서는 MyGroup이라는 사용자 그룹의 속성을 나열합니다. 이 그룹에는 우선 순위 및 연결된 IAM 역할이 있습니다.

```
aws cognito-idp get-group \
    --user-pool-id us-west-2_EXAMPLE \
    --group-name MyGroup
```

출력:

```
{
    "Group": {
        "GroupName": "MyGroup",
        "UserPoolId": "us-west-2_EXAMPLE",
        "RoleArn": "arn:aws:iam::123456789012:role/example-cognito-role",
        "Precedence": 7,
        "LastModifiedDate": 1697211218.305,
        "CreationDate": 1611685503.954
    }
}
```

자세한 내용은 Amazon Cognito 개발자 안내서의 사용자 풀에 그룹 추가를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetGroup을 참조하세요.

get-identity-provider-by-identifier

다음 코드 예시는 get-identity-provider-by-identifier의 사용 방법을 보여 줍니다.

AWS CLI

IdP 식별자에서 자격 증명 공급자의 구성 가져오기

다음 get-identity-provider-by-identifier 예제에서는 식별자(mysso)를 사용하여 자격 증명 공급자의 구성을 반환합니다.

```
aws cognito-idp get-identity-provider-by-identifier \
    --user-pool-id us-west-2_EXAMPLE \
    --idp-identifier mysso
```

출력:

```
{
    "IdentityProvider": {
        "UserPoolId": "us-west-2_EXAMPLE",
        "ProviderName": "MYSAML",
        "ProviderType": "SAML",
        "ProviderDetails": {
            "ActiveEncryptionCertificate": "[Certificate contents]",
            "IDPSignout": "false",
            "MetadataURL": "https://auth.example.com/saml/metadata/",
            "SLORedirectBindingURI": "https://auth.example.com/saml/logout/",
            "SSORedirectBindingURI": "https://auth.example.com/saml/assertion/"
        },
        "AttributeMapping": {
            "email": "email"
        },
        "IdpIdentifiers": [
            "mysso",
            "mysamlsso"
        ],
        "LastModifiedDate": 1705616729.188,
        "CreationDate": 1643734622.919
    }
```

```
}
```

자세한 내용은 Amazon Cognito 개발자 안내서의 Third-party IdP sign-in을 참조하시기 바랍니다.

- API 세부 정보는 AWS CLI 명령 참조의 GetIdentityProviderByIdentifier를 참조하시기 바랍니다.

`get-log-delivery-configuration`

다음 코드 예시는 `get-log-delivery-configuration`의 사용 방법을 보여 줍니다.

AWS CLI

로그 전송 구성 표시

다음 `get-log-delivery-configuration` 예제에서는 요청된 사용자 풀의 로그 내보내기 설정을 표시합니다.

```
aws cognito-idp get-log-delivery-configuration \
    --user-pool-id us-west-2_EXAMPLE
```

출력:

```
{
    "LogDeliveryConfiguration": {
        "UserPoolId": "us-west-2_EXAMPLE",
        "LogConfigurations": [
            {
                "LogLevel": "INFO",
                "EventSource": "userAuthEvents",
                "FirehoseConfiguration": {
                    "StreamArn": "arn:aws:firehose:us-
west-2:123456789012:deliverystream/my-test-deliverystream"
                }
            },
            {
                "LogLevel": "ERROR",
                "EventSource": "userNotification",
                "CloudWatchLogsConfiguration": {
                    "LogGroupArn": "arn:aws:logs:us-west-2:123456789012:log-
group:my-message-delivery-logs"
                }
            }
        ]
```

```
    }
}
```

자세한 내용은 Amazon Cognito 개발자 안내서의 Exporting user pool logs를 참조하시기 바랍니다.

- API 세부 정보는 AWS CLI 명령 참조의 GetLogDeliveryConfiguration을 참조하시기 바랍니다.

get-signing-certificate

다음 코드 예시는 get-signing-certificate의 사용 방법을 보여 줍니다.

AWS CLI

SAML 서명 인증서 표시

다음 get-signing-certificate 예제에서는 요청 사용자 풀에 대한 SAML 2.0 서명 인증서를 표시합니다.

```
aws cognito-idp get-signing-certificate \
    --user-pool-id us-west-2_EXAMPLE
```

출력:

```
{
    "Certificate": "[Certificate content]"
}
```

자세한 내용은 Amazon Cognito 개발자 안내서의 SAML signing and encryption을 참조하시기 바랍니다.

- API 세부 정보는 AWS CLI 명령 참조의 GetSigningCertificate 섹션을 참조하세요.

get-ui-customization

다음 코드 예시는 get-ui-customization의 사용 방법을 보여 줍니다.

AWS CLI

앱 클라이언트에 대한 클래식 호스팅 UI 사용자 지정 설정 표시

다음 get-ui-customization 예제에서는 사용자 풀에서 설정을 상속하지 않는 앱 클라이언트에 대한 클래식 호스팅 UI 사용자 지정 설정을 표시합니다.

```
aws cognito-idp get-ui-customization \
    --user-pool-id us-west-2_EXAMPLE \
    --client-id 1example23456789
```

출력:


```
{
    "UICustomization": {
        "UserPoolId": "us-west-2_EXAMPLE",
        "ClientId": "1example23456789",
        "ImageUrl": "https://example.cloudfront.net/us-
west-2_EXAMPLE/1example23456789/20250115191928/assets/images/image.jpg",
        "CSS": "\n.logo-customizable {\n  max-width: 80%;\n  max-height: 30%;
\n}\n\n.banner-customizable {\n  padding: 25px 0px 25px 0px;\n  background-color:
 lightgray;\n}\n\n.label-customizable {\n  font-weight: 400;\n}\n\n.textDescription-
customizable {\n  padding-top: 100px;\n  padding-bottom: 10px;\n  display: block;
\n  font-size: 12px;\n}\n\n.idpDescription-customizable {\n  padding-top: 10px;\n
  padding-bottom: 10px;\n  display: block;\n  font-size: 16px;\n}\n\n.legalText-
customizable {\n  color: #747474;\n  font-size: 11px;\n}\n\n.submitButton-
customizable {\n  font-size: 14px;\n  font-weight: bold;\n  margin: 20px 0px
 10px 0px;\n  height: 50px;\n  width: 100%;\n  color: #fff;\n  background-color:
 #337ab7;\n}\n\n.submitButton-customizable:hover {\n  color: #fff;\n  background-
color: #286090;\n}\n\n.errorMessage-customizable {\n  padding: 5px;\n  font-size:
 12px;\n  width: 100%;\n  background: #F5F5F5;\n  border: 2px solid #D64958;\n
  color: #D64958;\n}\n\n.inputField-customizable {\n  width: 100%;\n  height:
 34px;\n  color: #555;\n  background-color: #fff;\n  border: 1px solid #ccc;\n}\n
\n.inputField-customizable:focus {\n  border-color: #66afe9;\n  outline: 0;\n}\n
\n.idpButton-customizable {\n  height: 40px;\n  width: 100%;\n  width: 100%;\n
 text-align: center;\n  margin-bottom: 15px;\n  color: #fff;\n  background-color:
 #5bc0de;\n  border-color: #46b8da;\n}\n\n.idpButton-customizable:hover {\n
 color: #fff;\n  background-color: #31b0d5;\n}\n\n.socialButton-customizable {\n
  border-radius: 2px;\n  height: 60px;\n  margin-bottom: 15px;\n  padding: 1px;
\n  text-align: left;\n  width: 100%;\n}\n\n.redirect-customizable {\n  text-
align: center;\n}\n\n.passwordCheck-notValid-customizable {\n  color: #DF3312;
\n}\n\n.passwordCheck-valid-customizable {\n  color: #19BF00;\n}\n\n.background-
customizable {\n  background-color: #fff;\n}\n",
        "CSSVersion": "20250115191928"
    }
}
```


자세한 내용은 Amazon Cognito 개발자 안내서의 [Hosted UI (classic) branding](#)을 참조하시기 바랍니다.

- API 세부 정보는 AWS CLI 명령 참조의 GetUiCustomization 섹션을 참조하세요.

get-user-attribute-verification-code

다음 코드 예시는 get-user-attribute-verification-code의 사용 방법을 보여 줍니다.

AWS CLI

현재 사용자에게 속성 확인 코드 전송

다음 get-user-attribute-verification-code 예제에서는 현재 로그인한 사용자의 이메일 주소로 속성 확인 코드를 보냅니다.

```
aws cognito-idp get-user-attribute-verification-code \
    --access-token eyJra456defEXAMPLE \
    --attribute-name email
```

출력:

```
{
    "CodeDeliveryDetails": {
        "Destination": "a***@e***",
        "DeliveryMedium": "EMAIL",
        "AttributeName": "email"
    }
}
```

자세한 내용은 Amazon Cognito 개발자 안내서의 사용자 계정 가입 및 확인 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetUserAttributeVerificationCode를 참조하시기 바랍니다.

get-user-auth-factors

다음 코드 예시는 get-user-auth-factors의 사용 방법을 보여 줍니다.

AWS CLI

현재 사용자가 사용할 수 있는 인증 요소 나열

다음 get-user-auth-factors 예제에서는 현재 로그인한 사용자가 사용 가능한 인증 요소를 나열합니다.

```
aws cognito-idp get-user-auth-factors \
    --access-token eyJra456defEXAMPLE
```

출력:

```
{
    "Username": "testuser",
    "ConfiguredUserAuthFactors": [
        "PASSWORD",
        "EMAIL_OTP",
        "SMS_OTP",
        "WEB_AUTHN"
    ]
}
```

자세한 내용은 Amazon Cognito 개발자 안내서의 Authentication을 참조하시기 바랍니다.

- API 세부 정보는 AWS CLI 명령 참조의 GetUserAuthFactors를 참조하시기 바랍니다.

get-user-pool-mfa-config

다음 코드 예시는 get-user-pool-mfa-config의 사용 방법을 보여 줍니다.

AWS CLI

사용자 풀의 다중 인증 및 WebAuthn 설정 표시

다음 get-user-pool-mfa-config 예제에서는 요청된 사용자 풀의 MFA 및 WebAuthn 구성을 표시합니다.

```
aws cognito-idp get-user-pool-mfa-config \
    --user-pool-id us-west-2_EXAMPLE
```

출력:

```
{
    "SmsMfaConfiguration": {
        "SmsAuthenticationMessage": "Your OTP for MFA or sign-in: use {####}.",
        "SmsConfiguration": {
            "SnsCallerArn": "arn:aws:iam::123456789012:role/service-role/my-SMS-
Role",
            "ExternalId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
```

```
            "SnsRegion": "us-west-2"
        }
    },
    "SoftwareTokenMfaConfiguration": {
        "Enabled": true
    },
    "EmailMfaConfiguration": {
        "Message": "Your OTP for MFA or sign-in: use {####}",
        "Subject": "OTP test"
    },
    "MfaConfiguration": "OPTIONAL",
    "WebAuthnConfiguration": {
        "RelyingPartyId": "auth.example.com",
        "UserVerification": "preferred"
    }
}
```

자세한 내용은 Amazon Cognito 개발자 안내서의 Adding MFA를 참조하시기 바랍니다.

- API 세부 정보는 AWS CLI 명령 참조의 GetUserPoolMfaConfig를 참조하시기 바랍니다.

get-user

다음 코드 예시는 get-user의 사용 방법을 보여 줍니다.

AWS CLI

현재 사용자의 세부 정보 가져오기

다음 get-user 예제에서는 현재 로그인한 사용자의 프로필을 표시합니다.

```
aws cognito-idp get-user \
    --access-token eyJra456defEXAMPLE
```

출력:

```
{
    "Username": "johndoe",
    "UserAttributes": [
        {
            "Name": "sub",
            "Value": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
        },
```

```
        {
            "Name": "identities",
            "Value": "[{\"userId\":\"a1b2c3d4-5678-90ab-cdef-EXAMPLE22222\",
\"providerName\":\"SignInWithApple\",\"providerType\":\"SignInWithApple\",\"issuer
\":null,\"primary\":false,\"dateCreated\":1701125599632}]"
        },
        {
            "Name": "email_verified",
            "Value": "true"
        },
        {
            "Name": "custom:state",
            "Value": "Maine"
        },
        {
            "Name": "name",
            "Value": "John Doe"
        },
        {
            "Name": "phone_number_verified",
            "Value": "true"
        },
        {
            "Name": "phone_number",
            "Value": "+12065551212"
        },
        {
            "Name": "preferred_username",
            "Value": "jamesdoe"
        },
        {
            "Name": "locale",
            "Value": "EMEA"
        },
        {
            "Name": "email",
            "Value": "jamesdoe@example.com"
        }
    ]
}
```

자세한 정보는 Amazon Cognito 개발자 안내서의 사용자 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetUser를 참조하세요.

global-sign-out

다음 코드 예시는 global-sign-out의 사용 방법을 보여 줍니다.

AWS CLI

현재 사용자 로그아웃

다음 global-sign-out 예제에서는 현재 사용자를 로그아웃합니다.

```
aws cognito-idp global-sign-out \
    --access-token eyJra456defEXAMPLE
```

이 명령은 출력을 생성하지 않습니다.

자세한 정보는 Amazon Cognito 개발자 안내서의 사용자 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GlobalSignOut을 참조하시기 바랍니다.

initiate-auth

다음 코드 예시는 initiate-auth의 사용 방법을 보여 줍니다.

AWS CLI

사용자 로그인

다음 initiate-auth 예제에서는 기본 사용자 이름-암호 흐름으로 사용자를 로그인하며 추가 문제는 없습니다.

```
aws cognito-idp initiate-auth \
    --auth-flow USER_PASSWORD_AUTH \
    --client-id 1example23456789 \
    --analytics-metadata AnalyticsEndpointId=d70b2ba36a8c4dc5a04a0451aEXAMPLE \
    --auth-parameters USERNAME=testuser,PASSWORD=[Password] --user-context-
data EncodedData=mycontextdata --client-metadata MyTestKey=MyTestValue
```

출력:

```
{
    "AuthenticationResult": {
        "AccessToken": "eyJra456defEXAMPLE",
        "ExpiresIn": 3600,
```

```
        "TokenType": "Bearer",
        "RefreshToken": "eyJra123abcEXAMPLE",
        "IdToken": "eyJra789ghiEXAMPLE",
        "NewDeviceMetadata": {
            "DeviceKey": "us-west-2_a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "DeviceGroupKey": "-v7w9UcY6"
        }
    }
}
```

자세한 내용은 Amazon Cognito 개발자 안내서의 Authentication을 참조하시기 바랍니다.

- API 세부 정보는 AWS CLI 명령 참조의 InitiateAuth를 참조하시기 바랍니다.

list-devices

다음 코드 예시는 `list-devices`의 사용 방법을 보여 줍니다.

AWS CLI

사용자의 디바이스 나열

다음 `list-devices` 예제에서는 현재 사용자가 등록한 디바이스를 나열합니다.

```
aws cognito-idp list-devices \
    --access-token eyJra456defEXAMPLE
```

출력:

```
{
    "Devices": [
        {
            "DeviceAttributes": [
                {
                    "Name": "device_status",
                    "Value": "valid"
                },
                {
                    "Name": "device_name",
                    "Value": "Dart-device"
                },
                {
                    "Name": "last_ip_used",
```

```
                    "Value": "192.0.2.1"
                }
            ],
            "DeviceCreateDate": 1715100742.022,
            "DeviceKey": "us-west-2_a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "DeviceLastAuthenticatedDate": 1715100742.0,
            "DeviceLastModifiedDate": 1723233651.167
        },
        {
            "DeviceAttributes": [
                {
                    "Name": "device_status",
                    "Value": "valid"
                },
                {
                    "Name": "last_ip_used",
                    "Value": "192.0.2.2"
                }
            ],
            "DeviceCreateDate": 1726856147.993,
            "DeviceKey": "us-west-2_a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "DeviceLastAuthenticatedDate": 1726856147.0,
            "DeviceLastModifiedDate": 1726856147.993
        }
    ]
}
```

자세한 내용은 Amazon Cognito 개발자 안내서의 Working with devices를 참조하시기 바랍니다.

- API 세부 정보는 AWS CLI 명령 참조의 ListDevices를 참조하세요.

list-groups

다음 코드 예시는 `list-groups`의 사용 방법을 보여 줍니다.

AWS CLI

사용자 풀의 그룹 나열

다음 `list-groups` 예제에서는 요청된 사용자 풀의 처음 두 그룹을 나열합니다.

```
aws cognito-idp list-groups \
    --user-pool-id us-west-2_EXAMPLE \
```

```
    --max-items 2
```

출력:

```
{
    "Groups": [
        {
            "CreationDate": 1681760899.633,
            "Description": "My test group",
            "GroupName": "testgroup",
            "LastModifiedDate": 1681760899.633,
            "Precedence": 1,
            "UserPoolId": "us-west-2_EXAMPLE"
        },
        {
            "CreationDate": 1642632749.051,
            "Description": "Autogenerated group for users who sign in using
 Facebook",
            "GroupName": "us-west-2_EXAMPLE_Facebook",
            "LastModifiedDate": 1642632749.051,
            "UserPoolId": "us-west-2_EXAMPLE"
        }
    ],
    "NextToken": "[Pagination token]"
}
```

자세한 내용은 Amazon Cognito 개발자 안내서의 사용자 풀에 그룹 추가를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListGroups를 참조하세요.

list-identity-providers

다음 코드 예시는 `list-identity-providers`의 사용 방법을 보여 줍니다.

AWS CLI

자격 증명 공급자 나열

다음 `list-identity-providers` 예제에서는 요청된 사용자 풀의 처음 두 자격 증명 공급자를 나열합니다.

```
aws cognito-idp list-identity-providers \
```

```
    --user-pool-id us-west-2_EXAMPLE \
    --max-items 2
```

출력:

```
{
    "Providers": [
        {
            "CreationDate": 1619477386.504,
            "LastModifiedDate": 1703798328.142,
            "ProviderName": "Azure",
            "ProviderType": "SAML"
        },
        {
            "CreationDate": 1642698776.175,
            "LastModifiedDate": 1642699086.453,
            "ProviderName": "LoginWithAmazon",
            "ProviderType": "LoginWithAmazon"
        }
    ],
    "NextToken": "[Pagination token]"
}
```

자세한 내용은 Amazon Cognito 개발자 안내서의 Third-party IdP sign-in을 참조하시기 바랍니다.

- API 세부 정보는 AWS CLI 명령 참조의 ListIdentityProviders를 참조하시기 바랍니다.

`list-resource-servers`

다음 코드 예시는 `list-resource-servers`의 사용 방법을 보여 줍니다.

AWS CLI

리소스 서버 나열

다음 `list-resource-servers` 예제에서는 요청된 사용자 풀의 처음 두 리소스 서버를 나열합니다.

```
aws cognito-idp list-resource-servers \
    --user-pool-id us-west-2_EXAMPLE \
    --max-results 2
```

출력:

```
{
    "ResourceServers": [
        {
            "Identifier": "myapi.example.com",
            "Name": "Example API with custom access control scopes",
            "Scopes": [
                {
                    "ScopeDescription": "International customers",
                    "ScopeName": "international.read"
                },
                {
                    "ScopeDescription": "Domestic customers",
                    "ScopeName": "domestic.read"
                }
            ],
            "UserPoolId": "us-west-2_EXAMPLE"
        },
        {
            "Identifier": "myapi2.example.com",
            "Name": "Another example API for access control",
            "Scopes": [
                {
                    "ScopeDescription": "B2B customers",
                    "ScopeName": "b2b.read"
                }
            ],
            "UserPoolId": "us-west-2_EXAMPLE"
        }
    ],
    "NextToken": "[Pagination token]"
}
```

자세한 내용은 Amazon Cognito 개발자 안내서의 Access control with resource servers를 참조하시기 바랍니다.

- API 세부 정보는 AWS CLI 명령 참조의 ListResourceServers를 참조하시기 바랍니다.

list-tags-for-resource

다음 코드 예시는 `list-tags-for-resource`의 사용 방법을 보여 줍니다.

AWS CLI

사용자 풀 태그 나열

다음 list-tags-for-resource 예제에서는 요청된 ARN과 함께 사용자 풀에 할당된 태그를 나열합니다.

```
aws cognito-idp list-tags-for-resource \
    --resource-arn arn:aws:cognito-idp:us-west-2:123456789012:userpool/us-west-2_EXAMPLE
```

출력:

```
{
    "Tags": {
        "administrator": "Jie",
        "tenant": "ExampleCorp"
    }
}
```

자세한 내용은 Amazon Cognito 개발자 안내서의 Tagging Amazon Cognito resources를 참조하시기 바랍니다.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource를 참조하세요.

list-user-import-jobs

다음 코드 예시는 list-user-import-jobs의 사용 방법을 보여 줍니다.

AWS CLI

사용자 가져오기 작업 및 상태 나열

다음 list-user-import-jobs 예제에서는 처음 3개의 사용자 가져오기 작업과 해당 세부 정보를 요청된 사용자 풀에 나열합니다.

```
aws cognito-idp list-user-import-jobs \
    --user-pool-id us-west-2_EXAMPLE \
    --max-results 3
```

출력:

```
{
    "PaginationToken": "us-west-2_EXAMPLE#import-example3#1667948397084",
    "UserImportJobs": [
        {
            "CloudWatchLogsRoleArn": "arn:aws:iam::123456789012:role/service-role/
Cognito-UserImport-Role",
            "CompletionDate": 1735329786.142,
            "CompletionMessage": "The user import job has expired.",
            "CreationDate": 1735241621.022,
            "FailedUsers": 0,
            "ImportedUsers": 0,
            "JobId": "import-example1",
            "JobName": "Test-import-job-1",
            "PreSignedUrl": "https://aws-cognito-idp-user-import-pdx.s3.us-
west-2.amazonaws.com/123456789012/us-west-2_EXAMPLE/import-mAgUtd8PMm?
X-Amz-Security-Token=[token]&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Date=20241226T193341Z&X-Amz-SignedHeaders=host%3Bx-amz-server-side-encryption&X-Amz-
Expires=899&X-Amz-Credential=[credential]&X-Amz-Signature=[signature]",
            "SkippedUsers": 0,
            "Status": "Expired",
            "UserPoolId": "us-west-2_EXAMPLE"
        },
        {
            "CloudWatchLogsRoleArn": "arn:aws:iam::123456789012:role/service-role/
Cognito-UserImport-Role",
            "CompletionDate": 1681509058.408,
            "CompletionMessage": "Too many users have failed or been skipped during
 the import.",
            "CreationDate": 1681509001.477,
            "FailedUsers": 1,
            "ImportedUsers": 0,
            "JobId": "import-example2",
            "JobName": "Test-import-job-2",
            "PreSignedUrl": "https://aws-cognito-idp-user-import-pdx.s3.us-
west-2.amazonaws.com/123456789012/us-west-2_EXAMPLE/import-mAgUtd8PMm?
X-Amz-Security-Token=[token]&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Date=20241226T193341Z&X-Amz-SignedHeaders=host%3Bx-amz-server-side-encryption&X-Amz-
Expires=899&X-Amz-Credential=[credential]&X-Amz-Signature=[signature]",
            "SkippedUsers": 0,
            "StartDate": 1681509057.965,
            "Status": "Failed",
            "UserPoolId": "us-west-2_EXAMPLE"
        },
```

```
        {
            "CloudWatchLogsRoleArn": "arn:aws:iam::123456789012:role/service-role/
Cognito-UserImport-Role",
            "CompletionDate": 1.667864578676E9,
            "CompletionMessage": "Import Job Completed Successfully.",
            "CreationDate": 1.667864480281E9,
            "FailedUsers": 0,
            "ImportedUsers": 6,
            "JobId": "import-example3",
            "JobName": "Test-import-job-3",
            "PreSignedUrl": "https://aws-cognito-idp-user-import-pdx.s3.us-
west-2.amazonaws.com/123456789012/us-west-2_EXAMPLE/import-mAgUtd8PMm?
X-Amz-Security-Token=[token]&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Date=20241226T193341Z&X-Amz-SignedHeaders=host%3Bx-amz-server-side-encryption&X-Amz-
Expires=899&X-Amz-Credential=[credential]&X-Amz-Signature=[signature]",
            "SkippedUsers": 0,
            "StartDate": 1.667864578167E9,
            "Status": "Succeeded",
            "UserPoolId": "us-west-2_EXAMPLE"
        }
    ]
}
```

자세한 내용은 Amazon Cognito 개발자 안내서의 Importing users from a CSV file을 참조하시기 바랍니다.

- API 세부 정보는 AWS CLI 명령 참조의 ListUserImportJobs 섹션을 참조하세요.

list-user-pool-clients

다음 코드 예시는 `list-user-pool-clients`의 사용 방법을 보여 줍니다.

AWS CLI

앱 클라이언트 나열

다음 `list-user-pool-clients` 예제에서는 요청된 사용자 풀의 처음 3개 앱 클라이언트를 나열합니다.

```
aws cognito-idp list-user-pool-clients \
    --user-pool-id us-west-2_EXAMPLE \
    --max-results 3
```

출력:

```
{
    "NextToken": "[Pagination token]",
    "UserPoolClients": [
        {
            "ClientId": "1example23456789",
            "ClientName": "app-client-1",
            "UserPoolId": "us-west-2_EXAMPLE"
        },
        {
            "ClientId": "2example34567890",
            "ClientName": "app-client-2",
            "UserPoolId": "us-west-2_EXAMPLE"
        },
        {
            "ClientId": "3example45678901",
            "ClientName": "app-client-3",
            "UserPoolId": "us-west-2_EXAMPLE"
        }
    ]
}
```

자세한 내용은 Amazon Cognito 개발자 안내서의 App clients를 참조하시기 바랍니다.

- API 세부 정보는 AWS CLI 명령 참조의 ListUserPoolClients를 참조하시기 바랍니다.

`list-user-pools`

다음 코드 예시는 `list-user-pools`의 사용 방법을 보여 줍니다.

AWS CLI

사용자 풀 나열

다음 `list-user-pools` 예제에서는 현재 CLI 자격 증명의 AWS 계정에서 사용 가능한 사용자 풀 3개를 나열합니다.

```
aws cognito-idp list-user-pools \
    --max-results 3
```

출력:

```
{
    "NextToken": "[Pagination token]",
    "UserPools": [
        {
            "CreationDate": 1681502497.741,
            "Id": "us-west-2_EXAMPLE1",
            "LambdaConfig": {
                "CustomMessage": "arn:aws:lambda:us-
east-1:123456789012:function:MyFunction",
                "PreSignUp": "arn:aws:lambda:us-
east-1:123456789012:function:MyFunction",
                "PreTokenGeneration": "arn:aws:lambda:us-
east-1:123456789012:function:MyFunction",
                "PreTokenGenerationConfig": {
                    "LambdaArn": "arn:aws:lambda:us-
east-1:123456789012:function:MyFunction",
                    "LambdaVersion": "V1_0"
                }
            },
            "LastModifiedDate": 1681502497.741,
            "Name": "user pool 1"
        },
        {
            "CreationDate": 1686064178.717,
            "Id": "us-west-2_EXAMPLE2",
            "LambdaConfig": {
            },
            "LastModifiedDate": 1686064178.873,
            "Name": "user pool 2"
        },
        {
            "CreationDate": 1627681712.237,
            "Id": "us-west-2_EXAMPLE3",
            "LambdaConfig": {
                "UserMigration": "arn:aws:lambda:us-
east-1:123456789012:function:MyFunction"
            },
            "LastModifiedDate": 1678486942.479,
            "Name": "user pool 3"
        }
    ]
}
```

자세한 내용은 Amazon Cognito 개발자 안내서의 Amazon Cognito 사용자 풀을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조에서 ListUserPools 섹션을 참조하세요.

list-users-in-group

다음 코드 예시에서는 `list-users-in-group` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

그룹의 사용자를 나열하는 방법

이 예시에서는 MyGroup 그룹의 사용자를 나열합니다.

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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListUsersInGroup 섹션을 참조하세요.

list-users

다음 코드 예시는 list-users의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 서버 측 필터를 사용하여 사용자 나열

다음 list-users 예제에서는 이메일 주소가 testuser로 시작하는 요청된 사용자 풀의 사용자 3명을 나열합니다.

```
aws cognito-idp list-users \
    --user-pool-id us-west-2_EXAMPLE \
    --filter email^=\"testuser\" \
    --max-items 3
```

출력:


```
{
    "PaginationToken": "efgh5678EXAMPLE",
    "Users": [
        {
            "Attributes": [
                {
                    "Name": "sub",
                    "Value": "eaad0219-2117-439f-8d46-4db20e59268f"
                },
                {
                    "Name": "email",
                    "Value": "testuser@example.com"
                }
            ],
            "Enabled": true,
            "UserCreateDate": 1682955829.578,
            "UserLastModifiedDate": 1689030181.63,
            "UserStatus": "CONFIRMED",
            "Username": "testuser"
        },
        {
            "Attributes": [
                {
                    "Name": "sub",
                    "Value": "3b994cfd-0b07-4581-be46-3c82f9a70c90"
                },
                {
                    "Name": "email",
                    "Value": "testuser2@example.com"
                }
            ],
            "Enabled": true,
            "UserCreateDate": 1684427979.201,
            "UserLastModifiedDate": 1684427979.201,
            "UserStatus": "UNCONFIRMED",
            "Username": "testuser2"
        },
        {
            "Attributes": [
                {
                    "Name": "sub",
                    "Value": "5929e0d1-4c34-42d1-9b79-a5ecacfe66f7"
```

```
                },
                {
                    "Name": "email",
                    "Value": "testuser3@example.com"
                }
            ],
            "Enabled": true,
            "UserCreateDate": 1684427823.641,
            "UserLastModifiedDate": 1684427823.641,
            "UserStatus": "UNCONFIRMED",
            "Username": "testuser3@example.com"
        }
    ]
}
```

자세한 내용은 Amazon Cognito 개발자 안내서의 Managing and searching for users를 참조하시기 바랍니다.

예제 2: 클라이언트 측 필터를 사용하여 사용자 나열

다음 `list-users` 예제에서는 이메일 도메인 '@example.com'이 포함된 이메일 주소와 같은 속성을 가진 사용자 3명의 속성을 나열합니다. 다른 속성에 이 문자열이 포함된 경우 해당 속성도 표시됩니다. 두 번째 사용자는 쿼리와 일치하는 속성이 없으며 표시된 출력에서 제외되지만 서버 응답에서는 제외되지 않습니다.

```
aws cognito-idp list-users \
    --user-pool-id us-west-2_EXAMPLE \
    --max-items 3
    --query Users\[\*\].Attributes\[\?Value\.contains\(\@\,\'@example.com\'\)\]
```

출력:

```
[
    [
        {
            "Name": "email",
            "Value": "admin@example.com"
        }
    ],
    [],
    [
        {
```

```
            "Name": "email",
            "Value": "operator@example.com"
        }
    ]
]
```

자세한 내용은 Amazon Cognito 개발자 안내서의 Managing and searching for users를 참조하시기 바랍니다.

- API 세부 정보는 AWS CLI 명령 참조의 ListUsers를 참조하세요.

list-web-authn-credentials

다음 코드 예시는 `list-web-authn-credentials`의 사용 방법을 보여 줍니다.

AWS CLI

패스키 자격 증명 나열

다음 `list-web-authn-credentials` 예제에서는 현재 사용자의 패스키 또는 WebAuthn 자격 증명을 나열합니다. 등록된 디바이스가 하나 있습니다.

```
aws cognito-idp list-web-authn-credentials \
    --access-token eyJra456defEXAMPLE
```

출력:

```
{
    "Credentials": [
        {
            "AuthenticatorAttachment": "cross-platform",
            "CreatedAt": 1736293876.115,
            "CredentialId": "8LApgk4-lNUFHbhm2w6Und7-
uxcc8coJGsPxiogvHoItc64xWQc3r4CEXAMPLE",
            "FriendlyCredentialName": "Roaming passkey",
            "RelyingPartyId": "auth.example.com"
        }
    ]
}
```

자세한 내용은 Amazon Cognito 개발자 안내서의 Passkey sign-in을 참조하시기 바랍니다.

- API 세부 정보는 AWS CLI 명령 참조의 ListWebAuthnCredentials를 참조하시기 바랍니다.

resend-confirmation-code

다음 코드 예시는 resend-confirmation-code의 사용 방법을 보여 줍니다.

AWS CLI

확인 코드 다시 보내기

다음 resend-confirmation-code 예시에서는 사용자 jane에게 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.

- API 세부 정보는 AWS CLI 명령 참조에서 ResendConfirmationCode 섹션을 참조하세요.

respond-to-auth-challenge

다음 코드 예시는 respond-to-auth-challenge의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: NEW_PASSWORD_REQUIRED 과제에 응답

다음 respond-to-auth-challenge 예제에서는 초기 인증이 반환된 NEW_PASSWORD_REQUIRED 챌린지에 응답합니다. 사용자(jane@example.com)용 새 암호를 설정합니다.

```
aws cognito-idp respond-to-auth-challenge \
    --client-id 1example23456789 \
    --challenge-name NEW_PASSWORD_REQUIRED \
```

```
    --challenge-responses USERNAME=jane@example.com,NEW_PASSWORD=[Password] \
    --session AYABeEv5HklEXAMPLE
```

출력:

```
{
    "ChallengeParameters": {},
    "AuthenticationResult": {
        "AccessToken": "ACCESS_TOKEN",
        "ExpiresIn": 3600,
        "TokenType": "Bearer",
        "RefreshToken": "REFRESH_TOKEN",
        "IdToken": "ID_TOKEN",
        "NewDeviceMetadata": {
            "DeviceKey": "us-west-2_a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "DeviceGroupKey": "-wt2ha1Zd"
        }
    }
}
```

자세한 내용은 Amazon Cognito 개발자 안내서의 Authentication을 참조하시기 바랍니다.

예제 2: SELECT_MFA_TYPE 챌린지에 응답

다음 `respond-to-auth-challenge` 예제에서는 TOTP MFA를 현재 사용자의 MFA 옵션으로 선택합니다. 사용자에게 MFA 유형을 선택하라는 메시지가 표시되고 다음으로 MFA 코드를 입력하라는 메시지가 표시됩니다.

```
aws cognito-idp respond-to-auth-challenge \
    --client-id 1example23456789
    --session AYABeEv5HklEXAMPLE
    --challenge-name SELECT_MFA_TYPE
    --challenge-responses USERNAME=testuser,ANSWER=SOFTWARE_TOKEN_MFA
```

출력:

```
{
    "ChallengeName": "SOFTWARE_TOKEN_MFA",
    "Session": "AYABeEv5HklEXAMPLE",
    "ChallengeParameters": {
        "FRIENDLY_DEVICE_NAME": "transparent"
    }
```

```
}
```

자세한 내용은 Amazon Cognito 개발자 안내서의 Adding MFA를 참조하시기 바랍니다.

예제 3: SOFTWARE_TOKEN_MFA 챌린지에 응답

다음 `respond-to-auth-challenge` 예제에서는 TOTP MFA 코드를 제공하고 로그인을 완료합니다.

```
aws cognito-idp respond-to-auth-challenge \
    --client-id 1example23456789 \
    --session AYABeEv5HklEXAMPLE \
    --challenge-name SOFTWARE_TOKEN_MFA \
    --challenge-responses USERNAME=testuser,SOFTWARE_TOKEN_MFA_CODE=123456
```

출력:

```
{
    "AuthenticationResult": {
        "AccessToken": "eyJra456defEXAMPLE",
        "ExpiresIn": 3600,
        "TokenType": "Bearer",
        "RefreshToken": "eyJra123abcEXAMPLE",
        "IdToken": "eyJra789ghiEXAMPLE",
        "NewDeviceMetadata": {
            "DeviceKey": "us-west-2_a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "DeviceGroupKey": "-v7w9UcY6"
        }
    }
}
```

자세한 내용은 Amazon Cognito 개발자 안내서의 Adding MFA를 참조하시기 바랍니다.

- API 세부 정보는 AWS CLI 명령 참조에서 RespondToAuthChallenge 섹션을 참조하세요.

`revoke-token`

다음 코드 예시는 `revoke-token`의 사용 방법을 보여 줍니다.

AWS CLI

새로 고침 토큰 취소

다음 revoke-token는 요청된 새로 고침 토큰 및 연결된 액세스 토큰을 취소합니다.

```
aws cognito-idp revoke-token \
    --token eyJjd123abcEXAMPLE \
    --client-id 1example23456789
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Cognito 개발자 안내서의 Revoking tokens 섹션을 참조하시기 바랍니다.

- API 세부 정보는 AWS CLI 명령 참조의 RevokeToken을 참조하시기 바랍니다.

set-log-delivery-configuration

다음 코드 예시는 set-log-delivery-configuration의 사용 방법을 보여 줍니다.

AWS CLI

사용자 풀에서 로그 내보내기 설정

다음 set-log-delivery-configuration 예제에서는 로그 그룹에 대한 사용자 알림 오류 로깅 및 S3 버킷에 대한 사용자 인증 정보 로깅으로 요청된 사용자 풀을 구성합니다.

```
aws cognito-idp set-log-delivery-configuration \
    --user-pool-id us-west-2_EXAMPLE \
    --log-
configurations LogLevel=ERROR,EventSource=userNotification,CloudWatchLogsConfiguration={LogG
west-2:123456789012:log-group:cognito-
exported} LogLevel=INFO,EventSource=userAuthEvents,S3Configuration={BucketArn=arn:aws:s3:::a
s3-demo-bucket1}
```

출력:

```
{
   "LogDeliveryConfiguration": {
        "LogConfigurations": [
            {
                "CloudWatchLogsConfiguration": {
                    "LogGroupArn": "arn:aws:logs:us-west-2:123456789012:log-
group:cognito-exported"
                },
                "EventSource": "userNotification",
                "LogLevel": "ERROR"
```

```
            },
            {
                "EventSource": "userAuthEvents",
                "LogLevel": "INFO",
                "S3Configuration": {
                    "BucketArn": "arn:aws:s3:::amzn-s3-demo-bucket1"
                }
            }
        ],
        "UserPoolId": "us-west-2_EXAMPLE"
    }
}
```

자세한 내용은 Amazon Cognito 개발자 안내서의 Exporting user pool logs를 참조하시기 바랍니다.

- API 세부 정보는 AWS CLI 명령 참조의 SetLogDeliveryConfiguration을 참조하시기 바랍니다.

set-risk-configuration

다음 코드 예시는 `set-risk-configuration`의 사용 방법을 보여 줍니다.

AWS CLI

위협 방지 위험 구성 설정

다음 `set-risk-configuration` 예제에서는 요청된 앱 클라이언트에서 위협 방지 메시지 및 작업, 손상된 자격 증명 및 IP 주소 예외를 구성합니다. NotifyConfiguration 객체의 복잡성으로 인해 이 명령의 모범 사례는 JSON 입력입니다.

```
aws cognito-idp set-risk-configuration \
    --cli-input-json file://set-risk-configuration.json
```

`set-risk-configuration.json`의 콘텐츠:

```
{
    "AccountTakeoverRiskConfiguration": {
        "Actions": {
            "HighAction": {
                "EventAction": "MFA_REQUIRED",
                "Notify": true
            },
            "LowAction": {
                "EventAction": "NO_ACTION",
```

```
                "Notify": true
            },
            "MediumAction": {
                "EventAction": "MFA_IF_CONFIGURED",
                "Notify": true
            }
        },
        "NotifyConfiguration": {
            "BlockEmail": {
                "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>",
                "Subject": "Blocked sign-in attempt",
                "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"
            },
            "From": "admin@example.com",
            "MfaEmail": {
                "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>",
                "Subject": "New sign-in attempt",
                "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"
            },
            "NoActionEmail": {
                "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>",
                "Subject": "New sign-in attempt",
                "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"
            },
            "ReplyTo": "admin@example.com",
            "SourceArn": "arn:aws:ses:us-west-2:123456789012:identity/
admin@example.com"
        }
    },
    "ClientId": "1example23456789",
    "CompromisedCredentialsRiskConfiguration": {
        "Actions": {
            "EventAction": "BLOCK"
        },
        "EventFilter": [
            "PASSWORD_CHANGE",
            "SIGN_UP",
            "SIGN_IN"
        ]
    },
    "RiskExceptionConfiguration": {
        "BlockedIPRangeList": [
            "192.0.2.1/32",
            "192.0.2.2/32"
        ],
        "SkippedIPRangeList": [
            "203.0.113.1/32",
            "203.0.113.2/32"
        ]
    },
    "UserPoolId": "us-west-2_EXAMPLE"
}
```

출력:

```
{
    "RiskConfiguration": {
        "AccountTakeoverRiskConfiguration": {
            "Actions": {
                "HighAction": {
                    "EventAction": "MFA_REQUIRED",
                    "Notify": true
                },
                "LowAction": {
                    "EventAction": "NO_ACTION",
                    "Notify": true
                },
                "MediumAction": {
                    "EventAction": "MFA_IF_CONFIGURED",
                    "Notify": true
                }
            },
            "NotifyConfiguration": {
                "BlockEmail": {
                    "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>",
                    "Subject": "Blocked sign-in attempt",
                    "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"
                },
                "From": "admin@example.com",
                "MfaEmail": {
                    "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>",
                    "Subject": "New sign-in attempt",
                    "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"
                },
                "NoActionEmail": {
                    "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>",
                    "Subject": "New sign-in attempt",
                    "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"
                },
                "ReplyTo": "admin@example.com",
                "SourceArn": "arn:aws:ses:us-west-2:123456789012:identity/
admin@example.com"
            }
        },
        "ClientId": "1example23456789",
        "CompromisedCredentialsRiskConfiguration": {
            "Actions": {
                "EventAction": "BLOCK"
            },
            "EventFilter": [
                "PASSWORD_CHANGE",
                "SIGN_UP",
                "SIGN_IN"
            ]
        },
        "RiskExceptionConfiguration": {
            "BlockedIPRangeList": [
                "192.0.2.1/32",
```

```
                "192.0.2.2/32"
            ],
            "SkippedIPRangeList": [
                "203.0.113.1/32",
                "203.0.113.2/32"
            ]
        },
        "UserPoolId": "us-west-2_EXAMPLE"
    }
}
```

자세한 내용은 Amazon Cognito 개발자 안내서의 Threat protection을 참조하시기 바랍니다.

- API 세부 정보는 AWS CLI 명령 참조의 SetRiskConfiguration 섹션을 참조하세요.

set-ui-customization

다음 코드 예시는 `set-ui-customization`의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 앱 클라이언트의 클래식 호스팅 UI를 사용자 지정

다음 `set-ui-customization` 예제에서는 일부 사용자 지정 CSS와 Amazon Cognito 로고를 애플리케이션 로고로 사용하여 요청된 앱 클라이언트를 구성합니다.

```
aws cognito-idp set-ui-customization \
    --user-pool-id us-west-2_ywDJHlIfU \
    --client-id 14pq32c5q2uq2q7keorloqvb23 \
    --css ".logo-customizable {\n\tmax-width: 60%;\n\tmax-height: 30%;\n}\n.banner-customizable {\n\tpadding: 25px 0px 25px 0px;\n\tbackground-color: lightgray;\n}\n.label-customizable {\n\tfont-weight: 400;\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: 11px;\n\tfont-weight: normal;\n\tmargin: 20px -15px 10px -13px;\n\theight: 40px;\n\twidth: 108%;\n\tcolor: #fff;\n\tbackground-color: #337ab7;\n\ttext-align: center;\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\tborder-radius:
```

```
 0px;\n}\n.inputField-customizable:focus {\n\tborder-color: #66afe9;\n\toutline:
 0;\n}\n.idpButton-customizable {\n\theight: 40px;\n\twidth: 100%;\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
\tborder-radius: 2px;\n\theight: 40px;\n\tmargin-bottom: 15px;\n\tpadding: 1px;
\n\ttext-align: left;\n\twidth: 100%;\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: #fff;\n}\n" \
    --image-
file iVBORw0KGgoAAAANSUhEUgAAAFAAAABQCAMAAAC5zwKfAAAAAXNSR0IArs4c6QAAAARnQU1BAACxjwv8YQUAAAA
Cmsfvm6f3y9P////fM0uqAj
+yNmu6ZpvnZ3eNabuFNYuZneehzhPKzvPTAxwAAAOiMMlkAAAASdFJOU///////////////////////
AOK/vxIAAAAJcEhZcwAADsMAAA7DAcdvqGQAAAKDSURBVFhH7ZfpkoMgEISDHKuEw/
d/2u2BQWMiBrG29o+fVsKatdPMAeZxc3Nz8w+ISekzmB++sYIw/I/
tjHzrPpO2Tx62EbR2PNxFac+jVuKxRaV50IzXkUe76NOCoUuwlvnQKei02gNF0ykotOLRBq/
nboeWRxAISx2EbsHFoRhK6Igk2JJlwScfQjgt06dOaWWiTbEDAe/
iq8N9kqCw2uCbHkHlYkaXEF8EYeL9RDqT4FhC6XMIIEifdcUwCc4leNyhabadWU6OlKYJE1Oac3NSPhB5rlaXlSgmr/1
F0L6Q5pZiSG0SfZTSTCOUhxOCH1AdIoCpTTIjtd
+VpEjUDDytQH/0Fpc661Aisas/4qmyUItD557pSCOSQQzlx27J
+meyDGc5zZgfhWuXE1lGgmVOMwmWdeGdzhjqZV14x5vSj7vsC5JDz/Cl0Vhp56n2NQt1wQIpury1EPbwyaYm
+IhmAQKoajkH51wg4cMZ1wQ3QG9efKWWOaDhYWnU6jXjCMdRmm21PArI
+Pb5DYoH93hq0ZCPlxeGJho/DI15C6sQc/L2sTC47UFBKZGHT6k+zlXg7WebA0Nr0HTcLMfk/
Y4Rc65D3iG6WDd7YLSlVqk87bVhUwhnClrx11RsVQwlAA818Mn
+QEs71BhSFU6orsUfKhHp72XMGYXi4q9c64RXRvzkWurRfG2vI2be/VaNcNgpX0Evb/
vio7nPMmj5qujkpQgSaPd1UcVqciHFDNZpOcGlcOPyi+AamCbIL9fitxAGeFN2Dl
+3vZubm5u/4fH4Bd14HhIPdwZPAAAAAElFTkSuQmCC
```

출력:

```
{
    "UICustomization": {
        "UserPoolId": "us-west-2_ywDJHlIfU",
        "ClientId": "14pq32c5q2uq2q7keorloqvb23",
        "ImageUrl": "https://
cf.thewrong.club/14pq32c5q2uq2q7keorloqvb23/20250117005911/assets/images/image.jpg",
        "CSS": ".logo-customizable {\n\tmax-width: 60%;\n\tmax-height: 30%;
\n}\n.banner-customizable {\n\tpadding: 25px 0px 25px 0px;\n\tbackground-color:
 lightgray;\n}\n.label-customizable {\n\tfont-weight: 400;\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: 11px;\n\tfont-weight: normal;\n\tmargin: 20px -15px 10px -13px;
\n\theight: 40px;\n\twidth: 108%;\n\tcolor: #fff;\n\tbackground-color: #337ab7;
\n\ttext-align: center;\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\tborder-radius: 0px;\n}\n.inputField-customizable:focus {\n
\tborder-color: #66afe9;\n\toutline: 0;\n}\n.idpButton-customizable {\n\theight:
 40px;\n\twidth: 100%;\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\tborder-radius: 2px;\n\theight: 40px;\n\tmargin-
bottom: 15px;\n\tpadding: 1px;\n\ttext-align: left;\n\twidth: 100%;\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: #fff;\n}\n",
        "CSSVersion": "20250117005911"
    }
}
```

예제 2: 모든 앱 클라이언트에 대한 기본 UI 사용자 지정을 설정

다음 `set-ui-customization` 예제에서는 클라이언트별 구성이 없는 모든 앱 클라이언트에 대해 요청된 사용자 풀을 구성합니다. 명령은 Amazon Cognito 로고를 애플리케이션 로고로 사용하여 일부 사용자 지정 CSS를 적용합니다.

```
aws cognito-idp set-ui-customization \
--user-pool-id us-west-2_ywDJHlIfU \
--client-id ALL \
--css ".logo-customizable {\n\tmax-width: 60%;\n\tmax-height: 30%;\n}\n.banner-
customizable {\n\tpadding: 25px 0px 25px 0px;\n\tbackground-color: lightgray;
\n}\n.label-customizable {\n\tfont-weight: 400;\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: 11px;\n\tfont-weight: normal;\n\tmargin: 20px -15px 10px -13px;\n\theight:
 40px;\n\twidth: 108%;\n\tcolor: #fff;\n\tbackground-color: #337ab7;\n\ttext-align:
 center;\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\tborder-radius:
0px;\n}\n.inputField-customizable:focus {\n\tborder-color: #66afe9;\n\toutline:
0;\n}\n.idpButton-customizable {\n\theight: 40px;\n\twidth: 100%;\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
\tborder-radius: 2px;\n\theight: 40px;\n\tmargin-bottom: 15px;\n\tpadding: 1px;
\n\ttext-align: left;\n\twidth: 100%;\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: #fff;\n}\n" \
--image-
file iVBORw0KGgoAAAANSUhEUgAAAFAAAABQCAMAAAC5zwKfAAAAAXNSR0IArs4c6QAAAARnQU1BAACxjwv8YQUAAAA
Cmsfvm6f3y9P////fM0uqAj
+yNmu6ZpvnZ3eNabuFNYuZneehzhPKzvPTAxwAAAOiMMlkAAAASdFJOU//////////////////////
AOK/vxIAAAAJcEhZcwAADsMAAA7DAcdvqGQAAAKDSURBVFhH7ZfpkoMgEISDHKuEw/
d/2u2BQWMiBrG29o+fVsKatdPMAeZxc3Nz8w+ISekzmB++sYIw/I/
tjHzrPpO2Tx62EbR2PNxFac+jVuKxRaV50IzXkUe76NOCoUuwlvnQKei02gNF0ykotOLRBq/
nboeWRxAISx2EbsHFoRhK6Igk2JJlwScfQjgt06dOaWWiTbEDAe/
iq8N9kqCw2uCbHkHlYkaXEF8EYeL9RDqT4FhC6XMIIEifdcUwCc4leNyhabadWU6OlKYJE1Oac3NSPhB5rlaXlSgmr/1
F0L6Q5pZiSG0SfZTSTCOUhxOCH1AdIoCpTTIjtd
+VpEjUDDytQH/0Fpc661Aisas/4qmyUItD557pSCOSQQzlx27J
+meyDGc5zZgfhWuXE1lGgmVOMwmWdeGdzhjqZV14x5vSj7vsC5JDz/Cl0Vhp56n2NQt1wQIpury1EPbwyaYm
+IhmAQKoajkH51wg4cMZ1wQ3QG9efKWWOaDhYWnU6jXjCMdRmm21PArI
+Pb5DYoH93hq0ZCPlxeGJho/DI15C6sQc/L2sTC47UFBKZGHT6k+zlXg7WebA0Nr0HTcLMfk/
Y4Rc65D3iG6WDd7YLSlVqk87bVhUwhnClrx11RsVQwlAA818Mn
+QEs71BhSFU6orsUfKhHp72XMGYXi4q9c64RXRvzkWurRfG2vI2be/VaNcNgpX0Evb/
vio7nPMmj5qujkpQgSaPd1UcVqciHFDNZpOcGlcOPyi+AamCbIL9fitxAGeFN2Dl
+3vZubm5u/4fH4Bd14HhIPdwZPAAAAAElFTkSuQmCC
```

출력:

```
{
    "UICustomization": {
        "UserPoolId": "us-west-2_ywDJHlIfU",
        "ClientId": "14pq32c5q2uq2q7keorloqvb23",
        "ImageUrl": "https://
cf.thewrong.club/14pq32c5q2uq2q7keorloqvb23/20250117005911/assets/images/image.jpg",
        "CSS": ".logo-customizable {\n\tmax-width: 60%;\n\tmax-height: 30%;
\n}\n.banner-customizable {\n\tpadding: 25px 0px 25px 0px;\n\tbackground-color:
 lightgray;\n}\n.label-customizable {\n\tfont-weight: 400;\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: 11px;\n\tfont-weight: normal;\n\tmargin: 20px -15px 10px -13px;
\n\theight: 40px;\n\twidth: 108%;\n\tcolor: #fff;\n\tbackground-color: #337ab7;
\n\ttext-align: center;\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\tborder-radius: 0px;\n}\n.inputField-customizable:focus {\n
\tborder-color: #66afe9;\n\toutline: 0;\n}\n.idpButton-customizable {\n\theight:
 40px;\n\twidth: 100%;\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\tborder-radius: 2px;\n\theight: 40px;\n\tmargin-
bottom: 15px;\n\tpadding: 1px;\n\ttext-align: left;\n\twidth: 100%;\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: #fff;\n}\n",
        "CSSVersion": "20250117005911"
    }
}
```

자세한 내용은 Amazon Cognito 개발자 안내서의 Hosted UI (classic) branding을 참조하시기 바랍니다.

- API 세부 정보는 AWS CLI 명령 참조의 SetUiCustomization 섹션을 참조하세요.

`set-user-mfa-preference`

다음 코드 예시는 `set-user-mfa-preference`의 사용 방법을 보여 줍니다.

AWS CLI

사용자 MFA 기본 설정 지정

다음 `set-user-mfa-preference` 예제에서는 TOTP MFA를 사용하도록 현재 사용자를 구성하고 다른 모든 MFA 요소를 사용 해제합니다.

```
aws cognito-idp set-user-mfa-preference \
    --access-token eyJra456defEXAMPLE \
    --software-token-mfa-settings Enabled=true,PreferredMfa=true \
```

```
    --sms-mfa-settings Enabled=false,PreferredMfa=false \
    --email-mfa-settings Enabled=false,PreferredMfa=false
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Cognito 개발자 안내서의 Adding MFA를 참조하시기 바랍니다.

- API 세부 정보는 AWS CLI 명령 참조의 SetUserMfaPreference 섹션을 참조하세요.

`set-user-pool-mfa-config`

다음 코드 예시는 `set-user-pool-mfa-config`의 사용 방법을 보여 줍니다.

AWS CLI

사용자 풀 MFA 및 WebAuthn 구성

다음 `set-user-pool-mfa-config` 예제에서는 모든 사용 가능한 MFA 메서드를 사용하여 MFA(선택 사항)로 요청된 사용자 풀을 구성하고 WebAuthn 구성을 설정합니다.

```
aws cognito-idp set-user-pool-mfa-config \
    --user-pool-id us-west-2_EXAMPLE \
    --sms-mfa-configuration "SmsAuthenticationMessage=\"Your OTP for MFA or sign-
in: use {####}.\",SmsConfiguration={SnsCallerArn=arn:aws:iam::123456789012:role/
service-role/test-SMS-Role,ExternalId=a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111,SnsRegion=us-west-2}" \
    --software-token-mfa-configuration Enabled=true \
    --email-mfa-configuration "Message=\"Your OTP for MFA or sign-in: use
 {####}\",Subject=\"OTP test\"" \
    --mfa-configuration OPTIONAL \
    --web-authn-
configuration RelyingPartyId=auth.example.com,UserVerification=preferred
```

출력:

```
{
    "EmailMfaConfiguration": {
        "Message": "Your OTP for MFA or sign-in: use {####}",
        "Subject": "OTP test"
    },
    "MfaConfiguration": "OPTIONAL",
    "SmsMfaConfiguration": {
```

```
        "SmsAuthenticationMessage": "Your OTP for MFA or sign-in: use {####}.",
        "SmsConfiguration": {
            "ExternalId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "SnsCallerArn": "arn:aws:iam::123456789012:role/service-role/test-SMS-
Role",
            "SnsRegion": "us-west-2"
        }
    },
    "SoftwareTokenMfaConfiguration": {
        "Enabled": true
    },
    "WebAuthnConfiguration": {
        "RelyingPartyId": "auth.example.com",
        "UserVerification": "preferred"
    }
}
```

자세한 내용은 Amazon Cognito 개발자 안내서의 Adding MFA 및 Passkey sign-in을 참조하시기 바랍니다.

- API 세부 정보는 AWS CLI 명령 참조의 SetUserPoolMfaConfig를 참조하시기 바랍니다.

set-user-settings

다음 코드 예시는 set-user-settings의 사용 방법을 보여 줍니다.

AWS CLI

사용자 설정 구성

이 예시에서는 MFA 배달 기본 설정을 EMAIL로 설정합니다.

명령:

```
aws cognito-idp set-user-settings --access-token ACCESS_TOKEN --mfa-
options DeliveryMedium=EMAIL
```

- API 세부 정보는 AWS CLI 명령 참조의 SetUserSettings 섹션을 참조하세요.

sign-up

다음 코드 예시에서는 sign-up 코드를 사용하는 방법을 보여줍니다.

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"
}
```

- API 세부 정보는 AWS CLI 명령 참조에서 SignUp 섹션을 참조하세요.

start-user-import-job

다음 코드 예시는 `start-user-import-job`의 사용 방법을 보여 줍니다.

AWS CLI

가져오기 작업 시작

다음 `start-user-import-job` 예제에서는 요청된 사용자 풀에서 요청된 가져오기 작업을 시작합니다.

```
aws cognito-idp start-user-import-job \
    --user-pool-id us-west-2_EXAMPLE \
    --job-id import-mAgUtd8PMm
```

출력:

```
{
    "UserImportJob": {
        "CloudWatchLogsRoleArn": "arn:aws:iam::123456789012:role/example-cloudwatch-
logs-role",
```

```
        "CreationDate": 1736442975.904,
        "FailedUsers": 0,
        "ImportedUsers": 0,
        "JobId": "import-mAgUtd8PMm",
        "JobName": "Customer import",
        "PreSignedUrl": "https://aws-cognito-idp-user-import-pdx.s3.us-
west-2.amazonaws.com/123456789012/us-west-2_EXAMPLE/import-mAgUtd8PMm?
X-Amz-Security-Token=[token]&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Date=20241226T193341Z&X-Amz-SignedHeaders=host%3Bx-amz-server-side-encryption&X-Amz-
Expires=899&X-Amz-Credential=[credential]&X-Amz-Signature=[signature]",
        "SkippedUsers": 0,
        "StartDate": 1736443020.081,
        "Status": "Pending",
        "UserPoolId": "us-west-2_EXAMPLE"
    }
}
```

자세한 내용은 Amazon Cognito 개발자 안내서의 Importing users into a user pool을 참조하시기 바랍니다.

- API 세부 정보는 AWS CLI 명령 참조의 StartUserImportJob 섹션을 참조하세요.

start-web-authn-registration

다음 코드 예시는 start-web-authn-registration의 사용 방법을 보여 줍니다.

AWS CLI

로그인한 사용자의 패스키 등록 정보를 가져오려면 다음과 같이 합니다.

다음 start-web-authn-registration 예제에서는 현재 사용자에 대한 WebAuthn 등록 옵션을 생성합니다.

```
aws cognito-idp start-web-authn-registration \
    --access-token eyJra456defEXAMPLE
```

출력:

```
{
    "CredentialCreationOptions": {
        "authenticatorSelection": {
            "requireResidentKey": true,
            "residentKey": "required",
```

```
                "userVerification": "preferred"
            },
            "challenge": "wxvbDicyqQqvF2EXAMPLE",
            "excludeCredentials": [
                {
                    "id": "8LApgk4-lNUFHbhm2w6Und7-
uxcc8coJGsPxiogvHoItc64xWQc3r4CEXAMPLE",
                    "type": "public-key"
                }
            ],
            "pubKeyCredParams": [
                {
                    "alg": -7,
                    "type": "public-key"
                },
                {
                    "alg": -257,
                    "type": "public-key"
                }
            ],
            "rp": {
                "id": "auth.example.com",
                "name": "auth.example.com"
            },
            "timeout": 60000,
            "user": {
                "displayName": "testuser",
                "id": "ZWFhZDAyMTktMjExNy00MzlmLThkNDYtNGRiMjBlNEXAMPLE",
                "name": "testuser"
            }
        }
}
```

자세한 내용은 Amazon Cognito 개발자 안내서의 Passkey sign-in을 참조하시기 바랍니다.

- API 세부 정보는 AWS CLI 명령 참조의 StartWebAuthnRegistration을 참조하시기 바랍니다.

`stop-user-import-job`

다음 코드 예시는 `stop-user-import-job`의 사용 방법을 보여 줍니다.

AWS CLI

가져오기 작업 중지

다음 `stop-user-import-job` 예제에서는 요청된 사용자 풀에서 요청된 실행 중인 사용자 가져오기 작업을 중지합니다.

```
aws cognito-idp stop-user-import-job \
    --user-pool-id us-west-2_EXAMPLE \
    --job-id import-mAgUtd8PMm
```

출력:

```
{
    "UserImportJob": {
        "CloudWatchLogsRoleArn": "arn:aws:iam::123456789012:role/example-cloudwatch-logs-role",
        "CompletionDate": 1736443496.379,
        "CompletionMessage": "The Import Job was stopped by the developer.",
        "CreationDate": 1736443471.781,
        "FailedUsers": 0,
        "ImportedUsers": 0,
        "JobId": "import-mAgUtd8PMm",
        "JobName": "Customer import",
        "PreSignedUrl": "https://aws-cognito-idp-user-import-pdx.s3.us-west-2.amazonaws.com/123456789012/us-west-2_EXAMPLE/import-mAgUtd8PMm?X-Amz-Security-Token=[token]&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20241226T193341Z&X-Amz-SignedHeaders=host%3Bx-amz-server-side-encryption&X-Amz-Expires=899&X-Amz-Credential=[credential]&X-Amz-Signature=[signature]",
        "SkippedUsers": 0,
        "StartDate": 1736443494.154,
        "Status": "Stopped",
        "UserPoolId": "us-west-2_EXAMPLE"
    }
}
```

자세한 내용은 Amazon Cognito 개발자 안내서의 [Importing users into a user pool]을 참조하시기 바랍니다.

- API 세부 정보는 AWS CLI 명령 참조의 [StopUserImportJob] 섹션을 참조하세요.

`tag-resource`

다음 코드 예시는 `tag-resource`의 사용 방법을 보여 줍니다.

AWS CLI

사용자 풀에 태그 지정

다음 `tag-resource` 예제에서는 요청된 사용자 풀에 `administrator` 및 `department` 태그를 적용합니다.

```
aws cognito-idp tag-resource \
    --resource-arn arn:aws:cognito-idp:us-west-2:123456789012:userpool/us-west-2_EXAMPLE \
    --tags administrator=Jie,tenant=ExampleCorp
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Cognito 개발자 안내서의 Tagging Amazon Cognito resources를 참조하시기 바랍니다.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

`untag-resource`

다음 코드 예시는 `untag-resource`의 사용 방법을 보여 줍니다.

AWS CLI

사용자 풀에서 태그 제거

다음 `untag-resource` 예제에서는 요청된 사용자 풀에서 `administrator` 및 `department` 태그를 제거합니다.

```
aws cognito-idp untag-resource \
    --resource-arn arn:aws:cognito-idp:us-west-2:767671399759:userpool/us-west-2_l5cxwdm2K \
    --tag-keys administrator tenant
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Cognito 개발자 안내서의 Tagging Amazon Cognito resources를 참조하시기 바랍니다.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

update-auth-event-feedback

다음 코드 예시는 update-auth-event-feedback의 사용 방법을 보여 줍니다.

AWS CLI

인증 이벤트 피드백을 업데이트하는 방법

이 예시에서는 권한 부여 이벤트 피드백을 업데이트합니다. 이벤트를 'Valid'로 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"
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateAuthEventFeedback 섹션을 참조하세요.

update-device-status

다음 코드 예시에서는 update-device-status 코드를 사용하는 방법을 보여줍니다.

AWS CLI

디바이스 상태 업데이트

이 예시에서는 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"
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateDeviceStatus 섹션을 참조하세요.

update-group

다음 코드 예시에서는 update-group 코드를 사용하는 방법을 보여줍니다.

AWS CLI

그룹 업데이트

이 예시에서는 MyGroup에 대한 설명과 우선 순위를 업데이트합니다.

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
  }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateGroup 섹션을 참조하세요.

update-identity-provider

다음 코드 예시는 update-identity-provider의 사용 방법을 보여 줍니다.

AWS CLI

사용자 풀 자격 증명 공급자 업데이트

다음 update-identity-provider 예제에서는 요청된 사용자 풀에서 OIDC 공급자 'MyOIDCIdP'를 업데이트합니다.

```
aws cognito-idp update-identity-provider \
    --cli-input-json file://update-identity-provider.json
```

update-identity-provider.json의 콘텐츠:

```
{
   "AttributeMapping": {
```

```
            "email": "idp_email",
            "email_verified": "idp_email_verified",
            "username": "sub"
    },
    "CreationDate": 1.701129701653E9,
    "IdpIdentifiers": [
        "corp",
        "dev"
    ],
    "LastModifiedDate": 1.701129701653E9,
    "ProviderDetails": {
        "attributes_request_method": "GET",
        "attributes_url": "https://example.com/userInfo",
        "attributes_url_add_attributes": "false",
        "authorize_scopes": "openid profile",
        "authorize_url": "https://example.com/authorize",
        "client_id": "idpexampleclient123",
        "client_secret": "idpexamplesecret456",
        "jwks_uri": "https://example.com/.well-known/jwks.json",
        "oidc_issuer": "https://example.com",
        "token_url": "https://example.com/token"
    },
    "ProviderName": "MyOIDCIdP",
    "UserPoolId": "us-west-2_EXAMPLE"
}
```

출력:

```
{
    "IdentityProvider": {
        "AttributeMapping": {
            "email": "idp_email",
            "email_verified": "idp_email_verified",
            "username": "sub"
        },
        "CreationDate": 1701129701.653,
        "IdpIdentifiers": [
            "corp",
            "dev"
        ],
        "LastModifiedDate": 1736444278.211,
        "ProviderDetails": {
            "attributes_request_method": "GET",
```

```
            "attributes_url": "https://example.com/userInfo",
            "attributes_url_add_attributes": "false",
            "authorize_scopes": "openid profile",
            "authorize_url": "https://example.com/authorize",
            "client_id": "idpexampleclient123",
            "client_secret": "idpexamplesecret456",
            "jwks_uri": "https://example.com/.well-known/jwks.json",
            "oidc_issuer": "https://example.com",
            "token_url": "https://example.com/token"
        },
        "ProviderName": "MyOIDCIdP",
        "ProviderType": "OIDC",
        "UserPoolId": "us-west-2_EXAMPLE"
    }
}
```

자세한 내용은 Amazon Cognito 개발자 안내서의 도메인 구성을 참조하시기 바랍니다.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateIdentityProvider를 참조하시기 바랍니다.

update-managed-login-branding

다음 코드 예시는 update-managed-login-branding의 사용 방법을 보여 줍니다.

AWS CLI

관리형 로그인 브랜딩 스타일 업데이트

다음 update-managed-login-branding 예제에서는 요청된 앱 클라이언트 브랜딩 스타일을 업데이트합니다.

```
aws cognito-idp update-managed-login-branding \
    --cli-input-json file://update-managed-login-branding.json
```

update-managed-login-branding.json의 콘텐츠:

```
{
    "Assets": [
        {
            "Bytes":
 "PHN2ZyB3aWR0aD0iMjAwMDAiIGhlaWdodD0iNDAwIiB2aWV3Qm94PSIwIDAgMjAwMDAgNDAwIiBmaWxsPSJub25lIi
+CjxyZWN0IHdpZHRoPSIyMDAwMCIgaGVpZ2h0PSI0MDAiIGZpbGw9InVybCgjcGFpbnQwX2xpbmVhcl8xNzI1OV8yMzY
```

```
+CjxsaW5lYXJHcmFkaWVudCBpZD0icGFpbnQwX2xpbmVhcl8xNzI1OV8yMzY2NzQiIHgxPSItODk0LjI0OSIgeTE9IjE
+Cjwvc3ZnPgo=",
            "Category": "PAGE_FOOTER_BACKGROUND",
            "ColorMode": "DARK",
            "Extension": "SVG"
        }
    ],
    "ManagedLoginBrandingId": "63f30090-6b1f-4278-b885-2bbb81f8e545",
    "Settings": {
        "categories": {
            "auth": {
                "authMethodOrder": [
                    [
                        {
                            "display": "BUTTON",
                            "type": "FEDERATED"
                        },
                        {
                            "display": "INPUT",
                            "type": "USERNAME_PASSWORD"
                        }
                    ]
                ],
                "federation": {
                    "interfaceStyle": "BUTTON_LIST",
                    "order": [
                    ]
                }
            },
            "form": {
                "displayGraphics": true,
                "instructions": {
                    "enabled": false
                },
                "languageSelector": {
                    "enabled": false
                },
                "location": {
                    "horizontal": "CENTER",
                    "vertical": "CENTER"
                },
                "sessionTimerDisplay": "NONE"
            },
            "global": {
```

```
                "colorSchemeMode": "LIGHT",
                "pageFooter": {
                    "enabled": false
                },
                "pageHeader": {
                    "enabled": false
                },
                "spacingDensity": "REGULAR"
            },
            "signUp": {
                "acceptanceElements": [
                    {
                        "enforcement": "NONE",
                        "textKey": "en"
                    }
                ]
            }
        },
        "componentClasses": {
            "buttons": {
                "borderRadius": 8.0
            },
            "divider": {
                "darkMode": {
                    "borderColor": "232b37ff"
                },
                "lightMode": {
                    "borderColor": "ebebf0ff"
                }
            },
            "dropDown": {
                "borderRadius": 8.0,
                "darkMode": {
                    "defaults": {
                        "itemBackgroundColor": "192534ff"
                    },
                    "hover": {
                        "itemBackgroundColor": "081120ff",
                        "itemBorderColor": "5f6b7aff",
                        "itemTextColor": "e9ebedff"
                    },
                    "match": {
                        "itemBackgroundColor": "d1d5dbff",
                        "itemTextColor": "89bdeeff"
```

```
                    }
                },
                "lightMode": {
                    "defaults": {
                        "itemBackgroundColor": "ffffffff"
                    },
                    "hover": {
                        "itemBackgroundColor": "f4f4f4ff",
                        "itemBorderColor": "7d8998ff",
                        "itemTextColor": "000716ff"
                    },
                    "match": {
                        "itemBackgroundColor": "414d5cff",
                        "itemTextColor": "0972d3ff"
                    }
                }
            },
            "focusState": {
                "darkMode": {
                    "borderColor": "539fe5ff"
                },
                "lightMode": {
                    "borderColor": "0972d3ff"
                }
            },
            "idpButtons": {
                "icons": {
                    "enabled": true
                }
            },
            "input": {
                "borderRadius": 8.0,
                "darkMode": {
                    "defaults": {
                        "backgroundColor": "0f1b2aff",
                        "borderColor": "5f6b7aff"
                    },
                    "placeholderColor": "8d99a8ff"
                },
                "lightMode": {
                    "defaults": {
                        "backgroundColor": "ffffffff",
                        "borderColor": "7d8998ff"
                    },
```

```
                    "placeholderColor": "5f6b7aff"
                }
            },
            "inputDescription": {
                "darkMode": {
                    "textColor": "8d99a8ff"
                },
                "lightMode": {
                    "textColor": "5f6b7aff"
                }
            },
            "inputLabel": {
                "darkMode": {
                    "textColor": "d1d5dbff"
                },
                "lightMode": {
                    "textColor": "000716ff"
                }
            },
            "link": {
                "darkMode": {
                    "defaults": {
                        "textColor": "539fe5ff"
                    },
                    "hover": {
                        "textColor": "89bdeeff"
                    }
                },
                "lightMode": {
                    "defaults": {
                        "textColor": "0972d3ff"
                    },
                    "hover": {
                        "textColor": "033160ff"
                    }
                }
            },
            "optionControls": {
                "darkMode": {
                    "defaults": {
                        "backgroundColor": "0f1b2aff",
                        "borderColor": "7d8998ff"
                    },
                    "selected": {
```

```
                        "backgroundColor": "539fe5ff",
                        "foregroundColor": "000716ff"
                    }
                },
                "lightMode": {
                    "defaults": {
                        "backgroundColor": "ffffffff",
                        "borderColor": "7d8998ff"
                    },
                    "selected": {
                        "backgroundColor": "0972d3ff",
                        "foregroundColor": "ffffffff"
                    }
                }
            },
            "statusIndicator": {
                "darkMode": {
                    "error": {
                        "backgroundColor": "1a0000ff",
                        "borderColor": "eb6f6fff",
                        "indicatorColor": "eb6f6fff"
                    },
                    "pending": {
                        "indicatorColor": "AAAAAAAA"
                    },
                    "success": {
                        "backgroundColor": "001a02ff",
                        "borderColor": "29ad32ff",
                        "indicatorColor": "29ad32ff"
                    },
                    "warning": {
                        "backgroundColor": "1d1906ff",
                        "borderColor": "e0ca57ff",
                        "indicatorColor": "e0ca57ff"
                    }
                },
                "lightMode": {
                    "error": {
                        "backgroundColor": "fff7f7ff",
                        "borderColor": "d91515ff",
                        "indicatorColor": "d91515ff"
                    },
                    "pending": {
                        "indicatorColor": "AAAAAAAA"
```

```
                },
                "success": {
                    "backgroundColor": "f2fcf3ff",
                    "borderColor": "037f0cff",
                    "indicatorColor": "037f0cff"
                },
                "warning": {
                    "backgroundColor": "fffce9ff",
                    "borderColor": "8d6605ff",
                    "indicatorColor": "8d6605ff"
                }
            }
        }
    },
    "components": {
        "alert": {
            "borderRadius": 12.0,
            "darkMode": {
                "error": {
                    "backgroundColor": "1a0000ff",
                    "borderColor": "eb6f6fff"
                }
            },
            "lightMode": {
                "error": {
                    "backgroundColor": "fff7f7ff",
                    "borderColor": "d91515ff"
                }
            }
        },
        "favicon": {
            "enabledTypes": [
                "ICO",
                "SVG"
            ]
        },
        "form": {
            "backgroundImage": {
                "enabled": false
            },
            "borderRadius": 8.0,
            "darkMode": {
                "backgroundColor": "0f1b2aff",
                "borderColor": "424650ff"
```

```
                },
                "lightMode": {
                    "backgroundColor": "ffffffff",
                    "borderColor": "c6c6cdff"
                },
                "logo": {
                    "enabled": false,
                    "formInclusion": "IN",
                    "location": "CENTER",
                    "position": "TOP"
                }
            },
            "idpButton": {
                "custom": {
                },
                "standard": {
                    "darkMode": {
                        "active": {
                            "backgroundColor": "354150ff",
                            "borderColor": "89bdeeff",
                            "textColor": "89bdeeff"
                        },
                        "defaults": {
                            "backgroundColor": "0f1b2aff",
                            "borderColor": "c6c6cdff",
                            "textColor": "c6c6cdff"
                        },
                        "hover": {
                            "backgroundColor": "192534ff",
                            "borderColor": "89bdeeff",
                            "textColor": "89bdeeff"
                        }
                    },
                    "lightMode": {
                        "active": {
                            "backgroundColor": "d3e7f9ff",
                            "borderColor": "033160ff",
                            "textColor": "033160ff"
                        },
                        "defaults": {
                            "backgroundColor": "ffffffff",
                            "borderColor": "424650ff",
                            "textColor": "424650ff"
                        },
```

```
                        "hover": {
                            "backgroundColor": "f2f8fdff",
                            "borderColor": "033160ff",
                            "textColor": "033160ff"
                        }
                    }
                }
            },
            "pageBackground": {
                "darkMode": {
                    "color": "0f1b2aff"
                },
                "image": {
                    "enabled": true
                },
                "lightMode": {
                    "color": "ffffffff"
                }
            },
            "pageFooter": {
                "backgroundImage": {
                    "enabled": false
                },
                "darkMode": {
                    "background": {
                        "color": "0f141aff"
                    },
                    "borderColor": "424650ff"
                },
                "lightMode": {
                    "background": {
                        "color": "fafafaff"
                    },
                    "borderColor": "d5dbdbff"
                },
                "logo": {
                    "enabled": false,
                    "location": "START"
                }
            },
            "pageHeader": {
                "backgroundImage": {
                    "enabled": false
                },
```

```
                "darkMode": {
                    "background": {
                        "color": "0f141aff"
                    },
                    "borderColor": "424650ff"
                },
                "lightMode": {
                    "background": {
                        "color": "fafafaff"
                    },
                    "borderColor": "d5dbdbff"
                },
                "logo": {
                    "enabled": false,
                    "location": "START"
                }
            },
            "pageText": {
                "darkMode": {
                    "bodyColor": "b6bec9ff",
                    "descriptionColor": "b6bec9ff",
                    "headingColor": "d1d5dbff"
                },
                "lightMode": {
                    "bodyColor": "414d5cff",
                    "descriptionColor": "414d5cff",
                    "headingColor": "000716ff"
                }
            },
            "phoneNumberSelector": {
                "displayType": "TEXT"
            },
            "primaryButton": {
                "darkMode": {
                    "active": {
                        "backgroundColor": "539fe5ff",
                        "textColor": "000716ff"
                    },
                    "defaults": {
                        "backgroundColor": "539fe5ff",
                        "textColor": "000716ff"
                    },
                    "disabled": {
                        "backgroundColor": "ffffffff",
```

```
                    "borderColor": "ffffffff"
                },
                "hover": {
                    "backgroundColor": "89bdeeff",
                    "textColor": "000716ff"
                }
            },
            "lightMode": {
                "active": {
                    "backgroundColor": "033160ff",
                    "textColor": "ffffffff"
                },
                "defaults": {
                    "backgroundColor": "0972d3ff",
                    "textColor": "ffffffff"
                },
                "disabled": {
                    "backgroundColor": "ffffffff",
                    "borderColor": "ffffffff"
                },
                "hover": {
                    "backgroundColor": "033160ff",
                    "textColor": "ffffffff"
                }
            }
        },
        "secondaryButton": {
            "darkMode": {
                "active": {
                    "backgroundColor": "354150ff",
                    "borderColor": "89bdeeff",
                    "textColor": "89bdeeff"
                },
                "defaults": {
                    "backgroundColor": "0f1b2aff",
                    "borderColor": "539fe5ff",
                    "textColor": "539fe5ff"
                },
                "hover": {
                    "backgroundColor": "192534ff",
                    "borderColor": "89bdeeff",
                    "textColor": "89bdeeff"
                }
            },
```

```
                "lightMode": {
                    "active": {
                        "backgroundColor": "d3e7f9ff",
                        "borderColor": "033160ff",
                        "textColor": "033160ff"
                    },
                    "defaults": {
                        "backgroundColor": "ffffffff",
                        "borderColor": "0972d3ff",
                        "textColor": "0972d3ff"
                    },
                    "hover": {
                        "backgroundColor": "f2f8fdff",
                        "borderColor": "033160ff",
                        "textColor": "033160ff"
                    }
                }
            }
        }
    },
    "UseCognitoProvidedValues": false,
    "UserPoolId": "ca-central-1_EXAMPLE"
}
```

출력:

```
{
    "ManagedLoginBranding": {
        "Assets": [
            {
                "Bytes":
 "PHN2ZyB3aWR0aD0iMjAwMDAiIGhlaWdodD0iNDAwIiB2aWV3Qm94PSIwIDAgMjAwMDAgNDAwIiBmaWxsPSJub25lIi
+CjxyZWN0IHdpZHRoPSIyMDAwMCIgaGVpZ2h0PSI0MDAiIGZpbGw9InVybCgjcGFpbnQwX2xpbmVhcl8xNzI1OV8yMzY
+CjxsaW5lYXJHcmFkaWVudCBpZD0icGFpbnQwX2xpbmVhcl8xNzI1OV8yMzY2NzQiIHgxPSItODk0LjI0OSIgeTE9IjE
+Cjwvc3ZnPgo=",
                "Category": "PAGE_FOOTER_BACKGROUND",
                "ColorMode": "DARK",
                "Extension": "SVG"
            }
        ],
        "CreationDate": 1732138490.642,
        "LastModifiedDate": 1732140420.301,
        "ManagedLoginBrandingId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
```

```
        "Settings": {
            "categories": {
                "auth": {
                    "authMethodOrder": [
                        [
                            {
                                "display": "BUTTON",
                                "type": "FEDERATED"
                            },
                            {
                                "display": "INPUT",
                                "type": "USERNAME_PASSWORD"
                            }
                        ]
                    ],
                    "federation": {
                        "interfaceStyle": "BUTTON_LIST",
                        "order": [
                        ]
                    }
                },
                "form": {
                    "displayGraphics": true,
                    "instructions": {
                        "enabled": false
                    },
                    "languageSelector": {
                        "enabled": false
                    },
                    "location": {
                        "horizontal": "CENTER",
                        "vertical": "CENTER"
                    },
                    "sessionTimerDisplay": "NONE"
                },
                "global": {
                    "colorSchemeMode": "LIGHT",
                    "pageFooter": {
                        "enabled": false
                    },
                    "pageHeader": {
                        "enabled": false
                    },
                    "spacingDensity": "REGULAR"
```

```
                },
                "signUp": {
                    "acceptanceElements": [
                        {
                            "enforcement": "NONE",
                            "textKey": "en"
                        }
                    ]
                }
            },
            "componentClasses": {
                "buttons": {
                    "borderRadius": 8.0
                },
                "divider": {
                    "darkMode": {
                        "borderColor": "232b37ff"
                    },
                    "lightMode": {
                        "borderColor": "ebebf0ff"
                    }
                },
                "dropDown": {
                    "borderRadius": 8.0,
                    "darkMode": {
                        "defaults": {
                            "itemBackgroundColor": "192534ff"
                        },
                        "hover": {
                            "itemBackgroundColor": "081120ff",
                            "itemBorderColor": "5f6b7aff",
                            "itemTextColor": "e9ebedff"
                        },
                        "match": {
                            "itemBackgroundColor": "d1d5dbff",
                            "itemTextColor": "89bdeeff"
                        }
                    },
                    "lightMode": {
                        "defaults": {
                            "itemBackgroundColor": "ffffffff"
                        },
                        "hover": {
                            "itemBackgroundColor": "f4f4f4ff",
```

```
                    "itemBorderColor": "7d8998ff",
                    "itemTextColor": "000716ff"
                },
                "match": {
                    "itemBackgroundColor": "414d5cff",
                    "itemTextColor": "0972d3ff"
                }
            }
        },
        "focusState": {
            "darkMode": {
                "borderColor": "539fe5ff"
            },
            "lightMode": {
                "borderColor": "0972d3ff"
            }
        },
        "idpButtons": {
            "icons": {
                "enabled": true
            }
        },
        "input": {
            "borderRadius": 8.0,
            "darkMode": {
                "defaults": {
                    "backgroundColor": "0f1b2aff",
                    "borderColor": "5f6b7aff"
                },
                "placeholderColor": "8d99a8ff"
            },
            "lightMode": {
                "defaults": {
                    "backgroundColor": "ffffffff",
                    "borderColor": "7d8998ff"
                },
                "placeholderColor": "5f6b7aff"
            }
        },
        "inputDescription": {
            "darkMode": {
                "textColor": "8d99a8ff"
            },
            "lightMode": {
```

```
                    "textColor": "5f6b7aff"
                }
            },
            "inputLabel": {
                "darkMode": {
                    "textColor": "d1d5dbff"
                },
                "lightMode": {
                    "textColor": "000716ff"
                }
            },
            "link": {
                "darkMode": {
                    "defaults": {
                        "textColor": "539fe5ff"
                    },
                    "hover": {
                        "textColor": "89bdeeff"
                    }
                },
                "lightMode": {
                    "defaults": {
                        "textColor": "0972d3ff"
                    },
                    "hover": {
                        "textColor": "033160ff"
                    }
                }
            },
            "optionControls": {
                "darkMode": {
                    "defaults": {
                        "backgroundColor": "0f1b2aff",
                        "borderColor": "7d8998ff"
                    },
                    "selected": {
                        "backgroundColor": "539fe5ff",
                        "foregroundColor": "000716ff"
                    }
                },
                "lightMode": {
                    "defaults": {
                        "backgroundColor": "ffffffff",
                        "borderColor": "7d8998ff"
```

```
                },
                "selected": {
                    "backgroundColor": "0972d3ff",
                    "foregroundColor": "ffffffff"
                }
            }
        },
        "statusIndicator": {
            "darkMode": {
                "error": {
                    "backgroundColor": "1a0000ff",
                    "borderColor": "eb6f6fff",
                    "indicatorColor": "eb6f6fff"
                },
                "pending": {
                    "indicatorColor": "AAAAAAAA"
                },
                "success": {
                    "backgroundColor": "001a02ff",
                    "borderColor": "29ad32ff",
                    "indicatorColor": "29ad32ff"
                },
                "warning": {
                    "backgroundColor": "1d1906ff",
                    "borderColor": "e0ca57ff",
                    "indicatorColor": "e0ca57ff"
                }
            },
            "lightMode": {
                "error": {
                    "backgroundColor": "fff7f7ff",
                    "borderColor": "d91515ff",
                    "indicatorColor": "d91515ff"
                },
                "pending": {
                    "indicatorColor": "AAAAAAAA"
                },
                "success": {
                    "backgroundColor": "f2fcf3ff",
                    "borderColor": "037f0cff",
                    "indicatorColor": "037f0cff"
                },
                "warning": {
                    "backgroundColor": "fffce9ff",
```

```
                        "borderColor": "8d6605ff",
                        "indicatorColor": "8d6605ff"
                    }
                }
            }
        },
        "components": {
            "alert": {
                "borderRadius": 12.0,
                "darkMode": {
                    "error": {
                        "backgroundColor": "1a0000ff",
                        "borderColor": "eb6f6fff"
                    }
                },
                "lightMode": {
                    "error": {
                        "backgroundColor": "fff7f7ff",
                        "borderColor": "d91515ff"
                    }
                }
            },
            "favicon": {
                "enabledTypes": [
                    "ICO",
                    "SVG"
                ]
            },
            "form": {
                "backgroundImage": {
                    "enabled": false
                },
                "borderRadius": 8.0,
                "darkMode": {
                    "backgroundColor": "0f1b2aff",
                    "borderColor": "424650ff"
                },
                "lightMode": {
                    "backgroundColor": "ffffffff",
                    "borderColor": "c6c6cdff"
                },
                "logo": {
                    "enabled": false,
                    "formInclusion": "IN",
```

```
                    "location": "CENTER",
                    "position": "TOP"
                }
            },
            "idpButton": {
                "custom": {
                },
                "standard": {
                    "darkMode": {
                        "active": {
                            "backgroundColor": "354150ff",
                            "borderColor": "89bdeeff",
                            "textColor": "89bdeeff"
                        },
                        "defaults": {
                            "backgroundColor": "0f1b2aff",
                            "borderColor": "c6c6cdff",
                            "textColor": "c6c6cdff"
                        },
                        "hover": {
                            "backgroundColor": "192534ff",
                            "borderColor": "89bdeeff",
                            "textColor": "89bdeeff"
                        }
                    },
                    "lightMode": {
                        "active": {
                            "backgroundColor": "d3e7f9ff",
                            "borderColor": "033160ff",
                            "textColor": "033160ff"
                        },
                        "defaults": {
                            "backgroundColor": "ffffffff",
                            "borderColor": "424650ff",
                            "textColor": "424650ff"
                        },
                        "hover": {
                            "backgroundColor": "f2f8fdff",
                            "borderColor": "033160ff",
                            "textColor": "033160ff"
                        }
                    }
                }
            },
```

```
                "pageBackground": {
                    "darkMode": {
                        "color": "0f1b2aff"
                    },
                    "image": {
                        "enabled": true
                    },
                    "lightMode": {
                        "color": "ffffffff"
                    }
                },
                "pageFooter": {
                    "backgroundImage": {
                        "enabled": false
                    },
                    "darkMode": {
                        "background": {
                            "color": "0f141aff"
                        },
                        "borderColor": "424650ff"
                    },
                    "lightMode": {
                        "background": {
                            "color": "fafafaff"
                        },
                        "borderColor": "d5dbdbff"
                    },
                    "logo": {
                        "enabled": false,
                        "location": "START"
                    }
                },
                "pageHeader": {
                    "backgroundImage": {
                        "enabled": false
                    },
                    "darkMode": {
                        "background": {
                            "color": "0f141aff"
                        },
                        "borderColor": "424650ff"
                    },
                    "lightMode": {
                        "background": {
```

```
                        "color": "fafafaff"
                    },
                    "borderColor": "d5dbdbff"
                },
                "logo": {
                    "enabled": false,
                    "location": "START"
                }
            },
            "pageText": {
                "darkMode": {
                    "bodyColor": "b6bec9ff",
                    "descriptionColor": "b6bec9ff",
                    "headingColor": "d1d5dbff"
                },
                "lightMode": {
                    "bodyColor": "414d5cff",
                    "descriptionColor": "414d5cff",
                    "headingColor": "000716ff"
                }
            },
            "phoneNumberSelector": {
                "displayType": "TEXT"
            },
            "primaryButton": {
                "darkMode": {
                    "active": {
                        "backgroundColor": "539fe5ff",
                        "textColor": "000716ff"
                    },
                    "defaults": {
                        "backgroundColor": "539fe5ff",
                        "textColor": "000716ff"
                    },
                    "disabled": {
                        "backgroundColor": "ffffffff",
                        "borderColor": "ffffffff"
                    },
                    "hover": {
                        "backgroundColor": "89bdeeff",
                        "textColor": "000716ff"
                    }
                },
                "lightMode": {
```

```
                    "active": {
                        "backgroundColor": "033160ff",
                        "textColor": "ffffffff"
                    },
                    "defaults": {
                        "backgroundColor": "0972d3ff",
                        "textColor": "ffffffff"
                    },
                    "disabled": {
                        "backgroundColor": "ffffffff",
                        "borderColor": "ffffffff"
                    },
                    "hover": {
                        "backgroundColor": "033160ff",
                        "textColor": "ffffffff"
                    }
                }
            },
            "secondaryButton": {
                "darkMode": {
                    "active": {
                        "backgroundColor": "354150ff",
                        "borderColor": "89bdeeff",
                        "textColor": "89bdeeff"
                    },
                    "defaults": {
                        "backgroundColor": "0f1b2aff",
                        "borderColor": "539fe5ff",
                        "textColor": "539fe5ff"
                    },
                    "hover": {
                        "backgroundColor": "192534ff",
                        "borderColor": "89bdeeff",
                        "textColor": "89bdeeff"
                    }
                },
                "lightMode": {
                    "active": {
                        "backgroundColor": "d3e7f9ff",
                        "borderColor": "033160ff",
                        "textColor": "033160ff"
                    },
                    "defaults": {
                        "backgroundColor": "ffffffff",
```

```
                                "borderColor": "0972d3ff",
                                "textColor": "0972d3ff"
                            },
                            "hover": {
                                "backgroundColor": "f2f8fdff",
                                "borderColor": "033160ff",
                                "textColor": "033160ff"
                            }
                        }
                    }
                }
            },
            "UseCognitoProvidedValues": false,
            "UserPoolId": "ca-central-1_EXAMPLE"
        }
    }
```

자세한 내용은 Amazon Cognito 개발자 안내서의 Apply branding to managed login pages를 참조하시기 바랍니다.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateManagedLoginBranding을 참조하시기 바랍니다.

update-resource-server

다음 코드 예시는 update-resource-server의 사용 방법을 보여 줍니다.

AWS CLI

리소스 서버 업데이트

이 예시에서는 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"
            }
        ]
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateResourceServer 섹션을 참조하세요.

update-user-attributes

다음 코드 예시에서는 update-user-attributes 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용자 속성을 업데이트하는 방법

이 예시에서는 사용자 속성 'nickname'을 업데이트합니다.

명령:

```
aws cognito-idp update-user-attributes --access-token ACCESS_TOKEN --user-attributes
 Name="nickname",Value="Dan"
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateUserAttributes 섹션을 참조하세요.

update-user-pool-client

다음 코드 예시는 update-user-pool-client의 사용 방법을 보여 줍니다.

AWS CLI

앱 클라이언트 업데이트

다음 update-user-pool-client 예제에서는 요청된 앱 클라이언트의 구성을 업데이트합니다.

```
aws cognito-idp update-user-pool-client \
    --user-pool-id us-west-2_EXAMPLE \
    --client-id 1example23456789 \
    --client-name my-test-app \
    --refresh-token-validity 30 \
    --access-token-validity 60 \
    --id-token-validity 60 \
    --token-validity-units AccessToken=minutes,IdToken=minutes,RefreshToken=days \
    --read-
attributes "address" "birthdate" "email" "email_verified" "family_name" "gender" "locale" "m
 \
    --write-
attributes "address" "birthdate" "email" "family_name" "gender" "locale" "middle_name" "name
 \
    --explicit-auth-
flows "ALLOW_ADMIN_USER_PASSWORD_AUTH" "ALLOW_CUSTOM_AUTH" "ALLOW_REFRESH_TOKEN_AUTH" "ALLOW
 \
    --supported-identity-providers "MySAML" "COGNITO" "Google" \
    --callback-urls "https://www.example.com" "https://app2.example.com" \
    --logout-urls "https://auth.example.com/login?
client_id=1example23456789&response_type=code&redirect_uri=https%3A%2F
%2Fwww.example.com" "https://example.com/logout" \
    --default-redirect-uri "https://www.example.com" \
    --allowed-o-auth-flows "code" "implicit" \
    --allowed-o-auth-scopes "openid" "profile" "aws.cognito.signin.user.admin" \
    --allowed-o-auth-flows-user-pool-client \
    --prevent-user-existence-errors ENABLED \
    --enable-token-revocation \
    --no-enable-propagate-additional-user-context-data \
    --auth-session-validity 3
```

출력:

```
{
    "UserPoolClient": {
        "UserPoolId": "us-west-2_EXAMPLE",
        "ClientName": "my-test-app",
        "ClientId": "1example23456789",
        "LastModifiedDate": "2025-01-31T14:40:12.498000-08:00",
        "CreationDate": "2023-09-13T16:26:34.408000-07:00",
        "RefreshTokenValidity": 30,
        "AccessTokenValidity": 60,
        "IdTokenValidity": 60,
```

```
        "TokenValidityUnits": {
            "AccessToken": "minutes",
            "IdToken": "minutes",
            "RefreshToken": "days"
        },
        "ReadAttributes": [
            "website",
            "zoneinfo",
            "address",
            "birthdate",
            "email_verified",
            "gender",
            "profile",
            "phone_number_verified",
            "preferred_username",
            "locale",
            "middle_name",
            "picture",
            "updated_at",
            "name",
            "nickname",
            "phone_number",
            "family_name",
            "email"
        ],
        "WriteAttributes": [
            "website",
            "zoneinfo",
            "address",
            "birthdate",
            "gender",
            "profile",
            "preferred_username",
            "locale",
            "middle_name",
            "picture",
            "updated_at",
            "name",
            "nickname",
            "phone_number",
            "family_name",
            "email"
        ],
        "ExplicitAuthFlows": [
```

```
            "ALLOW_CUSTOM_AUTH",
            "ALLOW_USER_PASSWORD_AUTH",
            "ALLOW_ADMIN_USER_PASSWORD_AUTH",
            "ALLOW_USER_SRP_AUTH",
            "ALLOW_REFRESH_TOKEN_AUTH"
        ],
        "SupportedIdentityProviders": [
            "Google",
            "COGNITO",
            "MySAML"
        ],
        "CallbackURLs": [
            "https://www.example.com",
            "https://app2.example.com"
        ],
        "LogoutURLs": [
            "https://example.com/logout",
            "https://auth.example.com/login?
client_id=1example23456789&response_type=code&redirect_uri=https%3A%2F
%2Fwww.example.com"
        ],
        "DefaultRedirectURI": "https://www.example.com",
        "AllowedOAuthFlows": [
            "implicit",
            "code"
        ],
        "AllowedOAuthScopes": [
            "aws.cognito.signin.user.admin",
            "openid",
            "profile"
        ],
        "AllowedOAuthFlowsUserPoolClient": true,
        "PreventUserExistenceErrors": "ENABLED",
        "EnableTokenRevocation": true,
        "EnablePropagateAdditionalUserContextData": false,
        "AuthSessionValidity": 3
    }
}
```

자세한 내용은 Amazon Cognito 개발자 안내서의 앱 클라이언트를 사용한 애플리케이션별 설정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateUserPoolClient 섹션을 참조하세요.

update-user-pool-domain

다음 코드 예시는 update-user-pool-domain의 사용 방법을 보여 줍니다.

AWS CLI

사용자 지정 도메인 이름 업데이트

다음 update-user-pool-domain 예제에서는 요청된 사용자 풀의 사용자 지정 도메인에 대한 브랜딩 버전과 인증을 구성합니다.

```
aws cognito-idp update-user-pool-domain \
    --user-pool-id ca-central-1_EXAMPLE \
    --domain auth.example.com \
    --managed-login-version 2 \
    --custom-domain-config CertificateArn=arn:aws:acm:us-
east-1:123456789012:certificate/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

출력:

```
{
    "CloudFrontDomain": "example.cloudfront.net",
    "ManagedLoginVersion": 2
}
```

자세한 내용은 Amazon Cognito 개발자 안내서의 Managed login 및 Configuring a domain을 참조하시기 바랍니다.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateUserPoolDomain을 참조하시기 바랍니다.

update-user-pool

다음 코드 예시는 update-user-pool의 사용 방법을 보여 줍니다.

AWS CLI

사용자 풀 업데이트

다음 update-user-pool 예시에서는 사용 가능한 각 구성 옵션에 대한 예시 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 Updating user pool configuration을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateUserPool 섹션을 참조하세요.

verify-software-token

다음 코드 예시는 verify-software-token의 사용 방법을 보여 줍니다.

AWS CLI

TOTP 인증자의 등록 확인

다음 verify-software-token 예제에서는 현재 사용자의 TOTP 등록을 완료합니다.

```
aws cognito-idp verify-software-token \
    --access-token eyJra456defEXAMPLE \
    --user-code 123456
```

출력:

```
{
    "Status": "SUCCESS"
}
```

자세한 내용은 Amazon Cognito 개발자 안내서의 Adding MFA to a user pool을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 VerifySoftwareToken을 참조하시기 바랍니다.

verify-user-attribute

다음 코드 예시는 verify-user-attribute의 사용 방법을 보여 줍니다.

AWS CLI

속성 변경 확인

다음 verify-user-attribute 예제에서는 현재 사용자의 이메일 속성에 대한 변경 사항을 확인합니다.

```
aws cognito-idp verify-user-attribute \
    --access-token eyJra456defEXAMPLE \
    --attribute-name email \
```

```
    --code 123456
```

자세한 내용을 알아보려면 Amazon Cognito 개발자 안내서의 Configuring email or phone verification을 참조하시기 바랍니다.

- API 세부 정보는 AWS CLI 명령 참조의 VerifyUserAttribute를 참조하시기 바랍니다.

AWS CLI를 사용한 Amazon Comprehend 예제

다음 코드 예제는 Amazon Comprehend와 함께 AWS Command Line Interface를 사용하여 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

batch-detect-dominant-language

다음 코드 예시는 `batch-detect-dominant-language`의 사용 방법을 보여 줍니다.

AWS CLI

여러 입력 텍스트의 주 언어를 감지하려면

다음 `batch-detect-dominant-language` 예시에서는 여러 입력 텍스트를 분석하고 각각의 주 언어를 반환합니다. 각 예측에 대한 사전 훈련된 모델의 신뢰도 점수도 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 안내서의 주로 사용되는 언어 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 BatchDetectDominantLanguage 섹션을 참조하세요.

batch-detect-entities

다음 코드 예시는 batch-detect-entities의 사용 방법을 보여 줍니다.

AWS CLI

여러 입력 텍스트에서 엔터티를 감지하려면

다음 batch-detect-entities 예시에서는 입력 텍스트를 분석하고 각각에 대해 이름이 지정된 엔터티를 반환합니다. 각 예측에 대한 사전 훈련된 모델의 신뢰도 점수도 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 안내서의 엔티티 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 BatchDetectEntities 섹션을 참조하세요.

batch-detect-key-phrases

다음 코드 예시는 batch-detect-key-phrases의 사용 방법을 보여 줍니다.

AWS CLI

여러 텍스트 입력의 키 구문을 감지하려면

다음 batch-detect-key-phrases 예시에서는 여러 입력 텍스트를 분석하여 각각의 핵심 명사 구문을 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 안내서의 핵심 문구 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 BatchDetectKeyPhrases 섹션을 참조하세요.

batch-detect-sentiment

다음 코드 예시는 batch-detect-sentiment의 사용 방법을 보여 줍니다.

AWS CLI

여러 입력 텍스트의 주된 감정을 감지하려면

다음 batch-detect-sentiment 예시에서는 여러 입력 텍스트를 분석하고 주된 감정 (POSITIVE, NEUTRAL, MIXED 또는 NEGATIVE)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 Sentiment 섹션을 참조하세요

- API 세부 정보는 AWS CLI 명령 참조의 BatchDetectSentiment 섹션을 참조하세요.

batch-detect-syntax

다음 코드 예시는 `batch-detect-syntax`의 사용 방법을 보여 줍니다.

AWS CLI

여러 입력 텍스트에서 단어의 구문과 품사를 검사하려면

다음 `batch-detect-syntax` 예시에서는 여러 입력 텍스트의 구문을 분석하고 다양한 품사를 반환합니다. 각 예측에 대한 사전 훈련된 모델의 신뢰도 점수도 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 안내서의 구문 분석 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 BatchDetectSyntax 섹션을 참조하세요.

batch-detect-targeted-sentiment

다음 코드 예시는 batch-detect-targeted-sentiment의 사용 방법을 보여 줍니다.

AWS CLI

여러 입력 텍스트에 대해 감정과 이름이 지정된 각 엔터티를 감지하려면

다음 batch-detect-targeted-sentiment 예시에서는 여러 입력 텍스트를 분석하여 각 엔터티에 연결된 주된 감정과 함께 이름이 지정된 엔터티를 반환합니다. 각 예측에 대한 사전 훈련된 모델의 신뢰도 점수도 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 Targeted Sentiment 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 BatchDetectTargetedSentiment 섹션을 참조하세요.

classify-document

다음 코드 예시는 classify-document의 사용 방법을 보여 줍니다.

AWS CLI

모델별 엔드포인트로 문서를 분류하려면

다음 classify-document 예시에서는 사용자 지정 모델의 엔드포인트로 문서를 분류합니다. 이 예시의 모델은 스팸 또는 스팸이 아닌 메시지, 즉 '햄'으로 분류된 문자 메시지가 포함된 데이터세트에 대해 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": 0.000140012052725069220
        }
    ]
}
```

자세한 내용은 Amazon Comprehend 개발자 안내서의 [사용자 지정 분류] 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ClassifyDocument] 섹션을 참조하세요.

contains-pii-entities

다음 코드 예시는 `contains-pii-entities`의 사용 방법을 보여 줍니다.

AWS CLI

입력 텍스트에서 PII 정보의 존재를 분석하려면

다음 `contains-pii-entities` 예시에서는 입력 텍스트에 개인 식별 정보(PII)가 있는지 분석하여 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) 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ContainsPiiEntities 섹션을 참조하세요.

create-dataset

다음 코드 예시는 `create-dataset`의 사용 방법을 보여 줍니다.

AWS CLI

플라이휠 데이터세트를 만들려면

다음 `create-dataset` 예시에서는 플라이휠에 대한 데이터세트를 생성합니다. 이 데이터세트는 `--dataset-type` 태그에 지정된 대로 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.json의 콘텐츠:

```
{
    "DataFormat": "COMPREHEND_CSV",
    "DocumentClassifierInputDataConfig": {
        "S3Uri": "s3://amzn-s3-demo-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 Overview 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateDataset 섹션을 참조하세요.

create-document-classifier

다음 코드 예시는 create-document-classifier의 사용 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.csv는 --input-data-config 태그에 있습니다. training.csv는 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://amzn-s3-demo-bucket/" \
```

```
    --language-code en
```

출력:

```
{
    "DocumentClassifierArn": "arn:aws:comprehend:us-west-2:111122223333:document-classifier/example-classifier"
}
```

자세한 내용은 Amazon Comprehend 개발자 안내서의 사용자 지정 분류 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateDocumentClassifier 섹션을 참조하세요.

`create-endpoint`

다음 코드 예시는 `create-endpoint`의 사용 방법을 보여 줍니다.

AWS CLI

사용자 지정 모델의 엔드포인트를 생성하려면

다음 `create-endpoint` 예시에서는 이전에 훈련된 사용자 지정 모델의 동기식 추론을 위한 엔드포인트를 만드는 예시입니다.

```
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 안내서의 Amazon Comprehend 엔드포인트 관리 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateEndpoint 섹션을 참조하세요.

create-entity-recognizer

다음 코드 예시는 create-entity-recognizer의 사용 방법을 보여 줍니다.

AWS CLI

사용자 지정 엔터티 인식기를 생성하려면

다음 create-entity-recognizer 예시에서는 사용자 지정 엔터티 인식기 모델에 대한 훈련 프로세스를 시작합니다. 이 예시에서는 훈련 문서인 raw_text.csv 및 CSV 엔터티 목록 entity_list.csv가 포함된 CSV 파일을 사용하여 모델을 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://amzn-s3-
demo-bucket/trainingdata/raw_text.csv},EntityList={S3Uri=s3://amzn-s3-demo-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 Custom entity recognition 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateEntityRecognizer 섹션을 참조하세요.

create-flywheel

다음 코드 예시는 create-flywheel의 사용 방법을 보여 줍니다.

AWS CLI

플라이휠을 만들려면

다음 create-flywheel 예시에서는 문서 분류 또는 엔터티 인식 모델의 지속적인 훈련을 오케스트레이션하는 플라이휠을 생성합니다. 이 예시의 플라이휠은 --active-model-arn 태그에 지정된 기존 훈련된 모델을 관리하기 위해 만들어졌습니다. 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://amzn-s3-demo-bucket"
```

출력:

```
{
    "FlywheelArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/example-
flywheel"
}
```

자세한 내용은 Amazon Comprehend 개발자 안내서의 Flywheel Overview 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateFlywheel 섹션을 참조하세요.

delete-document-classifier

다음 코드 예시는 delete-document-classifier의 사용 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 안내서의 Amazon Comprehend 엔드포인트 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteDocumentClassifier를 참조하세요.

delete-endpoint

다음 코드 예시는 delete-endpoint의 사용 방법을 보여 줍니다.

AWS CLI

사용자 지정 모델의 엔드포인트를 삭제하려면

다음 delete-endpoint 예시에서는 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 안내서의 Amazon Comprehend 엔드포인트 관리 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteEndpoint 섹션을 참조하세요.

delete-entity-recognizer

다음 코드 예시는 delete-entity-recognizer의 사용 방법을 보여 줍니다.

AWS CLI

사용자 엔터티 객체 인식기 모델을 삭제하려면

다음 delete-entity-recognizer 예시에서는 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 안내서의 Amazon Comprehend 엔드포인트 관리 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteEntityRecognizer 섹션을 참조하세요.

delete-flywheel

다음 코드 예시는 delete-flywheel의 사용 방법을 보여 줍니다.

AWS CLI

플라이휠을 삭제하려면

다음 delete-flywheel 예시에서는 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 overview 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteFlywheel 섹션을 참조하세요.

delete-resource-policy

다음 코드 예시는 delete-resource-policy의 사용 방법을 보여 줍니다.

AWS CLI

리소스 기반 정책을 삭제하려면

다음 delete-resource-policy 예시에서는 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 Copying custom models between AWS accounts 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조에서 DeleteResourcePolicy 섹션을 참조하세요.

describe-dataset

다음 코드 예시는 describe-dataset의 사용 방법을 보여 줍니다.

AWS CLI

플라이휠 데이터세트를 설명하려면

다음 describe-dataset 예시에서는 플라이휠 데이터세트의 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://amzn-s3-demo-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 Flywheel Overview 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDataset 섹션을 참조하세요.

describe-document-classification-job

다음 코드 예시는 describe-document-classification-job의 사용 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://amzn-s3-demo-bucket/jobdata/",
            "InputFormat": "ONE_DOC_PER_LINE"
        },
        "OutputDataConfig": {
            "S3Uri": "s3://amzn-s3-demo-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 안내서의 사용자 지정 분류를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조에서 DescribeDocumentClassificationJob를 참조하세요.

describe-document-classifier

다음 코드 예시는 describe-document-classifier의 사용 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://amzn-s3-demo-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 안내서의 사용자 지정 모델 생성 및 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조에서 DescribeDocumentClassifier를 참조하세요.

describe-dominant-language-detection-job

다음 코드 예시는 `describe-dominant-language-detection-job`의 사용 방법을 보여 줍니다.

AWS CLI

주된 언어 감지 작업을 설명하려면.

다음 `describe-dominant-language-detection-job` 예시에서는 비동기 주된 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://amzn-s3-demo-bucket",
            "InputFormat": "ONE_DOC_PER_LINE"
        },
        "OutputDataConfig": {
            "S3Uri": "s3://amzn-s3-demo-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 안내서의 Amazon Comprehend 통찰력 비동기 분석을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDominantLanguageDetectionJob 섹션을 참조하세요.

describe-endpoint

다음 코드 예시는 describe-endpoint의 사용 방법을 보여 줍니다.

AWS CLI

특정 엔드포인트를 설명하려면

다음 describe-endpoint 예시에서는 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 안내서의 Amazon Comprehend 엔드포인트 관리 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeEndpoint를 참조하세요.

describe-entities-detection-job

다음 코드 예시는 describe-entities-detection-job의 사용 방법을 보여 줍니다.

AWS CLI

엔터티 감지 작업을 설명하려면

다음 describe-entities-detection-job 예시에서는 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://amzn-s3-demo-bucket/AsyncBatchJobs/",
            "InputFormat": "ONE_DOC_PER_LINE"
        },
        "OutputDataConfig": {
            "S3Uri": "s3://amzn-s3-demo-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 안내서의 Amazon Comprehend 통찰력 비동기 분석을 참조하세요.

- API 세부 정보는 AWS CLI명령 참조의 DescribeEntitiesDetectionJob 섹션을 참조하세요.

describe-entity-recognizer

다음 코드 예시는 describe-entity-recognizer의 사용 방법을 보여 줍니다.

AWS CLI

엔터티 인식기를 설명하려면

다음 describe-entity-recognizer 예시에서는 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://amzn-s3-demo-bucket/trainingdata/dataset/",
                "InputFormat": "ONE_DOC_PER_LINE"
```

```
            },
            "EntityList": {
                "S3Uri": "s3://amzn-s3-demo-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 Custom entity recognition 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeEntityRecognizer 섹션을 참조하세요.

describe-events-detection-job

다음 코드 예시는 describe-events-detection-job의 사용 방법을 보여 줍니다.

AWS CLI

이벤트 감지 작업을 설명하려면

다음 describe-events-detection-job 예시에서는 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://amzn-s3-demo-bucket/EventsData",
            "InputFormat": "ONE_DOC_PER_LINE"
        },
        "OutputDataConfig": {
            "S3Uri": "s3://amzn-s3-demo-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 안내서의 Amazon Comprehend 통찰력 비동기 분석을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeEventsDetectionJob 섹션을 참조하세요.

describe-flywheel-iteration

다음 코드 예시는 describe-flywheel-iteration의 사용 방법을 보여 줍니다.

AWS CLI

플라이휠 반복을 설명하려면

다음 describe-flywheel-iteration 예시에서는 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://amzn-s3-demo-destination-bucket/
flywheel-entity/schemaVersion=1/20230616T200543Z/evaluation/20230616T211026Z/"
    }
}
```

자세한 내용은 Amazon Comprehend 개발자 안내서의 Flywheel overview 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeFlywheelIteration 섹션을 참조하세요.

describe-flywheel

다음 코드 예시는 describe-flywheel의 사용 방법을 보여 줍니다.

AWS CLI

플라이휠을 설명하려면

다음 describe-flywheel 예시에서는 플라이휠의 속성을 가져옵니다. 이 예시에서는 플라이휠과 연결된 모델이 문서를 스팸 또는 스팸이 아닌 문서, 즉 '햄'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://amzn-s3-demo-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 Flywheel Overview 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeFlywheel 섹션을 참조하세요.

describe-key-phrases-detection-job

다음 코드 예시는 describe-key-phrases-detection-job의 사용 방법을 보여 줍니다.

AWS CLI

핵심 문구 감지 작업을 설명하려면

다음 describe-key-phrases-detection-job 예시에서는 비동기 핵심 문구 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 안내서의 Amazon Comprehend 통찰력 비동기 분석을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeKeyPhrasesDetectionJob 섹션을 참조하세요.

describe-pii-entities-detection-job

다음 코드 예시는 describe-pii-entities-detection-job의 사용 방법을 보여 줍니다.

AWS CLI

PII 엔터티 감지 작업을 설명하려면

다음 describe-pii-entities-detection-job 예시에서는 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://amzn-s3-demo-bucket/AsyncBatchJobs/",
            "InputFormat": "ONE_DOC_PER_LINE"
        },
        "OutputDataConfig": {
            "S3Uri": "s3://amzn-s3-demo-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 안내서의 Amazon Comprehend 통찰력 비동기 분석을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribePiiEntitiesDetectionJob 섹션을 참조하세요.

describe-resource-policy

다음 코드 예시는 `describe-resource-policy`의 사용 방법을 보여 줍니다.

AWS CLI

모델에 연결된 리소스 정책을 설명하려면

다음 `describe-resource-policy` 예시에서는 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 Copying custom models between AWS accounts 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeResourcePolicy 섹션을 참조하세요.

describe-sentiment-detection-job

다음 코드 예시는 `describe-sentiment-detection-job`의 사용 방법을 보여 줍니다.

AWS CLI

감정 감지 작업을 설명하려면

다음 `describe-sentiment-detection-job` 예시에서는 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://amzn-s3-demo-bucket/MovieData",
            "InputFormat": "ONE_DOC_PER_LINE"
        },
        "OutputDataConfig": {
            "S3Uri": "s3://amzn-s3-demo-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 안내서의 Amazon Comprehend 통찰력 비동기 분석을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeSentimentDetectionJob 섹션을 참조하세요.

describe-targeted-sentiment-detection-job

다음 코드 예시는 describe-targeted-sentiment-detection-job의 사용 방법을 보여 줍니다.

AWS CLI

대상 감정 감지 작업을 설명하려면

다음 describe-targeted-sentiment-detection-job 예시에서는 비동기 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://amzn-s3-demo-bucket/MovieData",
            "InputFormat": "ONE_DOC_PER_LINE"
        },
        "OutputDataConfig": {
            "S3Uri": "s3://amzn-s3-demo-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 안내서의 Amazon Comprehend 통찰력 비동기 분석을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeTargetedSentimentDetectionJob 섹션을 참조하세요.

describe-topics-detection-job

다음 코드 예시는 describe-topics-detection-job의 사용 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://amzn-s3-demo-bucket",
            "InputFormat": "ONE_DOC_PER_LINE"
        },
        "OutputDataConfig": {
            "S3Uri": "s3://amzn-s3-demo-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 안내서의 Amazon Comprehend 통찰력 비동기 분석을 참조하세요.

- API 세부 정보는 AWS CLI명령 참조에서 DescribeTopicsDetectionJob을 참조하세요.

detect-dominant-language

다음 코드 예시는 detect-dominant-language의 사용 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 안내서의 주로 사용되는 언어를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DetectDominantLanguage를 참조하세요.

detect-entities

다음 코드 예시는 detect-entities의 사용 방법을 보여 줍니다.

AWS CLI

입력 텍스트에서 이름이 지정된 엔터티를 감지하려면

다음 detect-entities 예제에서는 입력 텍스트를 분석하고 이름이 지정된 엔터티를 반환합니다. 각 예측에 대한 사전 훈련된 모델의 신뢰도 점수도 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 안내서의 엔티티를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DetectEntities를 참조하세요.

detect-key-phrases

다음 코드 예시는 detect-key-phrases의 사용 방법을 보여 줍니다.

AWS CLI

입력 텍스트에서 핵심 문구 탐지

다음 detect-key-phrases 예제에서는 입력 텍스트를 분석하고 핵심 명사구를 식별합니다. 각 예측에 대한 사전 훈련된 모델의 신뢰도 점수도 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 안내서의 핵심 문구를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DetectKeyPhrases를 참조하세요.

`detect-pii-entities`

다음 코드 예시는 `detect-pii-entities`의 사용 방법을 보여 줍니다.

AWS CLI

입력 텍스트에서 pii 엔티티 탐지

다음 `detect-pii-entities` 예제는 입력 텍스트를 분석하고 개인 식별 정보(PII)가 포함된 엔티티를 식별합니다. 각 예측에 대한 사전 훈련된 모델의 신뢰도 점수도 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.

- API 세부 정보는 AWS CLI 명령 참조의 DetectPiiEntities를 참조하세요.

`detect-sentiment`

다음 코드 예시는 `detect-sentiment`의 사용 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 안내서의 Sentiment 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DetectSentiment를 참조하세요.

detect-syntax

다음 코드 예시는 detect-syntax의 사용 방법을 보여 줍니다.

AWS CLI

입력 텍스트에서 품사 탐지

다음 detect-syntax 예제에서는 입력 텍스트의 구문을 분석하고 품사의 여러 부분을 반환합니다. 각 예측에 대한 사전 훈련된 모델의 신뢰도 점수도 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 안내서의 구문 분석을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DetectSyntax를 참조하세요.

`detect-targeted-sentiment`

다음 코드 예시는 `detect-targeted-sentiment`의 사용 방법을 보여 줍니다.

AWS CLI

입력 텍스트에서 이름이 지정된 엔터티의 대상 감정을 감지하려면

다음 `detect-targeted-sentiment` 예시에서는 입력 텍스트를 분석하여 각 엔티티와 연관된 대상 감정과 함께 이름이 지정된 엔터티를 반환합니다. 각 예측에 대한 사전 훈련된 모델의 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 Targeted Sentiment 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DetectTargetedSentiment 섹션을 참조하세요.

`import-model`

다음 코드 예시는 `import-model`의 사용 방법을 보여 줍니다.

AWS CLI

모델을 가져오려면

다음 `import-model` 예시에서는 다른 AWS 계정에서 모델을 가져옵니다. 444455556666 계정의 문서 분류기 모델에는 111122223333 계정이 모델을 가져올 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 Copying custom models between AWS accounts 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ImportModel 섹션을 참조하세요.

list-datasets

다음 코드 예시는 `list-datasets`의 사용 방법을 보여 줍니다.

AWS CLI

모든 플라이휠 데이터세트를 나열하려면

다음 `list-datasets` 예시에서는 플라이휠과 연결된 모든 데이터세트를 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://amzn-s3-demo-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://amzn-s3-demo-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 Flywheel Overview 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListDatasets 섹션을 참조하세요.

`list-document-classification-jobs`

다음 코드 예시는 `list-document-classification-jobs`의 사용 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://amzn-s3-demo-bucket/jobdata/",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-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://amzn-s3-demo-bucket/jobdata/",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-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 안내서의 사용자 지정 분류를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListDocumentClassificationJobs를 참조하세요.

list-document-classifier-summaries

다음 코드 예시는 `list-document-classifier-summaries`의 사용 방법을 보여 줍니다.

AWS CLI

생성된 모든 문서 분류기의 요약을 나열하려면

다음 `list-document-classifier-summaries` 예시에서는 모든 문서 분류기 작업이 나열되어 있습니다.

```
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 안내서의 사용자 지정 모델 생성 및 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListDocumentClassifierSummaries 섹션을 참조하세요.

`list-document-classifiers`

다음 코드 예시는 `list-document-classifiers`의 사용 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://amzn-s3-demo-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://amzn-s3-demo-bucket/trainingdata"
            },
            "OutputDataConfig": {},
            "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-testorle",
            "Mode": "MULTI_CLASS"
        }
    ]
}
```

자세한 내용은 Amazon Comprehend 개발자 안내서의 사용자 지정 모델 생성 및 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListDocumentClassifiers를 참조하세요.

`list-dominant-language-detection-jobs`

다음 코드 예시는 `list-dominant-language-detection-jobs`의 사용 방법을 보여 줍니다.

AWS CLI

주된 언어 감지 작업을 나열하려면

다음 `list-dominant-language-detection-jobs` 예시에서는 진행 중이거나 완료된 모든 비동기 주된 언어 감지 작업을 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://amzn-s3-demo-bucket",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-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://amzn-s3-demo-bucket",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-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 안내서의 Amazon Comprehend 통찰력 비동기 분석을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListDominantLanguageDetectionJobs 섹션을 참조하세요.

list-endpoints

다음 코드 예시는 list-endpoints의 사용 방법을 보여 줍니다.

AWS CLI

모든 엔드포인트를 나열하려면

다음 list-endpoints 예시에서는 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 안내서의 [Amazon Comprehend 엔드포인트 관리](#) 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ListEndpoints](#) 섹션을 참조하세요.

`list-entities-detection-jobs`

다음 코드 예시는 `list-entities-detection-jobs`의 사용 방법을 보여 줍니다.

AWS CLI

모든 엔터티 감지 작업 나열

다음 `list-entities-detection-jobs` 예시에서는 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://amzn-s3-demo-bucket/AsyncBatchJobs/",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-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://amzn-s3-demo-bucket/AsyncBatchJobs/",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-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://amzn-s3-demo-bucket/AsyncBatchJobs/",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-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 안내서의 엔티티 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListEntitiesDetectionJobs 섹션을 참조하세요.

list-entity-recognizer-summaries

다음 코드 예시는 `list-entity-recognizer-summaries`의 사용 방법을 보여 줍니다.

AWS CLI

생성된 모든 엔터티 인식기에 대한 요약을 나열하려면

다음 `list-entity-recognizer-summaries` 예시에서는 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 Custom entity recognition 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListEntityRecognizerSummaries 섹션을 참조하세요.

list-entity-recognizers

다음 코드 예시는 `list-entity-recognizers`의 사용 방법을 보여 줍니다.

AWS CLI

모든 사용자 지정 엔터티 인식기를 나열하려면

다음 `list-entity-recognizers` 예시에서는 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://amzn-s3-demo-bucket/trainingdata/dataset/",
                    "InputFormat": "ONE_DOC_PER_LINE"
```

```
                },
                "EntityList": {
                    "S3Uri": "s3://amzn-s3-demo-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://amzn-s3-demo-bucket/trainingdata/raw_txt.csv",
                    "InputFormat": "ONE_DOC_PER_LINE"
                },
                "EntityList": {
                    "S3Uri": "s3://amzn-s3-demo-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 Custom entity recognition 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListEntityRecognizers 섹션을 참조하세요.

list-events-detection-jobs

다음 코드 예시는 `list-events-detection-jobs`의 사용 방법을 보여 줍니다.

AWS CLI

모든 이벤트 감지 작업을 나열하려면

다음 `list-events-detection-jobs` 예시에서는 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://amzn-s3-demo-source-bucket/EventsData/",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-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://amzn-s3-demo-source-bucket/EventsData/",
                    "InputFormat": "ONE_DOC_PER_LINE"
                },
                "OutputDataConfig": {
                    "S3Uri": "s3://amzn-s3-demo-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 안내서의 Amazon Comprehend 통찰력 비동기 분석을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListEventsDetectionJobs 섹션을 참조하세요.

`list-flywheel-iteration-history`

다음 코드 예시는 `list-flywheel-iteration-history`의 사용 방법을 보여 줍니다.

AWS CLI

모든 플라이휠 반복 기록을 나열하려면

다음 `list-flywheel-iteration-history` 예시에서는 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://amzn-s3-demo-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://amzn-s3-demo-bucket/example-
flywheel-2/schemaVersion=1/20230616TEXAMPLE/evaluation/20230616TEXAMPLE/"
        }
    ]
}
```

자세한 내용은 Amazon Comprehend 개발자 안내서의 Flywheel overview 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListFlywheelIterationHistory 섹션을 참조하세요.

list-flywheels

다음 코드 예시는 `list-flywheels`의 사용 방법을 보여 줍니다.

AWS CLI

모든 플라이휠을 나열하려면

다음 `list-flywheels` 예시에서는 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://amzn-s3-demo-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://amzn-s3-demo-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 overview 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListFlywheels 섹션을 참조하세요.

list-key-phrases-detection-jobs

다음 코드 예시는 list-key-phrases-detection-jobs의 사용 방법을 보여 줍니다.

AWS CLI

모든 핵심 문구 감지 작업 나열

다음 list-key-phrases-detection-jobs 예시에서는 진행 중이거나 완료된 모든 비동기 핵심 문구 감지 작업을 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://amzn-s3-demo-source-bucket/AsyncBatchJobs/",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-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://amzn-s3-demo-bucket/AsyncBatchJobs/",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-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://amzn-s3-demo-bucket",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-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 안내서의 Amazon Comprehend 통찰력 비동기 분석을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListKeyPhrasesDetectionJobs 섹션을 참조하세요.

list-pii-entities-detection-jobs

다음 코드 예시는 list-pii-entities-detection-jobs의 사용 방법을 보여 줍니다.

AWS CLI

모든 PII 엔터티 감지 작업을 나열하려면

다음 list-pii-entities-detection-jobs 예시에서는 진행 중이거나 완료된 모든 비동기 PII 감지 작업을 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://amzn-s3-demo-bucket/AsyncBatchJobs/",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-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://amzn-s3-demo-bucket/AsyncBatchJobs/",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-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 안내서의 Amazon Comprehend 통찰력 비동기 분석을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListPiiEntitiesDetectionJobs 섹션을 참조하세요.

list-sentiment-detection-jobs

다음 코드 예시는 `list-sentiment-detection-jobs`의 사용 방법을 보여 줍니다.

AWS CLI

모든 감정 감지 작업을 나열하려면

다음 `list-sentiment-detection-jobs` 예시에서는 진행 중이거나 완료된 모든 비동기 감정 감지 작업을 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://amzn-s3-demo-bucket/MovieData",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-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://amzn-s3-demo-bucket/MovieData2",
                    "InputFormat": "ONE_DOC_PER_LINE"
                },
                "OutputDataConfig": {
                    "S3Uri": "s3://amzn-s3-demo-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 안내서의 Amazon Comprehend 통찰력 비동기 분석을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListSentimentDetectionJobs 섹션을 참조하세요.

list-tags-for-resource

다음 코드 예시는 list-tags-for-resource의 사용 방법을 보여 줍니다.

AWS CLI

리소스 태그를 나열하려면

다음 list-tags-for-resource 예시에서는 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 Tagging your resources 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource를 참조하세요.

`list-targeted-sentiment-detection-jobs`

다음 코드 예시는 `list-targeted-sentiment-detection-jobs`의 사용 방법을 보여 줍니다.

AWS CLI

모든 대상 감정 감지 작업을 나열하려면

다음 `list-targeted-sentiment-detection-jobs` 예시에서는 진행 중이거나 완료된 모든 비동기 대상 감정 감지 작업을 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://amzn-s3-demo-bucket/MovieData",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-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://amzn-s3-demo-bucket/MovieData2",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-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 안내서의 Amazon Comprehend 통찰력 비동기 분석을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTargetedSentimentDetectionJobs 섹션을 참조하세요.

list-topics-detection-jobs

다음 코드 예시는 `list-topics-detection-jobs`의 사용 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://amzn-s3-demo-bucket",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-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://amzn-s3-demo-bucket",
```

```
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-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://amzn-s3-demo-bucket",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-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 안내서의 Amazon Comprehend 통찰력 비동기 분석을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTopicsDetectionJobs를 참조하세요.

put-resource-policy

다음 코드 예시는 put-resource-policy의 사용 방법을 보여 줍니다.

AWS CLI

리소스 기반 정책을 연결하려면

다음 put-resource-policy 예시에서는 다른 AWS 계정으로 가져올 수 있도록 리소스 기반 정책을 모델에 연결합니다. 정책은 111122223333 계정의 모델에 연결되며 444455556666 계정을 통해 모델을 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"]}}]}'
```

출력:

```
{
    "PolicyRevisionId": "aaa111d069d07afaa2aa3106aEXAMPLE"
}
```

자세한 내용은 Amazon Comprehend 개발자 안내서의 Copying custom models between AWS accounts 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutResourcePolicy 섹션을 참조하세요.

start-document-classification-job

다음 코드 예시는 start-document-classification-job의 사용 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 스팸과 비스팸 또는 “햄”, SMS 메시지의 문서 분류에 대해 학습되었습니다. 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 출력되지만 여기서는 가독성을 위해 형식이 지정되어 있습니다.

```
aws comprehend start-document-classification-job \
    --job-name exampleclassificationjob \
    --input-data-config "S3Uri=s3://amzn-s3-demo-bucket-INPUT/jobdata/" \
    --output-data-config "S3Uri=s3://amzn-s3-demo-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 안내서의 사용자 지정 분류를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartDocumentClassificationJob을 참조하세요.

start-dominant-language-detection-job

다음 코드 예시는 `start-dominant-language-detection-job`의 사용 방법을 보여 줍니다.

AWS CLI

비동기 언어 감지 작업을 시작하려면

다음 `start-dominant-language-detection-job` 예시에서는 `--input-data-config` 태그가 지정한 주소에 위치한 모든 파일에 대해 비동기 언어 감지 작업을 시작합니다. 이 예시의 S3 버킷에는 `Sampletext1.txt` 파일이 포함되어 있습니다. 작업이 완료되면 `output` 폴더가 `--output-data-config` 태그에 지정된 위치에 배치됩니다. 폴더에는 각 텍스트 파일의 주된 언어와 각 예측에 대한 사전 훈련된 모델의 신뢰도 점수가 포함된 `output.txt`가 포함되어 있습니다.

```
aws comprehend start-dominant-language-detection-job \
    --job-name example_language_analysis_job \
    --language-code en \
    --input-data-config "S3Uri=s3://amzn-s3-demo-bucket/" \
    --output-data-config "S3Uri=s3://amzn-s3-demo-destination-bucket/testfolder/" \
    --data-access-role-arn arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role \
    --language-code en
```

Sampletext1.txt의 콘텐츠:

```
"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.txt의 콘텐츠:

```
{"File": "Sampletext1.txt", "Languages": [{"LanguageCode": "en", "Score": 
 0.9913753867149353}], "Line": 0}
```

자세한 내용은 Amazon Comprehend 개발자 안내서의 Amazon Comprehend 통찰력 비동기 분석을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartDominantLanguageDetectionJob 섹션을 참조하세요.

start-entities-detection-job

다음 코드 예시는 start-entities-detection-job의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 사전 훈련된 모델을 사용하여 표준 엔터티 감지 작업을 시작하려면

다음 start-entities-detection-job 예시에서는 --input-data-config 태그로 지정된 주소에 있는 모든 파일에 대해 비동기 엔터티 감지 작업을 시작합니다. 이 예시의 S3 버킷에는 Sampletext1.txt, Sampletext2.txt, Sampletext3.txt 파일이 포함되어 있습니다. 작업이 완료되면 output 폴더가 --output-data-config 태그에 지정된 위치에 배치됩니다. 이 폴더에는 각 텍스트 파일에서 감지된 모든 이름이 지정된 엔터티와 각 예측에 대한 사전 훈련된 모델의 신뢰도 점수가 나열된 output.txt가 포함되어 있습니다. JSON 출력은 입력 파일당 한 줄로 출력되지만 여기서는 가독성을 위해 형식이 지정되어 있습니다.

```
aws comprehend start-entities-detection-job \
    --job-name entitiestest \
    --language-code en \
    --input-data-config "S3Uri=s3://amzn-s3-demo-bucket/" \
    --output-data-config "S3Uri=s3://amzn-s3-demo-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 위한 줄 들여쓰기가 있는 output.txt의 콘텐츠:

```
{
"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 안내서의 Amazon Comprehend 통찰력 비동기 분석을 참조하세요.

예시 2: 사용자 지정 엔터티 감지 작업을 시작하려면

다음 `start-entities-detection-job` 예시에서는 `--input-data-config` 태그가 지정한 주소에 있는 모든 파일에 대해 비동기 사용자 지정 엔터티 감지 작업을 시작합니다. 이 예시의 S3 버킷에는 `SampleFeedback1.txt`, `SampleFeedback2.txt`, `SampleFeedback3.txt` 파일이 포함되어 있습니다. 엔터티 인식기 모델은 고객 지원 피드백을 통해 디바이스 이름을 인식하도록 훈련되었습니다. 작업이 완료되면 `output` 폴더가 `--output-data-config` 태그로 지정된 위치에 배치됩니다. 이 폴더에는 각 텍스트 파일에서 감지된 모든 이름이 지정된 엔터티와 각 예측에 대한 사전 훈련된 모델의 신뢰도 점수가 나열된 `output.txt`가 포함되어 있습니다. JSON 출력은 파일당 한 줄로 출력되지만 여기서는 가독성을 위해 형식이 지정되어 있습니다.

```
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://amzn-s3-demo-bucket/jobdata/" \
    --output-data-config "S3Uri=s3://amzn-s3-demo-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 위한 줄 들여쓰기가 있는 output.txt의 콘텐츠:

```
{
"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 Custom entity recognition 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartEntitiesDetectionJob 섹션을 참조하세요.

start-events-detection-job

다음 코드 예시는 `start-events-detection-job`의 사용 방법을 보여 줍니다.

AWS CLI

비동기 이벤트 감지 작업을 시작하려면

다음 start-events-detection-job 예시에서는 --input-data-config 태그로 지정된 주소에 있는 모든 파일에 대해 비동기 이벤트 감지 작업을 시작합니다. 가능한 대상 이벤트 유형에는 BANKRUPCTY, EMPLOYMENT, CORPORATE_ACQUISITION, INVESTMENT_GENERAL, CORPORATE_MERGER, IPO, RIGHTS_ISSUE, SECONDARY_OFFERING, SHELF_OFFERING, TENDER_OFFERING, STOCK_SPLIT이 포함됩니다. 이 예시의 S3 버킷에는 SampleText1.txt, SampleText2.txt, SampleText3.txt 파일이 포함되어 있습니다. 작업이 완료되면 output 폴더가 --output-data-config 태그에 지정된 위치에 배치됩니다. 이 폴더에는 SampleText1.txt.out, SampleText2.txt.out, SampleText3.txt.out 파일이 포함되어 있습니다. JSON 출력은 파일당 한 줄로 출력되지만 여기서는 가독성을 위해 형식이 지정되어 있습니다.

```
aws comprehend start-events-detection-job \
    --job-name events-detection-1 \
    --input-data-config "S3Uri=s3://amzn-s3-demo-bucket/EventsData" \
    --output-data-config "S3Uri=s3://amzn-s3-demo-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 있는 `SampleText1.txt.out`의 콘텐츠:

```
{
    "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 안내서의 Amazon Comprehend 통찰력 비동기 분석을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartEventsDetectionJob 섹션을 참조하세요.

start-flywheel-iteration

다음 코드 예시는 start-flywheel-iteration의 사용 방법을 보여 줍니다.

AWS CLI

플라이휠 반복을 시작하려면

다음 start-flywheel-iteration 예시에서는 플라이휠 반복을 시작합니다. 이 작업은 플라이휠의 새 데이터세트를 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 Flywheel overview 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartFlywheelIteration 섹션을 참조하세요.

start-key-phrases-detection-job

다음 코드 예시는 start-key-phrases-detection-job의 사용 방법을 보여 줍니다.

AWS CLI

핵심 문구 감지 작업을 시작하려면

다음 start-key-phrases-detection-job 예시에서는 --input-data-config 태그로 지정된 주소에 있는 모든 파일에 대해 비동기 핵심 문구 감지 작업을 시작합니다. 이 예시의 S3 버킷에는 Sampletext1.txt, Sampletext2.txt, Sampletext3.txt 파일이 포함되어 있습니다. 작업이 완료되면 output 폴더가 --output-data-config 태그로 지정된 위치에 배치됩니다. 이 폴더에는 각 텍스트 파일에서 감지된 모든 핵심 문구와 각 예측에 대한 사전 훈련된 모델의 신뢰도 점수가 포함된 output.txt 파일이 포함되어 있습니다. JSON 출력은 파일당 한 줄로 출력되지만 여기서는 가독성을 위해 형식이 지정되어 있습니다.

```
aws comprehend start-key-phrases-detection-job \
    --job-name keyphrasesanalysistest1 \
    --language-code en \
    --input-data-config "S3Uri=s3://amzn-s3-demo-bucket/" \
    --output-data-config "S3Uri=s3://amzn-s3-demo-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 위한 줄 들여쓰기가 있는 output.txt의 콘텐츠:

```
{
    "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 안내서의 Amazon Comprehend 통찰력 비동기 분석을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartKeyPhrasesDetectionJob 섹션을 참조하세요.

start-pii-entities-detection-job

다음 코드 예시는 start-pii-entities-detection-job의 사용 방법을 보여 줍니다.

AWS CLI

비동기 PII 감지 작업을 시작하려면

다음 start-pii-entities-detection-job 예시에서는 --input-data-config 태그로 지정된 주소에 있는 모든 파일에 대해 비동기 개인 식별 정보(PII) 감지 작업을 시작합니다. 이 예시의 S3 버킷에는 Sampletext1.txt, Sampletext2.txt, Sampletext3.txt 파일이 포함되어 있습니다. 작업이 완료되면 output 폴더가 --output-data-config 태그에 지정된 위치에 배치됩니다. 이 폴더에는 각 텍스트 파일 내의 이름이 지정된 엔터티를 나열하는 SampleText1.txt.out, SampleText2.txt.out, SampleText3.txt.out 파일이 포함되어 있습니다. JSON 출력은 파일당 한 줄로 출력되지만 여기서는 가독성을 위해 형식이 지정되어 있습니다.

```
aws comprehend start-pii-entities-detection-job \
    --job-name entities_test \
    --language-code en \
    --input-data-config "S3Uri=s3://amzn-s3-demo-bucket/" \
    --output-data-config "S3Uri=s3://amzn-s3-demo-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 있는 SampleText1.txt.out의 콘텐츠:

```
{
    "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 위한 줄 들여쓰기가 있는 `SampleText2.txt.out`의 콘텐츠:

```
{
    "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 있는 `SampleText3.txt.out`의 콘텐츠:

```
{
        "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 안내서의 [Amazon Comprehend 통찰력 비동기 분석](#)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartPiiEntitiesDetectionJob 섹션을 참조하세요.

start-sentiment-detection-job

다음 코드 예시는 start-sentiment-detection-job의 사용 방법을 보여 줍니다.

AWS CLI

비동기 감정 분석 작업을 시작하려면

다음 start-sentiment-detection-job 예시에서는 --input-data-config 태그로 지정된 주소에 있는 모든 파일에 대해 비동기 감정 분석 감지 작업을 시작합니다. 이 예시의 S3 버킷에는 SampleMovieReview1.txt, SampleMovieReview2.txt, SampleMovieReview3.txt 파일이 포함되어 있습니다. 작업이 완료되면 output 폴더가 --output-data-config 태그에 지정된 위치에 배치됩니다. 이 폴더에는 각 텍스트 파일의 주된 감정과 각 예측에 대한 사전 훈련된 모델의 신뢰도 점수가 포함된 output.txt 파일이 포함되어 있습니다. JSON 출력은 파일당 한 줄로 출력되지만 여기서는 가독성을 위해 형식이 지정되어 있습니다.

```
aws comprehend start-sentiment-detection-job \
    --job-name example-sentiment-detection-job \
    --language-code en \
    --input-data-config "S3Uri=s3://amzn-s3-demo-bucket/MovieData" \
    --output-data-config "S3Uri=s3://amzn-s3-demo-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 있는 `output.txt`의 콘텐츠:

```
{
    "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 안내서의 Amazon Comprehend 통찰력 비동기 분석을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartSentimentDetectionJob 섹션을 참조하세요.

start-targeted-sentiment-detection-job

다음 코드 예시는 start-targeted-sentiment-detection-job의 사용 방법을 보여 줍니다.

AWS CLI

비동기 대상 감정 분석 작업을 시작하려면

다음 start-targeted-sentiment-detection-job 예시에서는 --input-data-config 태그로 지정된 주소에 있는 모든 파일에 대해 비동기 대상 감정 분석 감지 작업을 시작합니다. 이 예시의 S3 버킷에는 SampleMovieReview1.txt, SampleMovieReview2.txt, SampleMovieReview3.txt 파일이 포함되어 있습니다. 작업이 완료되면 output.tar.gz는 --output-data-config 태그에 의해 지정된 위치에 배치됩니다. output.tar.gz에는 SampleMovieReview1.txt.out, SampleMovieReview2.txt.out, SampleMovieReview3.txt.out 파일이 포함되어 있으며, 각 파일에는 단일 입력 텍스트 파일에 대한 이름이 지정된 엔터티 및 관련 감정이 모두 포함되어 있습니다.

```
aws comprehend start-targeted-sentiment-detection-job \
    --job-name targeted_movie_review_analysis1 \
    --language-code en \
    --input-data-config "S3Uri=s3://amzn-s3-demo-bucket/MovieData" \
    --output-data-config "S3Uri=s3://amzn-s3-demo-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 있는 SampleMovieReview1.txt.out의 콘텐츠:


```
{
    "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 줄 들여쓰기가 있는 `SampleMovieReview2.txt.out`의 콘텐츠:

```
{
    "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 위한 줄 들여쓰기가 있는 `SampleMovieReview3.txt.out`의 콘텐츠:

```
{
    "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 안내서의 Amazon Comprehend 통찰력 비동기 분석을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartTargetedSentimentDetectionJob 섹션을 참조하세요.

start-topics-detection-job

다음 코드 예시는 `start-topics-detection-job`의 사용 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. `output`에는 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://amzn-s3-demo-bucket/" \
    --output-data-config "S3Uri=s3://amzn-s3-demo-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 안내서의 주제 모델링을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartTopicsDetectionJob을 참조하세요.

stop-dominant-language-detection-job

다음 코드 예시는 stop-dominant-language-detection-job의 사용 방법을 보여 줍니다.

AWS CLI

비동기 주된 언어 감지 작업을 중지하려면

다음 stop-dominant-language-detection-job 예시에서는 진행 중인 비동기 주된 언어 감지 작업을 중지합니다. 현재 작업 상태가 IN_PROGRESS인 경우 작업은 종료로 표시되고 STOP_REQUESTED 상태로 전환됩니다. 작업을 중지하기 전에 작업이 완료되면 COMPLETED 상태로 전환됩니다.

```
aws comprehend stop-dominant-language-detection-job \
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

출력:

```
{
```

```
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE,
    "JobStatus": "STOP_REQUESTED"
}
```

자세한 내용은 Amazon Comprehend 개발자 안내서의 Amazon Comprehend 통찰력 비동기 분석을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StopDominantLanguageDetectionJob 섹션을 참조하세요.

stop-entities-detection-job

다음 코드 예시는 stop-entities-detection-job의 사용 방법을 보여 줍니다.

AWS CLI

비동기 엔터티 감지 작업을 중지하려면

다음 stop-entities-detection-job 예시에서는 진행 중인 비동기 엔터티 감지 작업을 중지합니다. 현재 작업 상태가 IN_PROGRESS인 경우 작업은 종료로 표시되고 STOP_REQUESTED 상태로 전환됩니다. 작업을 중지하기 전에 작업이 완료되면 COMPLETED 상태로 전환됩니다.

```
aws comprehend stop-entities-detection-job \
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

출력:

```
{
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE,
    "JobStatus": "STOP_REQUESTED"
}
```

자세한 내용은 Amazon Comprehend 개발자 안내서의 Amazon Comprehend 통찰력 비동기 분석을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StopEntitiesDetectionJob 섹션을 참조하세요.

stop-events-detection-job

다음 코드 예시는 stop-events-detection-job의 사용 방법을 보여 줍니다.

AWS CLI

비동기 이벤트 감지 작업을 중지하려면

다음 stop-events-detection-job 예시에서는 진행 중인 비동기 이벤트 감지 작업을 중지합니다. 현재 작업 상태가 IN_PROGRESS인 경우 작업은 종료로 표시되고 STOP_REQUESTED 상태로 전환됩니다. 작업을 중지하기 전에 작업이 완료되면 COMPLETED 상태로 전환됩니다.

```
aws comprehend stop-events-detection-job \
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

출력:

```
{
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE,
    "JobStatus": "STOP_REQUESTED"
}
```

자세한 내용은 Amazon Comprehend 개발자 안내서의 Amazon Comprehend 통찰력 비동기 분석을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StopEventsDetectionJob 섹션을 참조하세요.

stop-key-phrases-detection-job

다음 코드 예시는 stop-key-phrases-detection-job의 사용 방법을 보여 줍니다.

AWS CLI

비동기 핵심 문구 감지 작업을 중지하려면

다음 stop-key-phrases-detection-job 예시에서는 진행 중인 비동기 핵심 문구 감지 작업을 중지합니다. 현재 작업 상태가 IN_PROGRESS인 경우 작업은 종료로 표시되고 STOP_REQUESTED 상태로 전환됩니다. 작업을 중지하기 전에 작업이 완료되면 COMPLETED 상태로 전환됩니다.

```
aws comprehend stop-key-phrases-detection-job \
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

출력:

```
{
```

```
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE,
    "JobStatus": "STOP_REQUESTED"
}
```

자세한 내용은 Amazon Comprehend 개발자 안내서의 Amazon Comprehend 통찰력 비동기 분석을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StopKeyPhrasesDetectionJob 섹션을 참조하세요.

stop-pii-entities-detection-job

다음 코드 예시는 stop-pii-entities-detection-job의 사용 방법을 보여 줍니다.

AWS CLI

비동기 PII 엔터티 감지 작업을 중지하려면

다음 stop-pii-entities-detection-job 예시에서는 진행 중인 비동기 PII 엔터티 감지 작업을 중지합니다. 현재 작업 상태가 IN_PROGRESS인 경우 작업은 종료로 표시되고 STOP_REQUESTED 상태로 전환됩니다. 작업을 중지하기 전에 작업이 완료되면 COMPLETED 상태로 전환됩니다.

```
aws comprehend stop-pii-entities-detection-job \
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

출력:

```
{
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE,
    "JobStatus": "STOP_REQUESTED"
}
```

자세한 내용은 Amazon Comprehend 개발자 안내서의 Amazon Comprehend 통찰력 비동기 분석을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StopPiiEntitiesDetectionJob 섹션을 참조하세요.

stop-sentiment-detection-job

다음 코드 예시는 stop-sentiment-detection-job의 사용 방법을 보여 줍니다.

AWS CLI

비동기 감정 감지 작업을 중지하려면

다음 stop-sentiment-detection-job 예시에서는 진행 중인 비동기 감정 감지 작업을 중지합니다. 현재 작업 상태가 IN_PROGRESS인 경우 작업은 종료로 표시되고 STOP_REQUESTED 상태로 전환됩니다. 작업을 중지하기 전에 작업이 완료되면 COMPLETED 상태로 전환됩니다.

```
aws comprehend stop-sentiment-detection-job \
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

출력:

```
{
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE,
    "JobStatus": "STOP_REQUESTED"
}
```

자세한 내용은 Amazon Comprehend 개발자 안내서의 Amazon Comprehend 통찰력 비동기 분석을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StopSentimentDetectionJob 섹션을 참조하세요.

stop-targeted-sentiment-detection-job

다음 코드 예시는 stop-targeted-sentiment-detection-job의 사용 방법을 보여 줍니다.

AWS CLI

비동기 대상 감정 감지 작업을 중지하려면

다음 stop-targeted-sentiment-detection-job 예시에서는 진행 중인 비동기 대상 감정 감지 작업을 중지합니다. 현재 작업 상태가 IN_PROGRESS인 경우 작업은 종료로 표시되고 STOP_REQUESTED 상태로 전환됩니다. 작업을 중지하기 전에 작업이 완료되면 COMPLETED 상태로 전환됩니다.

```
aws comprehend stop-targeted-sentiment-detection-job \
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

출력:

```
{
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE,
    "JobStatus": "STOP_REQUESTED"
}
```

자세한 내용은 Amazon Comprehend 개발자 안내서의 Amazon Comprehend 통찰력 비동기 분석을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StopTargetedSentimentDetectionJob 섹션을 참조하세요.

stop-training-document-classifier

다음 코드 예시는 `stop-training-document-classifier`의 사용 방법을 보여 줍니다.

AWS CLI

문서 분류기 모델의 훈련을 중지하려면

다음 `stop-training-document-classifier` 예시에서는 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 안내서의 사용자 지정 모델 생성 및 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StopTrainingDocumentClassifier 섹션을 참조하세요.

stop-training-entity-recognizer

다음 코드 예시는 `stop-training-entity-recognizer`의 사용 방법을 보여 줍니다.

AWS CLI

엔터티 인식기 모델의 훈련을 중지하려면

다음 `stop-training-entity-recognizer` 예시에서는 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 안내서의 사용자 지정 모델 생성 및 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StopTrainingEntityRecognizer 섹션을 참조하세요.

`tag-resource`

다음 코드 예시는 `tag-resource`의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 리소스에 태그를 지정하려면

다음 `tag-resource` 예시에서는 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 Tagging your resources 섹션을 참조하세요.

예시 2: 리소스에 여러 태그를 추가하려면

다음 `tag-resource` 예시에서는 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 Tagging your resources 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

untag-resource

다음 코드 예시는 untag-resource의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 리소스에서 단일 태그를 제거하려면

다음 untag-resource 예시에서는 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 Tagging your resources 섹션을 참조하세요.

예시 2: 리소스에서 여러 태그를 제거하려면

다음 untag-resource 예시에서는 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 Tagging your resources 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

update-endpoint

다음 코드 예시는 update-endpoint의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 엔드포인트의 추론 단위를 업데이트하려면

다음 update-endpoint 예시에서는 엔드포인트에 대한 정보를 업데이트합니다. 이 예시에서는 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 안내서의 Amazon Comprehend 엔드포인트 관리 섹션을 참조하세요.

예시 2: 엔드포인트의 활성 모델을 업데이트하려면

다음 update-endpoint 예시에서는 엔드포인트에 대한 정보를 업데이트합니다. 이 예시에서는 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 안내서의 Amazon Comprehend 엔드포인트 관리 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateEndpoint 섹션을 참조하세요.

update-flywheel

다음 코드 예시는 update-flywheel의 사용 방법을 보여 줍니다.

AWS CLI

플라이휠 구성을 업데이트하려면

다음 `update-flywheel` 예시에서는 플라이휠 구성을 업데이트합니다. 이 예시에서는 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://amzn-s3-demo-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 overview 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateFlywheel 섹션을 참조하세요.

AWS CLI를 사용한 Amazon Comprehend Medical 예제

다음 코드 예제는 Amazon Comprehend Medical과 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`describe-entities-detection-v2-job`

다음 코드 예시는 `describe-entities-detection-v2-job`의 사용 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 안내서에서 Batch APIs를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeEntitiesDetectionV2Job을 참조하세요.

describe-icd10-cm-inference-job

다음 코드 예시는 describe-icd10-cm-inference-job의 사용 방법을 보여 줍니다.

AWS CLI

ICD-10-CM 추론 작업을 설명하려면

다음 describe-icd10-cm-inference-job 예제에서는 지정된 job-id로 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 안내서에서 Ontology linking batch analysis를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeIcd10CmInferenceJob을 참조하세요.

describe-phi-detection-job

다음 코드 예시는 describe-phi-detection-job의 사용 방법을 보여 줍니다.

AWS CLI

PHI 감지 작업을 설명하려면

다음 describe-phi-detection-job 예제에서는 비동기 보호 대상 건강 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 안내서에서 Batch APIs를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribePhiDetectionJob을 참조하세요.

describe-rx-norm-inference-job

다음 코드 예시는 `describe-rx-norm-inference-job`의 사용 방법을 보여 줍니다.

AWS CLI

RxNorm 추론 작업을 설명하려면

다음 `describe-rx-norm-inference-job` 예제에서는 지정된 job-id로 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 안내서에서 Ontology linking batch analysis를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeRxNormInferenceJob을 참조하세요.

describe-snomedct-inference-job

다음 코드 예시는 describe-snomedct-inference-job의 사용 방법을 보여 줍니다.

AWS CLI

SNOMED CT 추론 작업을 설명하려면

다음 describe-snomedct-inference-job 예제에서는 지정된 job-id로 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 안내서에서 Ontology linking batch analysis를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeSnomedctInferenceJob을 참조하세요.

detect-entities-v2

다음 코드 예시는 detect-entities-v2의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 텍스트에서 직접 엔터티를 감지하는 방법

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 안내서에서 Detect Entities Version 2를 참조하세요.

예제 2: 파일 경로에서 엔터티를 감지하는 방법

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.txt`의 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 안내서에서 Detect Entities Version 2를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DetectEntitiesV2를 참조하세요.

detect-phi

다음 코드 예시는 detect-phi의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 텍스트에서 보호 대상 건강 정보(PHI)를 직접 감지하는 방법

다음 detect-phi 예제에서는 입력 텍스트에서 직접 감지된 보호 대상 건강 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 안내서에서 Detect PHI를 참조하세요.

예제 2: 파일 경로에서 보호 대상 건강 정보(PHI)를 직접 감지하는 방법

다음 `detect-phi` 예제에서는 파일 경로에서 감지된 보호 대상 건강 정보(PHI) 엔터티를 보여줍니다.

```
aws comprehendmedical detect-phi \
    --text file://phi.txt
```

`phi.txt`의 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 안내서에서 Detect PHI를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DetectPhi를 참조하세요.

infer-icd10-cm

다음 코드 예시는 infer-icd10-cm의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 의료 상태 엔터티를 감지하고 텍스트에서 ICD-10-CM 온톨로지로 직접 연결하는 방법

다음 infer-icd10-cm 예제에서는 감지된 의료 상태 엔터티에 레이블을 지정하고 해당 엔터티를 국제질병분류 임상 수정(ICD-10-CM) 2019판의 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 안내서에서 Infer ICD10-CM을 참조하세요.

예제 2: 의료 상태 엔터티를 감지하고 파일 경로에서 ICD-10-CM 온톨로지로 연결하는 방법

다음 `infer-icd-10-cm` 예제에서는 감지된 의료 상태 엔터티에 레이블을 지정하고 해당 엔터티를 국제질병분류 임상 수정(ICD-10-CM) 2019판의 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 안내서에서 Infer-ICD10-CM을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 InferIcd10Cm을 참조하세요.

infer-rx-norm

다음 코드 예시는 `infer-rx-norm`의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 약물 엔터티를 감지하고 텍스트에서 RxNorm에 직접 연결하는 방법

다음 `infer-rx-norm` 예제에서는 감지된 약물 엔터티를 표시 및 레이블을 지정하고 해당 엔터티를 국립의학도서관 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 안내서에서 Infer RxNorm을 참조하세요.

예제 2: 약물 엔터티를 감지하고 파일 경로에서 RxNorm에 연결하는 방법

다음 `infer-rx-norm` 예제에서는 감지된 약물 엔터티를 표시 및 레이블을 지정하고 해당 엔터티를 국립의학도서관 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 안내서에서 Infer RxNorm을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 InferRxNorm을 참조하세요.

infer-snomedct

다음 코드 예시는 `infer-snomedct`의 사용 방법을 보여 줍니다.

AWS CLI

예제: 엔터티를 감지하고 텍스트에서 SNOMED CT 온톨로지에 직접 연결하는 방법

다음 `infer-snomedct` 예제에서는 의료 엔터티를 감지하고 이를 2021-03 버전의 체계화된 의학 명명법, 임상 용어(SNOMED CT) 개념과 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 안내서에서 InferSNOMEDCT를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 InferSnomedct를 참조하세요.

list-entities-detection-v2-jobs

다음 코드 예시는 `list-entities-detection-v2-jobs`의 사용 방법을 보여 줍니다.

AWS CLI

엔터티 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 안내서에서 Batch APIs를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListEntitiesDetectionV2Jobs를 참조하세요.

list-icd10-cm-inference-jobs

다음 코드 예시는 `list-icd10-cm-inference-jobs`의 사용 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 안내서에서 Ontology linking batch analysis를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListIcd10CmInferenceJobs를 참조하세요.

list-phi-detection-jobs

다음 코드 예시는 list-phi-detection-jobs의 사용 방법을 보여 줍니다.

AWS CLI

보호 대상 건강 정보(PHI) 감지 작업을 나열하려면

다음 list-phi-detection-jobs 예제에서는 현재 보호 대상 건강 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 안내서에서 Batch APIs를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListPhiDetectionJobs를 참조하세요.

list-rx-norm-inference-jobs

다음 코드 예시는 list-rx-norm-inference-jobs의 사용 방법을 보여 줍니다.

AWS CLI

현재 Rx-Norm 추론 작업을 모두 나열하려면

다음 예제에서는 list-rx-norm-inference-jobs가 현재 비동기 Rx-Norm 배치 추론 작업 목록을 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 안내서에서 Ontology linking batch analysis를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListRxNormInferenceJobs를 참조하세요.

list-snomedct-inference-jobs

다음 코드 예시는 `list-snomedct-inference-jobs`의 사용 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 안내서에서 Ontology linking batch analysis를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListSnomedctInferenceJobs를 참조하세요.

start-entities-detection-v2-job

다음 코드 예시는 `start-entities-detection-v2-job`의 사용 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 안내서에서 Batch APIs를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartEntitiesDetectionV2Job을 참조하세요.

`start-icd10-cm-inference-job`

다음 코드 예시는 `start-icd10-cm-inference-job`의 사용 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 안내서에서 Ontology linking batch analysis를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartIcd10CmInferenceJob을 참조하세요.

start-phi-detection-job

다음 코드 예시는 start-phi-detection-job의 사용 방법을 보여 줍니다.

AWS CLI

PHI 감지 작업을 시작하려면

다음 start-phi-detection-job 예제에서는 비동기 PHI 엔터티 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 안내서에서 Batch APIs를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartPhiDetectionJob을 참조하세요.

start-rx-norm-inference-job

다음 코드 예시는 start-rx-norm-inference-job의 사용 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 안내서에서 Ontology linking batch analysis를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartRxNormInferenceJob을 참조하세요.

start-snomedct-inference-job

다음 코드 예시는 `start-snomedct-inference-job`의 사용 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 안내서에서 Ontology linking batch analysis를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartSnomedctInferenceJob을 참조하세요.

stop-entities-detection-v2-job

다음 코드 예시는 stop-entities-detection-v2-job의 사용 방법을 보여 줍니다.

AWS CLI

엔터티 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 안내서에서 Batch APIs를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StopEntitiesDetectionV2Job을 참조하세요.

stop-icd10-cm-inference-job

다음 코드 예시는 stop-icd10-cm-inference-job의 사용 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 안내서에서 Ontology linking batch analysis를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StopIcd10CmInferenceJob을 참조하세요.

stop-phi-detection-job

다음 코드 예시는 `stop-phi-detection-job`의 사용 방법을 보여 줍니다.

AWS CLI

보호 대상 건강 정보(PHI) 감지 작업을 중지하려면

다음 `stop-phi-detection-job` 예제에서는 비동기 보호 대상 건강 정보(PHI) 감지 작업을 중지합니다.

```
aws comprehendmedical stop-phi-detection-job \
    --job-id "4750034166536cdb52ffa3295a1b00a3"
```

출력:

```
{
    "JobId": "ab9887877365fe70299089371c043b96"
}
```

자세한 내용은 Amazon Comprehend Medical 개발자 안내서에서 Batch APIs를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StopPhiDetectionJob을 참조하세요.

stop-rx-norm-inference-job

다음 코드 예시는 `stop-rx-norm-inference-job`의 사용 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 안내서에서 Ontology linking batch analysis를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StopRxNormInferenceJob을 참조하세요.

stop-snomedct-inference-job

다음 코드 예시는 stop-snomedct-inference-job의 사용 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 안내서에서 Ontology linking batch analysis를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StopSnomedctInferenceJob을 참조하세요.

AWS CLI를 사용한 AWS Config 예시

다음 코드 예시에서는 AWS Config에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`delete-config-rule`

다음 코드 예시에서는 `delete-config-rule` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS Config 규칙을 삭제하는 방법

다음 명령은 이름이 `MyConfigRule`인 AWS Config 규칙을 삭제합니다.

```
aws configservice delete-config-rule --config-rule-name MyConfigRule
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteConfigRule을 참조하세요.

`delete-delivery-channel`

다음 코드 예시에서는 `delete-delivery-channel` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

전송 채널을 삭제하는 방법

다음 명령은 기본 전송 채널을 삭제합니다.

```
aws configservice delete-delivery-channel --delivery-channel-name default
```

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteDeliveryChannel] 섹션을 참조하세요.

delete-evaluation-results

다음 코드 예시에서는 `delete-evaluation-results` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

평가 결과를 수동으로 삭제하는 방법

다음 명령은 AWS 관리형 규칙 s3-bucket-versioning-enabled에 대한 현재 평가 결과를 삭제합니다.

```
aws configservice delete-evaluation-results --config-rule-name s3-bucket-versioning-enabled
```

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteEvaluationResults] 섹션을 참조하세요.

deliver-config-snapshot

다음 코드 예시에서는 `deliver-config-snapshot` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

구성 스냅샷 전달

다음 명령은 기본 배달 채널에 속하는 Amazon S3 버킷에 구성 스냅샷을 전달합니다.

```
aws configservice deliver-config-snapshot --delivery-channel-name default
```

출력:

```
{
    "configSnapshotId": "d0333b00-a683-44af-921e-examplefb794"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [DeliverConfigSnapshot] 섹션을 참조하세요.

describe-compliance-by-config-rule

다음 코드 예시에서는 `describe-compliance-by-config-rule` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWSConfig 규칙에 대한 규정 준수 정보를 가져오는 방법

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 InstanceTypesAreT2micro라는 규칙을 준수하지 않음을 나타냅니다.

출력:

```
{
    "ComplianceByConfigRules": [
        {
            "Compliance": {
                "ComplianceContributorCount": {
                    "CappedCount": 3,
                    "CapExceeded": false
                },
                "ComplianceType": "NON_COMPLIANT"
            },
            "ConfigRuleName": "InstanceTypesAreT2micro"
        },
        {
            "Compliance": {
                "ComplianceContributorCount": {
                    "CappedCount": 10,
                    "CapExceeded": false
                },
                "ComplianceType": "NON_COMPLIANT"
            },
            "ConfigRuleName": "RequiredTagsForVolumes"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeComplianceByConfigRule 섹션을 참조하세요.

`describe-compliance-by-resource`

다음 코드 예시에서는 `describe-compliance-by-resource` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS 리소스에 대한 규정 준수 정보를 가져오는 방법

다음 명령은 AWS Config가 기록하고 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 인스턴스 `i-1a2b3c4d`가 2개의 규칙을 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"
            }
        }
```

```
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeComplianceByResource](#) 섹션을 참조하세요.

describe-config-rule-evaluation-status

다음 코드 예시에서는 `describe-config-rule-evaluation-status` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWSConfig 규칙의 상태 정보를 가져오는 방법

다음 명령은 이름이 `MyConfigRule`인 AWS Config 규칙의 상태 정보를 반환합니다.

```
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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeConfigRuleEvaluationStatus](#) 섹션을 참조하세요.

describe-config-rules

다음 코드 예시에서는 `describe-config-rules` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS Config 규칙의 세부 정보를 가져오는 방법

다음 명령은 이름이 `InstanceTypesAreT2micro`인 AWS Config 규칙의 세부 정보를 반환합니다.

```
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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeConfigRules](#)를 참조하세요.

describe-configuration-recorder-status

다음 코드 예시에서는 describe-configuration-recorder-status 코드를 사용하는 방법을 보여줍니다.

AWS CLI

구성 레코더의 상태 정보를 가져오는 방법

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
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeInstanceAssociationsStatus 섹션을 참조하세요.

describe-configuration-recorders

다음 코드 예시에서는 describe-configuration-recorders 코드를 사용하는 방법을 보여줍니다.

AWS CLI

구성 레코더에 대한 세부 정보를 가져오는 방법

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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeConfigurationRecorders](https://docs.aws.amazon.com/goto/aws-cli/config-2014-11-12/DescribeConfigurationRecorders) 섹션을 참조하세요.

describe-delivery-channel-status

다음 코드 예시에서는 `describe-delivery-channel-status` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

전송 채널의 상태 정보를 가져오는 방법

다음 명령은 배달 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDeliveryChannelStatus 섹션을 참조하세요.

describe-delivery-channels

다음 코드 예시에서는 describe-delivery-channels 코드를 사용하는 방법을 보여줍니다.

AWS CLI

전송 채널에 대한 세부 정보를 가져오는 방법

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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDeliveryChannels 섹션을 참조하세요.

get-compliance-details-by-config-rule

다음 코드 예시에서는 get-compliance-details-by-config-rule 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWSConfig 규칙에 대한 평가 결과를 가져오는 방법

다음 명령은 InstanceTypesAreT2micro라는 AWS Config 규칙을 준수하지 않는 모든 리소스에 대한 평가 결과를 반환합니다.

```
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"
            }
        ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetComplianceDetailsByConfigRule 섹션을 참조하세요.

get-compliance-details-by-resource

다음 코드 예시에서는 `get-compliance-details-by-resource` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS 리소스에 대한 평가 결과를 가져오는 방법

다음 명령은 EC2 인스턴스 `i-1a2b3c4d`가 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"
        }
    ]
}
```


- API 세부 정보는 AWS CLI 명령 참조의 GetComplianceDetailsByResource 섹션을 참조하세요.

`get-compliance-summary-by-config-rule`

다음 코드 예시에서는 `get-compliance-summary-by-config-rule` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWSConfig 규칙에 대한 규정 준수 요약을 가져오는 방법

다음 명령은 규정을 준수하는 규칙의 수와 규정을 미준수하는 규칙의 수를 반환합니다.

```
aws configservice get-compliance-summary-by-config-rule
```

출력에서 각 `CappedCount` 속성의 값은 규정 준수 또는 규정 미준수 규칙의 수를 나타냅니다.

출력:

```
{
    "ComplianceSummary": {
        "NonCompliantResourceCount": {
            "CappedCount": 3,
            "CapExceeded": false
        },
        "ComplianceSummaryTimestamp": 1452204131.493,
        "CompliantResourceCount": {
            "CappedCount": 2,
            "CapExceeded": false
        }
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetComplianceSummaryByConfigRule 섹션을 참조하세요.

get-compliance-summary-by-resource-type

다음 코드 예시에서는 `get-compliance-summary-by-resource-type` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

모든 리소스 유형에 대한 규정 준수 요약을 가져오는 방법

다음 명령은 규칙을 준수하는 리소스의 수와 규칙을 미준수하는 리소스의 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 가져오는 방법

다음 명령은 규정을 준수하지 않는 EC2 인스턴스 수와 규정을 준수하는 EC2 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
                }
            }
        }
    ]
```

```
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetComplianceSummaryByResourceType 섹션을 참조하세요.

get-resource-config-history

다음 코드 예시에서는 get-resource-config-history 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS 리소스의 구성 기록을 가져오는 방법

다음 명령은 ID가 i-1a2b3c4d인 EC2 인스턴스에 대한 구성 항목 목록을 반환합니다.

```
aws configservice get-resource-config-history --resource-type AWS::EC2::Instance --
resource-id i-1a2b3c4d
```

- API 세부 정보는 AWS CLI 명령 참조의 GetResourceConfigHistory 섹션을 참조하세요.

get-status

다음 코드 예시에서는 get-status 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWSConfig 상태를 가져오는 방법

다음 명령은 배달 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
```

- API 세부 정보는 AWS CLI 명령 참조의 GetStatus 섹션을 참조하세요.

list-discovered-resources

다음 코드 예시에서는 `list-discovered-resources` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWSConfig가 검색한 리소스를 나열하는 방법

다음 명령은 AWS Config가 검색한 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListDiscoveredResources 섹션을 참조하세요.

put-config-rule

다음 코드 예시에서는 `put-config-rule` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS 관리형 Config 규칙을 추가하는 방법

다음 명령은 AWS 관리형 Config 규칙을 추가하기 위한 JSON 코드를 제공합니다.

```
aws configservice put-config-rule --config-rule file://
RequiredTagsForEC2Instances.json
```

`RequiredTagsForEC2Instances.json`은 규칙 구성이 포함된 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 따라 EC2 인스턴스만 평가합니다. 규칙은 관리형 규칙이므로 `Owner` 속성은 `AWS`로 설정되고 `SourceIdentifier` 속성은 규칙 식별자인 `REQUIRED_TAGS`로 설정됩니다. `InputParameters` 속성의 경우 규칙에 필요한 태그 키인 `CostCenter` 및 `Owner`가 지정됩니다.

명령이 성공하면 AWS Config는 출력을 반환하지 않습니다. 규칙 구성을 확인하려면 describe-config-rules 명령을 실행하고 규칙 이름을 지정합니다.

고객 관리형 Config 규칙을 추가하는 방법

다음 명령은 고객 관리형 Config 규칙을 추가하기 위한 JSON 코드를 제공합니다.

```
aws configservice put-config-rule --config-rule file://InstanceTypesAreT2micro.json
```

InstanceTypesAreT2micro.json은 규칙 구성이 포함된 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 따라 EC2 인스턴스만 평가합니다. 이 규칙은 고객 관리형 규칙이므로 Owner 속성은 CUSTOM_LAMBDA로 설정되고 SourceIdentifier 속성은 AWS Lambda 함수의 ARN으로 설정됩니다. SourceDetails 객체가 필요합니다. AWS Config가 규칙을 기준으로 리소스를 평가하기 위해 AWS Lambda 함수를 호출하면 InputParameters 속성에 지정된 파라미터가 함수에 전달됩니다.

명령이 성공하면 AWS Config는 출력을 반환하지 않습니다. 규칙 구성을 확인하려면 describe-config-rules 명령을 실행하고 규칙 이름을 지정합니다.

- API 세부 정보는 AWS CLI 명령 참조의 PutConfigRule을 참조하세요.

put-configuration-recorder

다음 코드 예시에서는 put-configuration-recorder 코드를 사용하는 방법을 보여줍니다.

AWS CLI

예시 1: 지원되는 모든 리소스를 기록하는 방법

다음 명령은 전역 리소스 유형을 포함하여 지원되는 모든 리소스 유형의 변경 사항을 추적하는 구성 레코더를 만듭니다.

```
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 describe-configuration-recorders 명령을 실행합니다.

예시 2: 특정 유형의 리소스를 기록하는 방법

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 리소스 유형을 지정하려면 먼저 allSupported 및 includeGlobalResourceTypes 옵션을 false로 설정하거나 생략해야 합니다.

명령이 성공하면 AWS Config는 출력을 반환하지 않습니다. 구성 레코더의 설정을 확인하려면 describe-configuration-recorders 명령을 실행합니다.

예시 3: 특정 유형의 리소스를 제외한 지원되는 모든 리소스를 선택하는 방법

다음 명령은 --recording-group 옵션에 대해 JSON 파일에 지정된 리소스 유형을 제외한 현재 및 향후 지원되는 모든 리소스 유형의 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 먼저 1) allSupported 및 includeGlobalResourceTypes 옵션을 false 또는 생략으로 설정해야 하며, 2) RecordingStrategy의 useOnly 필드를 EXCLUSION_BY_RESOURCE_TYPES로 설정해야 합니다.

명령이 성공하면 AWS Config는 출력을 반환하지 않습니다. 구성 레코더의 설정을 확인하려면 describe-configuration-recorders 명령을 실행합니다.

- API 세부 정보는 AWS CLI 명령 참조의 PutConfigurationRecorder 섹션을 참조하세요.

put-delivery-channel

다음 코드 예시에서는 `put-delivery-channel` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

전송 채널을 생성하는 방법

다음 명령은 전송 채널에 대한 설정을 JSON 코드로 제공합니다.

```
aws configservice put-delivery-channel --delivery-channel file://
deliveryChannel.json
```

`deliveryChannel.json` 파일이 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 Config가 새 전송 채널에 `default` 이름을 할당합니다. `put-delivery-channel` 명령을 사용하여 전송 채널 이름을 업데이트할 수 없습니다. 이름을 변경하는 단계는 전송 채널 이름 변경을 참조하세요.`s3BucketName` - AWS Config이 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 Config이 구성 변경에 대한 알림을 보내는 Amazon SNS 주제의 Amazon 리소스 이름(ARN)입니다. 다른 계정에서 주제를 선택하는 경우 주제에 AWS Config에 액세스 권한을 부여하는 정책이 있어야 합니다. 자세한 내용은 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 describe-delivery-channels 명령을 실행합니다.

- API 세부 정보는 AWS CLI 명령 참조의 [PutDeliveryChannel](#) 섹션을 참조하세요.

`start-config-rules-evaluation`

다음 코드 예시에서는 `start-config-rules-evaluation` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWSConfig 규칙에 대한 온디맨드 평가를 실행하는 방법

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
```

- API 세부 정보는 AWS CLI 명령 참조의 [StartConfigRulesEvaluation](#) 섹션을 참조하세요.

`start-configuration-recorder`

다음 코드 예시에서는 `start-configuration-recorder` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

구성 레코더 시작

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.

- API 세부 정보는 AWS CLI 명령 참조의 [StartConfigurationRecorder](#) 섹션을 참조하세요.

`stop-configuration-recorder`

다음 코드 예시에서는 `stop-configuration-recorder` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

구성 레코더 중지

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.

- API 세부 정보는 AWS CLI 명령 참조의 StopConfigurationRecorder 섹션을 참조하세요.

subscribe

다음 코드 예시에서는 `subscribe` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS Config를 구독하는 방법

다음 명령은 기본 배달 채널 및 구성 레코더를 생성합니다. 또한 이 명령은 AWS Config가 구성 정보를 전달할 Amazon S3 버킷과 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 세부 정보는 AWS CLI명령 참조의 Subscribe를 참조하세요.

AWS CLI를 사용한 Amazon Connect 예제

다음 코드 예제에서는 Amazon Connect에서 AWS Command Line Interface를 사용하여 일반적인 시나리오를 구현하는 방법을 보여 줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

create-user

다음 코드 예시는 `create-user`의 사용 방법을 보여 줍니다.

AWS CLI

사용자를 생성하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateUser를 참조하세요.

delete-user

다음 코드 예시는 delete-user의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteUser를 참조하세요.

describe-user-hierarchy-group

다음 코드 예시는 describe-user-hierarchy-group의 사용 방법을 보여 줍니다.

AWS CLI

계층 그룹에 대한 세부 정보를 표시하는 방법

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 구조 설정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeUserHierarchyGroup 섹션을 참조하세요.

describe-user-hierarchy-structure

다음 코드 예시는 describe-user-hierarchy-structure의 사용 방법을 보여 줍니다.

AWS CLI

계층 구조에 대한 세부 정보를 표시하는 방법

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 구조 설정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeUserHierarchyStructure 섹션을 참조하세요.

describe-user

다음 코드 예시는 `describe-user`의 사용 방법을 보여 줍니다.

AWS CLI

사용자의 세부 정보를 표시하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeUser 섹션을 참조하세요.

get-contact-attributes

다음 코드 예시는 get-contact-attributes의 사용 방법을 보여 줍니다.

AWS CLI

연락처의 속성을 검색하는 방법

다음 get-contact-attributes 예제에서는 지정된 Amazon Connect Contact에 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 Contact 속성 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetContactAttributes 섹션을 참조하세요.

list-contact-flows

다음 코드 예시는 list-contact-flows의 사용 방법을 보여 줍니다.

AWS CLI

인스턴스의 고객 응대 흐름을 나열하는 방법

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 Contact 흐름 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListContactFlows 섹션 섹션을 참조하세요.

`list-hours-of-operations`

다음 코드 예시는 `list-hours-of-operations`의 사용 방법을 보여 줍니다.

AWS CLI

인스턴스에서 작업 시간을 나열하는 방법

다음 `list-hours-of-operations` 예제에서는 지정된 Amazon Connect 인스턴스의 운영 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListHoursOfOperations 섹션을 참조하세요.

list-phone-numbers

다음 코드 예시는 `list-phone-numbers`의 사용 방법을 보여 줍니다.

AWS CLI

인스턴스의 전화번호를 나열하는 방법

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 센터의 전화번호 설정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListPhoneNumbers 섹션 섹션을 참조하세요.

list-queues

다음 코드 예시는 `list-queues`의 사용 방법을 보여 줍니다.

AWS CLI

인스턴스의 대기열을 나열하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 ListQueues를 참조하세요.

list-routing-profiles

다음 코드 예시는 `list-routing-profiles`의 사용 방법을 보여 줍니다.

AWS CLI

인스턴스의 모든 라우팅 프로필을 나열하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 ListRoutingProfiles 섹션을 참조하세요.

list-security-profiles

다음 코드 예시는 `list-security-profiles`의 사용 방법을 보여 줍니다.

AWS CLI

인스턴스의 모든 보안 프로필을 나열하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 ListSecurityProfiles 섹션을 참조하세요.

list-user-hierarchy-groups

다음 코드 예시는 `list-user-hierarchy-groups`의 사용 방법을 보여 줍니다.

AWS CLI

인스턴스의 사용자 계층 그룹을 나열하는 방법

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 구조 설정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListUserHierarchyGroups 섹션을 참조하세요.

list-users

다음 코드 예시는 `list-users`의 사용 방법을 보여 줍니다.

AWS CLI

인스턴스의 사용자 계층 그룹을 나열하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 ListUsers를 참조하세요.

update-contact-attributes

다음 코드 예시는 update-contact-attributes의 사용 방법을 보여 줍니다.

AWS CLI

연락처의 속성을 업데이트하는 방법

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 Contact 속성 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateContactAttributes 섹션을 참조하세요.

update-user-hierarchy

다음 코드 예시는 update-user-hierarchy의 사용 방법을 보여 줍니다.

AWS CLI

사용자의 계층 구조를 업데이트하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateUserHierarchy 섹션을 참조하세요.

update-user-identity-info

다음 코드 예시는 update-user-identity-info의 사용 방법을 보여 줍니다.

AWS CLI

사용자의 자격 증명 정보를 업데이트하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateUserIdentityInfo 섹션을 참조하세요.

update-user-phone-config

다음 코드 예시는 update-user-phone-config의 사용 방법을 보여 줍니다.

AWS CLI

사용자의 전화 구성을 업데이트하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateUserPhoneConfig 섹션을 참조하세요.

update-user-routing-profile

다음 코드 예시는 update-user-routing-profile의 사용 방법을 보여 줍니다.

AWS CLI

사용자의 라우팅 프로필을 업데이트하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateUserRoutingProfile 섹션을 참조하세요.

`update-user-security-profiles`

다음 코드 예시는 `update-user-security-profiles`의 사용 방법을 보여 줍니다.

AWS CLI

사용자의 보안 프로필을 업데이트하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateUserSecurityProfiles 섹션을 참조하세요.

AWS CLI를 사용한 AWS Cost and Usage Report 예시

다음 코드 예시에서는 AWS Cost and Usage Report에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`delete-report-definition`

다음 코드 예시는 `delete-report-definition`의 사용 방법을 보여 줍니다.

AWS CLI

AWS 비용 및 사용 보고서를 삭제하는 방법

이 예제에서는 AWS 비용 및 사용 보고서를 삭제합니다.

명령:

```
aws cur --region us-east-1 delete-report-definition --report-name "ExampleReport"
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteReportDefinition을 참조하세요.

`describe-report-definitions`

다음 코드 예시는 `describe-report-definitions`의 사용 방법을 보여 줍니다.

AWS CLI

AWS 비용 및 사용 보고서 목록을 검색하는 방법

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"
      ]
  }
]
  }
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeReportDefinitions를 참조하세요.

put-report-definition

다음 코드 예시는 put-report-definition의 사용 방법을 보여 줍니다.

AWS CLI

AWS 비용 및 사용 보고서를 생성하는 방법

다음 put-report-definition 예제에서는 일일 AWS 비용 및 사용량 보고서를 생성하여 Amazon Redshift 또는 Amazon QuickSight에 업로드할 수 있습니다.

```
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"
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 PutReportDefinition을 참조하세요.

AWS CLI를 사용한 Cost Explorer Service 예제

다음 코드 예제에서는 Cost Explorer Service에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여 줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`get-cost-and-usage`

다음 코드 예시는 `get-cost-and-usage`의 사용 방법을 보여 줍니다.

AWS CLI

2017년 9월 한 달간 계정의 S3 사용량을 검색하는 방법

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.json`의 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"
                        }
                    }
                }
            ]
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetCostAndUsage를 참조하세요.

`get-dimension-values`

다음 코드 예시는 `get-dimension-values`의 사용 방법을 보여 줍니다.

AWS CLI

값이 “Elastic”인 SERVICE 차원에 대한 태그를 검색하는 방법

이 예제에서는 2017년 1월 01일부터 2017년 5월 18일까지의 값이 “Elastic”인 SERVICE차원에 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetDimensionValues를 참조하세요.

get-reservation-coverage

다음 코드 예시는 `get-reservation-coverage`의 사용 방법을 보여 줍니다.

AWS CLI

us-east-1 리전의 EC2 t2.nano 인스턴스에 대한 예약 범위를 검색하는 방법

이 예제에서는 2017년 7월~9월 동안 us-east-1 리전의 EC2 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetReservationCoverage를 참조하세요.

get-reservation-purchase-recommendation

다음 코드 예시는 `get-reservation-purchase-recommendation`의 사용 방법을 보여 줍니다.

AWS CLI

3년 기간의 부분 선불 EC2 RI에 대한 예약 권장 사항을 검색하는 방법

다음 `get-reservation-purchase-recommendation` 예제에서는 최근 60일간의 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetReservationPurchaseRecommendation을 참조하세요.

`get-reservation-utilization`

다음 코드 예시는 `get-reservation-utilization`의 사용 방법을 보여 줍니다.

AWS CLI

계정의 예약 사용률을 검색하는 방법

다음 `get-reservation-utilization` 예제에서는 계정에 대해 2018-03-01부터 2018-08-01까지 모든 t2.nano 인스턴스 유형에 대한 RI 사용률을 검색합니다.

```
aws ce get-reservation-utilization \
    --time-period Start=2018-03-01,End=2018-08-01 \
    --filter file://filters.json
```

`filters.json`의 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": []
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetReservationUtilization을 참조하세요.

get-tags

다음 코드 예시는 `get-tags`의 사용 방법을 보여 줍니다.

AWS CLI

비용 할당 태그의 키와 값을 검색하는 방법

이 예제에서는 키가 "Project"이고 값에 "secretProject"가 포함된 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetTags를 참조하세요.

AWS CLI를 사용한 Firehose 예제

다음 코드 예제에서는 Firehose에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여 줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`list-delivery-streams`

다음 코드 예시는 `list-delivery-streams`의 사용 방법을 보여 줍니다.

AWS CLI

사용 가능한 전송 스트림을 나열하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 ListDeliveryStreams를 참조하세요.

put-record-batch

다음 코드 예시는 `put-record-batch`의 사용 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.json`의 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.

- API 세부 정보는 AWS CLI 명령 참조의 PutRecordBatch를 참조하세요.

put-record

다음 코드 예시는 put-record의 사용 방법을 보여 줍니다.

AWS CLI

스트림에 레코드를 쓰는 방법

다음 put-record 예제에서는 하나의 스트림에 3개의 레코드를 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.

- API 세부 정보는 AWS CLI 명령 참조의 PutRecord를 참조하세요.

AWS CLI를 사용한 Amazon Data Lifecycle Manager 예제

다음 코드 예제에서는 Amazon Data Lifecycle Manager에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여 줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`create-default-role`

다음 코드 예제에서는 `create-default-role`의 사용 방법을 보여줍니다.

AWS CLI

Amazon DLM에 필요한 IAM 역할을 생성하려면

다음 `dlm create-default-role` 예제에서는 스냅샷 관리를 위한 AWSDataLifecycleManagerDefaultRole 기본 역할을 만듭니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateDefaultRole을 참조하세요.

`create-lifecycle-policy`

다음 코드 예제에서는 `create-lifecycle-policy`의 사용 방법을 보여줍니다.

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.json`의 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateLifecyclePolicy를 참조하세요.

delete-lifecycle-policy

다음 코드 예제에서는 delete-lifecycle-policy의 사용 방법을 보여줍니다.

AWS CLI

수명 주기 정책을 삭제하려면

다음 예제에서는 지정한 수명 주기 정책을 삭제합니다.

```
aws dlm delete-lifecycle-policy --policy-id policy-0123456789abcdef0
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteLifecyclePolicy를 참조하세요.

get-lifecycle-policies

다음 코드 예제에서는 `get-lifecycle-policies`의 사용 방법을 보여줍니다.

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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetLifecyclePolicies를 참조하세요.

get-lifecycle-policy

다음 코드 예제에서는 `get-lifecycle-policy`의 사용 방법을 보여줍니다.

AWS CLI

수명 주기 정책을 삭제하려면

다음 `get-lifecycle-policy` 예제에서는 지정된 정책의 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
                    }
                }
            ]
        }
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetLifecyclePolicy를 참조하세요.

update-lifecycle-policy

다음 코드 예제에서는 update-lifecycle-policy의 사용 방법을 보여줍니다.

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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateLifecyclePolicy를 참조하세요.

AWS CLI를 사용한 AWS Data Pipeline 예시

다음 코드 예시에서는 AWS Data Pipeline에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

activate-pipeline

다음 코드 예시는 activate-pipeline의 사용 방법을 보여 줍니다.

AWS CLI

파이프라인을 활성화하는 방법

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
```

- API 세부 정보는 AWS CLI 명령 참조의 ActivatePipeline을 참조하세요.

add-tags

다음 코드 예시는 add-tags의 사용 방법을 보여 줍니다.

AWS CLI

파이프라인에 태그를 추가하는 방법

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 명령에서 추가된 태그는 describe-pipelines의 출력에 다음과 같이 나타납니다.

```
{
    ...
        "tags": [
            {
                "value": "production",
                "key": "environment"
            },
            {
                "value": "sales",
                "key": "owner"
            }
        ]
    ...
}
```

- API 세부 정보는 AWS CLI 명령 참조의 AddTags를 참조하세요.

create-pipeline

다음 코드 예시는 create-pipeline의 사용 방법을 보여 줍니다.

AWS CLI

파이프라인을 생성하는 방법

이 예제에서는 파이프라인을 생성합니다.

```
aws datapipeline create-pipeline --name my-pipeline --unique-id my-pipeline-token
```

다음은 예 출력입니다.

```
{
    "pipelineId": "df-00627471SOVYZEXAMPLE"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreatePipeline을 참조하세요.

deactivate-pipeline

다음 코드 예시는 deactivate-pipeline의 사용 방법을 보여 줍니다.

AWS CLI

파이프라인을 비활성화하는 방법

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
```

- API 세부 정보는 AWS CLI 명령 참조의 DeactivatePipeline을 참조하세요.

delete-pipeline

다음 코드 예시는 delete-pipeline의 사용 방법을 보여 줍니다.

AWS CLI

파이프라인을 삭제하는 방법

이 예제에서는 지정된 파이프라인을 삭제합니다.

```
aws datapipeline delete-pipeline --pipeline-id df-00627471SOVYZEXAMPLE
```

- API 세부 정보는 AWS CLI 명령 참조의 DeletePipeline을 참조하세요.

describe-pipelines

다음 코드 예시는 describe-pipelines의 사용 방법을 보여 줍니다.

AWS CLI

파이프라인을 설명하는 방법

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": []
    }
  ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribePipelines를 참조하세요.

get-pipeline-definition

다음 코드 예시는 get-pipeline-definition의 사용 방법을 보여 줍니다.

AWS CLI

파이프라인 정의를 가져오는 방법

이 예제에서는 지정된 파이프라인의 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://amzn-s3-demo-bucket/",
```

```
      "myS3InputLoc": "s3://us-east-1.elasticmapreduce.samples/pig-apache-logs/
data",
      "myShellCmd": "grep -rc \"GET\" ${INPUT1_STAGING_DIR}/* >
 ${OUTPUT1_STAGING_DIR}/output.txt"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetPipelineDefinition을 참조하세요.

list-pipelines

다음 코드 예시는 `list-pipelines`의 사용 방법을 보여 줍니다.

AWS CLI

파이프라인을 나열하는 방법

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"
      }
  ]
```

```
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListPipelines를 참조하세요.

`list-runs`

다음 코드 예시는 `list-runs`의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 파이프라인 실행을 나열하는 방법

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 나열하는 방법

다음 `list-runs` 예제에서는 `--start-interval`을 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
```

- API 세부 정보는 AWS CLI 명령 참조의 ListRuns를 참조하세요.

put-pipeline-definition

다음 코드 예시는 put-pipeline-definition의 사용 방법을 보여 줍니다.

AWS CLI

파이프라인 정의를 업로드하는 방법

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": []
}
```

- API 세부 정보는 AWS CLI 명령 참조의 PutPipelineDefinition을 참조하세요.

remove-tags

다음 코드 예시는 remove-tags의 사용 방법을 보여 줍니다.

AWS CLI

파이프라인에서 태그를 제거하는 방법

이 예제에서는 지정된 파이프라인에서 지정된 태그를 제거합니다.

```
aws datapipeline remove-tags --pipeline-id df-00627471SOVYZEXAMPLE --tag-
keys environment
```

- API 세부 정보는 AWS CLI 명령 참조의 RemoveTags를 참조하세요.

AWS CLI를 사용한 DataSync 예제

다음 코드 예제에서는 DataSync에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

작업

update-location-azure-blob

다음 코드 예제에서는 `update-location-azure-blob`의 사용 방법을 보여줍니다.

AWS CLI

새 에이전트를 사용하여 전송 위치를 업데이트하려면

다음 `update-location-object-storage` 예제에서는 Microsoft Azure Blob Storage의 DataSync 위치를 새 에이전트로 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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateLocationAzureBlob을 참조하세요.

update-location-hdfs

다음 코드 예제에서는 update-location-hdfs의 사용 방법을 보여줍니다.

AWS CLI

새 에이전트를 사용하여 전송 위치를 업데이트하려면

다음 update-location-hdfs 예제에서는 DataSync HDFS 위치를 새 에이전트로 업데이트합니다. HDFS 클러스터에서 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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateLocationHdfs를 참조하세요.

update-location-nfs

다음 코드 예제에서는 update-location-nfs의 사용 방법을 보여줍니다.

AWS CLI

새 에이전트를 사용하여 전송 위치를 업데이트하려면

다음 update-location-nfs 예제에서는 DataSync NFS 위치를 새 에이전트로 업데이트합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateLocationNfs를 참조하세요.

update-location-object-storage

다음 코드 예제에서는 update-location-object-storage의 사용 방법을 보여줍니다.

AWS CLI

새 에이전트를 사용하여 전송 위치를 업데이트하려면

다음 update-location-object-storage 예제에서는 DataSync 객체 스토리지 위치를 새 에이전트로 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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateLocationObjectStorage를 참조하세요.

update-location-smb

다음 코드 예제에서는 update-location-smb의 사용 방법을 보여줍니다.

AWS CLI

새 에이전트를 사용하여 전송 위치를 업데이트하려면

다음 update-location-smb 예제에서는 DataSync SMB 위치를 새 에이전트로 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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateLocationSmb를 참조하세요.

AWS CLI를 사용한 DAX 예제

다음 코드 예제에서는 DAX에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`create-cluster`

다음 코드 예제에서는 `create-cluster`의 사용 방법을 보여줍니다.

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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateCluster를 참조하세요.

`create-parameter-group`

다음 코드 예제에서는 `create-parameter-group`의 사용 방법을 보여줍니다.

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 정보는 Amazon DynamoDB 개발자 안내서의 DAX 클러스터 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateParameterGroup을 참조하세요.

create-subnet-group

다음 코드 예제에서는 `create-subnet-group`의 사용 방법을 보여줍니다.

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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateSubnetGroup을 참조하세요.

decrease-replication-factor

다음 코드 예제에서는 `decrease-replication-factor`의 사용 방법을 보여줍니다.

AWS CLI

클러스터에서 하나 이상의 노드를 제거하려면

다음 `decrease-replication-factor` 예제에서는 지정된 DAX 클러스터의 노드 수를 1개로 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 정보는 Amazon DynamoDB 개발자 안내서의 DAX 클러스터 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DecreaseReplicationFactor를 참조하세요.

delete-cluster

다음 코드 예제에서는 `delete-cluster`의 사용 방법을 보여줍니다.

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 정보는 Amazon DynamoDB 개발자 안내서의 DAX 클러스터 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteCluster를 참조하세요.

delete-parameter-group

다음 코드 예제에서는 delete-parameter-group의 사용 방법을 보여줍니다.

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 정보는 Amazon DynamoDB 개발자 안내서의 DAX 클러스터 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteParameterGroup을 참조하세요.

delete-subnet-group

다음 코드 예제에서는 delete-subnet-group의 사용 방법을 보여줍니다.

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 정보는 Amazon DynamoDB 개발자 안내서의 DAX 클러스터 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteSubnetGroup을 참조하세요.

describe-clusters

다음 코드 예제에서는 describe-clusters의 사용 방법을 보여줍니다.

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 정보는 Amazon DynamoDB 개발자 안내서의 DAX 클러스터 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeClusters를 참조하세요.

describe-default-parameters

다음 코드 예제에서는 `describe-default-parameters`의 사용 방법을 보여줍니다.

AWS CLI

DAX의 기본 시스템 파라미터 정보를 반환하려면

다음 `describe-default-parameters` 예제에서는 DAX에 대한 기본 시스템 파라미터 정보를 표시합니다.

```
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 정보는 Amazon DynamoDB 개발자 안내서의 DAX 클러스터 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDefaultParameters를 참조하세요.

describe-events

다음 코드 예제에서는 describe-events의 사용 방법을 보여줍니다.

AWS CLI

DAX 클러스터 및 파라미터 그룹과 관련된 모든 이벤트를 반환하려면

다음 describe-events 예제에서는 DAX 클러스터 및 파라미터 그룹과 관련된 이벤트의 세부 정보를 표시합니다.

```
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 정보는 Amazon DynamoDB 개발자 안내서의 DAX 클러스터 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeEvents를 참조하세요.

describe-parameter-groups

다음 코드 예제에서는 `describe-parameter-groups`의 사용 방법을 보여줍니다.

AWS CLI

DAX에 정의된 파라미터 그룹을 설명하려면

다음 `describe-parameter-groups` 예제에서는 DAX에 정의된 파라미터 그룹에 대한 세부 정보를 검색합니다.

```
aws dax describe-parameter-groups
```

출력:

```
{
    "ParameterGroups": [
        {
            "ParameterGroupName": "default.dax1.0",
            "Description": "Default parameter group for dax1.0"
        }
    ]
}
```

자세한 정보는 Amazon DynamoDB 개발자 안내서의 DAX 클러스터 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeParameterGroups를 참조하세요.

describe-parameters

다음 코드 예제에서는 `describe-parameters`의 사용 방법을 보여줍니다.

AWS CLI

DB 파라미터 그룹의 파라미터를 설명하려면

다음 `describe-parameters` 예제에서는 지정된 DAX 파라미터 그룹에 정의된 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 정보는 Amazon DynamoDB 개발자 안내서의 DAX 클러스터 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeParameters를 참조하세요.

describe-subnet-groups

다음 코드 예제에서는 describe-subnet-groups의 사용 방법을 보여줍니다.

AWS CLI

DAX에 정의된 서브넷 그룹을 설명하려면

다음 describe-subnet-groups 예제에서는 DAX에 정의된 서브넷 그룹에 대한 세부 정보를 검색합니다.

```
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 정보는 Amazon DynamoDB 개발자 안내서의 DAX 클러스터 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeSubnetGroups를 참조하세요.

`increase-replication-factor`

다음 코드 예제에서는 `increase-replication-factor`의 사용 방법을 보여줍니다.

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 정보는 Amazon DynamoDB 개발자 안내서의 DAX 클러스터 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 IncreaseReplicationFactor를 참조하세요.

list-tags

다음 코드 예제에서는 `list-tags`의 사용 방법을 보여줍니다.

AWS CLI

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 정보는 Amazon DynamoDB 개발자 안내서의 DAX 클러스터 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ListTags](#)를 참조하세요.

tag-resource

다음 코드 예제에서는 `tag-resource`의 사용 방법을 보여줍니다.

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 정보는 Amazon DynamoDB 개발자 안내서의 [DAX 클러스터 관리](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [TagResource](#)를 참조하세요.

untag-resource

다음 코드 예제에서는 `untag-resource`의 사용 방법을 보여줍니다.

AWS CLI

DAX 리소스에서 태그를 제거하려면

다음 `untag-resource` 예제에서는 DAX 클러스터에서 키 이름이 지정된 태그를 제거합니다.

```
aws dax untag-resource  \
    --resource-name arn:aws:dax:us-west-2:123456789012:cache/daxcluster \
    --tag-keys="ClusterUsage"
```

출력:

```
{
    "Tags": []
}
```

자세한 정보는 Amazon DynamoDB 개발자 안내서의 DAX 클러스터 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

AWS CLI를 사용한 Detective 예제

다음 코드 예제에서는 Detective에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`accept-invitation`

다음 코드 예제에서는 `accept-invitation`의 사용 방법을 보여줍니다.

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.

- API 세부 정보는 AWS CLI 명령 참조의 AcceptInvitation을 참조하세요.

create-graph

다음 코드 예제에서는 `create-graph`의 사용 방법을 보여줍니다.

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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateGraph를 참조하세요.

create-members

다음 코드 예제에서는 `create-members`의 사용 방법을 보여줍니다.

AWS CLI

동작 그래프에 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. 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 그래프에 멤버 계정 초대 <https://docs.aws.amazon.com/detective/latest/adminguide/graph-admin-add-member-accounts.html>를 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 수신하지 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 그래프에 멤버 계정 초대 <https://docs.aws.amazon.com/detective/latest/adminguide/graph-admin-add-member-accounts.html>를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateMembers를 참조하세요.

delete-graph

다음 코드 예제에서는 delete-graph의 사용 방법을 보여줍니다.

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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteGraph를 참조하세요.

delete-members

다음 코드 예제에서는 delete-members의 사용 방법을 보여줍니다.

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 AWS개의 계정 ID를 제공합니다.

```
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 그래프에서 멤버 계정 제거 <https://docs.aws.amazon.com/detective/latest/adminguide/graph-admin-remove-member-accounts.html>를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteMembers를 참조하세요.

disassociate-membership

다음 코드 예제에서는 disassociate-membership의 사용 방법을 보여줍니다.

AWS CLI

동작 그래프에서 멤버십을 사임하려면

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 삭제 <https://docs.aws.amazon.com/detective/latest/adminguide/member-remove-self-from-graph.html>를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisassociateMembership을 참조하세요.

get-members

다음 코드 예제에서는 get-members의 사용 방법을 보여줍니다.

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 AWS개의 계정 ID를 제공합니다.

```
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 그래프에서 계정 목록 보기 <https://docs.aws.amazon.com/detective/latest/adminguide/graph-admin-view-accounts.html>를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetMembers를 참조하세요.

list-graphs

다음 코드 예제에서는 `list-graphs`의 사용 방법을 보여줍니다.

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
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListGraphs를 참조하세요.

list-invitations

다음 코드 예제에서는 `list-invitations`의 사용 방법을 보여줍니다.

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.

- API 세부 정보는 AWS CLI 명령 참조의 ListInvitations를 참조하세요.

list-members

다음 코드 예제에서는 list-members의 사용 방법을 보여줍니다.

AWS CLI

동작 그래프에 멤버 계정을 나열하려면

다음 list-members 예제에서는 동작 그래프 arn:aws:detective:us-east-1:111122223333:graph:123412341234에 대해 초대되고 활성화된 멤버 계정을 검색합니다. 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListMembers를 참조하세요.

`list-tags-for-resource`

다음 코드 예제에서는 `list-tags-for-resource`의 사용 방법을 보여줍니다.

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.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource를 참조하세요.

reject-invitation

다음 코드 예제에서는 reject-invitation의 사용 방법을 보여줍니다.

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 대한 응답 <https://docs.aws.amazon.com/detective/latest/adminguide/member-invitation-response.html>을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RejectInvitation을 참조하세요.

tag-resource

다음 코드 예제에서는 tag-resource의 사용 방법을 보여줍니다.

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 관리]를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [TagResource]를 참조하세요.

`untag-resource`

다음 코드 예제에서는 `untag-resource`의 사용 방법을 보여줍니다.

AWS CLI

리소스에서 태그를 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 관리]를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [UntagResource]를 참조하세요.

AWS CLI를 사용한 Device Farm 예제

다음 코드 예제에서는 Device Farm에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여 줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

create-device-pool

다음 코드 예제에서는 create-device-pool의 사용 방법을 보여줍니다.

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 list-projects 출력에서 프로젝트 ARN을 가져올 수 있습니다. 이 device-pool-rules.json 파일은 현재 폴더에 있는 디바이스 플랫폼을 지정하는 JSON 문서입니다.

```
[
    {
        "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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateDevicePool을 참조하세요.

`create-project`

다음 코드 예제에서는 `create-project`의 사용 방법을 보여줍니다.

AWS CLI

프로젝트를 생성하려면

다음 명령은 `my-project`라는 새 프로젝트를 생성합니다.

```
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
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateProject를 참조하세요.

`create-upload`

다음 코드 예제에서는 `create-upload`의 사용 방법을 보여줍니다.

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-projects 출력에서 프로젝트 ARN을 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"
    }
}
```

출력에 서명된 URL을 사용하여 Device Farm에 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"
```

- API 세부 정보는 AWS CLI 명령 참조의 [CreateUpload](https://docs.aws.amazon.com/goto/aws-cli/devicefarm-2015-06-23/CreateUpload)를 참조하세요.

`get-upload`

다음 코드 예제에서는 `get-upload`의 사용 방법을 보여줍니다.

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"
```

`create-upload`의 출력에서 업로드 ARN을 가져올 수 있습니다.

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\"}"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [GetUpload](https://docs.aws.amazon.com/goto/aws-cli/devicefarm-2015-06-23/GetUpload)를 참조하세요.

`list-projects`

다음 코드 예제에서는 `list-projects`의 사용 방법을 보여줍니다.

AWS CLI

프로젝트를 모두 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
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListProjects를 참조하세요.

AWS CLI를 사용한 AWS Direct Connect 예시

다음 코드 예시에서는 AWS Direct Connect에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

accept-direct-connect-gateway-association-proposal

다음 코드 예시에서는 accept-direct-connect-gateway-association-proposal 코드를 사용하는 방법을 보여줍니다.

AWS CLI

게이트웨이 연결 제안 수락

다음 accept-direct-connect-gateway-association-proposal은 지정된 제안을 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 Creating a Hosted Transit Virtual Interface를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AcceptDirectConnectGatewayAssociationProposal 섹션을 참조하세요.

allocate-connection-on-interconnect

다음 코드 예시에서는 `allocate-connection-on-interconnect` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

인터커넥트에서 호스팅 연결 생성

다음 `allocate-connection-on-interconnect` 명령은 인터커넥트에서 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 AllocateConnectionOnInterconnect 섹션을 참조하세요.

allocate-hosted-connection

다음 코드 예시에서는 allocate-hosted-connection 코드를 사용하는 방법을 보여줍니다.

AWS CLI

인터커넥트에서 호스팅 연결 생성

다음 allocate-hosted-connection 예시에서는 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 AllocateHostedConnection 섹션을 참조하세요.

allocate-private-virtual-interface

다음 코드 예시에서는 allocate-private-virtual-interface 코드를 사용하는 방법을 보여줍니다.

AWS CLI

프라이빗 가상 인터페이스 프로비저닝

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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 AllocatePrivateVirtualInterface 섹션을 참조하세요.

`allocate-public-virtual-interface`

다음 코드 예시에서는 `allocate-public-virtual-interface` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

가상 퍼블릭 인터페이스 프로비저닝

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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [AllocatePublicVirtualInterface](https://docs.aws.amazon.com/goto/aws-cli/directconnect-2012-10-25/AllocatePublicVirtualInterface) 섹션을 참조하세요.

allocate-transit-virtual-interface

다음 코드 예시에서는 allocate-transit-virtual-interface 코드를 사용하는 방법을 보여줍니다.

AWS CLI

지정된 AWS 계정에서 소유할 전송 가상 인터페이스를 프로비저닝하는 방법

다음 allocate-transit-virtual-interface 예시에서는 지정된 계정에 대한 전송 가상화 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 Creating a Hosted Transit Virtual Interface를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AllocateTransitVirtualInterface 섹션을 참조하세요.

associate-connection-with-lag

다음 코드 예시에서는 `associate-connection-with-lag` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

연결을 LAG에 연결

다음 예시에서는 지정된 연결을 지정된 LAG와 연결합니다.

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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 AssociateConnectionWithLag 섹션을 참조하세요.

associate-hosted-connection

다음 코드 예시에서는 associate-hosted-connection 코드를 사용하는 방법을 보여줍니다.

AWS CLI

호스팅 연결을 LAG에 연결

다음 예시에서는 지정된 호스팅 연결을 지정된 LAG와 연결합니다.

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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 AssociateHostedConnection 섹션을 참조하세요.

associate-virtual-interface

다음 코드 예시에서는 associate-virtual-interface 코드를 사용하는 방법을 보여줍니다.

AWS CLI

가상 인터페이스를 연결에 연결하는 방법

다음 예시에서는 지정된 가상 인터페이스를 지정된 LAG와 연결합니다. 또는 가상 인터페이스를 연결에 연결하려면 --connection-id에 대한 AWS Direct Connect 연결의 ID를 지정합니다(예: 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [AssociateVirtualInterface](#) 섹션을 참조하세요.

confirm-connection

다음 코드 예시에서는 `confirm-connection` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

인터커넥트에서 호스팅 연결의 생성 확인

다음 `confirm-connection` 명령은 인터커넥트에서 호스팅 연결 생성을 확인합니다.

```
aws directconnect confirm-connection --connection-id dxcon-fg2wi7hy
```

출력:

```
{
    "connectionState": "pending"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ConfirmConnection 섹션을 참조하세요.

`confirm-private-virtual-interface`

다음 코드 예시에서는 `confirm-private-virtual-interface` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

프라이빗 가상 인터페이스의 소유권을 수락하는 방법

다음 `confirm-private-virtual-interface` 명령은 다른 고객이 만든 프라이빗 가상 인터페이스의 소유권을 허용합니다.

```
aws directconnect confirm-private-virtual-interface --virtual-interface-id dxvif-fgy8orxu --virtual-gateway-id vgw-e4a47df9
```

출력:

```
{
    "virtualInterfaceState": "pending"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ConfirmPrivateVirtualInterface 섹션을 참조하세요.

confirm-public-virtual-interface

다음 코드 예시에서는 confirm-public-virtual-interface 코드를 사용하는 방법을 보여줍니다.

AWS CLI

퍼블릭 가상 인터페이스의 소유권을 수락하는 방법

다음 confirm-public-virtual-interface 명령은 다른 고객이 만든 퍼블릭 가상 인터페이스의 소유권을 허용합니다.

```
aws directconnect confirm-public-virtual-interface --virtual-interface-id dxvif-fg9xo9vp
```

출력:

```
{
    "virtualInterfaceState": "verifying"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ConfirmPublicVirtualInterface 섹션을 참조하세요.

confirm-transit-virtual-interface

다음 코드 예시에서는 confirm-transit-virtual-interface 코드를 사용하는 방법을 보여줍니다.

AWS CLI

전송 가상 인터페이스의 소유권을 수락하는 방법

다음 confirm-transit-virtual-interface는 다른 고객이 만든 전송 가상 인터페이스의 소유권을 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 Accepting a Hosted Virtual Interface를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ConfirmTransitVirtualInterface 섹션을 참조하세요.

create-bgp-peer

다음 코드 예시에서는 `create-bgp-peer` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

IPv6 BGP 피어링 세션을 생성하는 방법

다음 예시에서는 프라이빗 가상 인터페이스 `dxvif-fg1vuj3d`에서 IPv6 BGP 피어링 세션을 생성합니다. 피어 IPv6 주소는 Amazon에서 자동으로 할당합니다.

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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateBgpPeer 섹션을 참조하세요.

`create-connection`

다음 코드 예시에서는 `create-connection` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

네트워크에서 AWS Direct Connect 위치로 연결을 생성하는 방법

다음 create-connection 명령은 네트워크에서 AWS Direct Connect 위치로의 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateConnection 섹션을 참조하세요.

create-direct-connect-gateway-association-proposal

다음 코드 예시에서는 create-direct-connect-gateway-association-proposal 코드를 사용하는 방법을 보여줍니다.

AWS CLI

지정된 전송 게이트웨이를 지정된 Direct Connect 게이트웨이와 연결하는 제안 생성

다음 create-direct-connect-gateway-association-proposal 예시에서는 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 Creating a Transit Gateway Association Proposal을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateDirectConnectGatewayAssociationProposal 섹션을 참조하세요.

`create-direct-connect-gateway-association`

다음 코드 예시에서는 `create-direct-connect-gateway-association` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

가상 프라이빗 게이트웨이를 Direct Connect 게이트웨이와 연결

다음 예시에서는 가상 프라이빗 게이트웨이 `vgw-6efe725e`를 Direct Connect 게이트웨이 `5f294f92-bafb-4011-916d-9b0bexample`와 연결합니다. 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"
  }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [CreateDirectConnectGatewayAssociation](https://docs.aws.amazon.com/goto/aws-cli/directconnect-2012-10-25/CreateDirectConnectGatewayAssociation) 섹션을 참조하세요.

`create-direct-connect-gateway`

다음 코드 예시에서는 `create-direct-connect-gateway` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Direct Connect 게이트웨이 생성

다음 예시에서는 이름이 `DxGateway1`인 Direct Connect 게이트웨이를 생성합니다.

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"
  }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateDirectConnectGateway 섹션을 참조하세요.

create-interconnect

다음 코드 예시에서는 `create-interconnect` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

파트너의 네트워크와 간의 상호 연결을 생성하는 방법 AWS

다음 `create-interconnect` 예시에서는 AWS Direct Connect 파트너의 네트워크와 특정 AWS Direct Connect 위치 간에 새 인터커넥트를 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의CreateInterconnect 섹션을 참조하세요.

create-lag

다음 코드 예시에서는 `create-lag` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

새 연결을 사용하여 LAG 생성

다음 예시에서는 LAG를 생성하고 대역폭이 1Gbps인 LAG에 대한 두 개의 새 AWS Direct Connect 연결을 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 사용하여 LAG 생성

다음 예시에서는 계정의 기존 연결에서 LAG를 생성하고 기존 연결과 동일한 대역폭 및 위치를 가진 LAG에 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"
```

```
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateLag 섹션을 참조하세요.

`create-private-virtual-interface`

다음 코드 예시에서는 `create-private-virtual-interface` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

가상 프라이빗 인터페이스 생성

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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [CreatePrivateVirtualInterface](#) 섹션을 참조하세요.

create-public-virtual-interface

다음 코드 예시에서는 `create-public-virtual-interface` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

가상 퍼블릭 인터페이스 생성

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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreatePublicVirtualInterface 섹션을 참조하세요.

`create-transit-virtual-interface`

다음 코드 예시에서는 `create-transit-virtual-interface` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

호스팅되는 전송 가상 인터페이스 생성

다음 `create-transit-virtual-interface` 예시에서는 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 AWS Direct Connect 사용 설명서의 Creating a Transit Virtual Interface to the Direct Connect Gateway를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateTransitVirtualInterface 섹션을 참조하세요.

delete-bgp-peer

다음 코드 예시에서는 delete-bgp-peer 코드를 사용하는 방법을 보여줍니다.

AWS CLI

가상 인터페이스에서 BGP 피어를 삭제하는 방법

다음 예시에서는 가상 인터페이스 dxvif-fg1vuj3d에서 IPv6 BGP 피어를 삭제합니다.

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"
  }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteBgpPeer 섹션을 참조하세요.

delete-connection

다음 코드 예시에서는 delete-connection 코드를 사용하는 방법을 보여줍니다.

AWS CLI

연결 삭제

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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteConnection 섹션을 참조하세요.

`delete-direct-connect-gateway-association`

다음 코드 예시에서는 `delete-direct-connect-gateway-association` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Direct Connect 게이트웨이 연결 삭제

다음 `delete-direct-connect-gateway-association` 예시에서는 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 Associating and Disassociating Transit Gateways를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteDirectConnectGatewayAssociation 섹션을 참조하세요.

delete-direct-connect-gateway

다음 코드 예시에서는 `delete-direct-connect-gateway` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Direct Connect 게이트웨이 삭제

다음 예시에서는 Direct Connect 게이트웨이 `5f294f92-bafb-4011-916d-9b0bexample`을 삭제합니다.

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"
  }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteDirectConnectGateway 섹션을 참조하세요.

delete-interconnect

다음 코드 예시에서는 `delete-interconnect` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

상호 연결을 삭제하는 방법

다음 delete-interconnect 명령은 지정된 인터커넥트를 삭제합니다.

```
aws directconnect delete-interconnect --interconnect-id dxcon-fgktov66
```

출력:

```
{
    "interconnectState": "deleted"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteInterconnect 섹션을 참조하세요.

delete-lag

다음 코드 예시에서는 delete-lag 코드를 사용하는 방법을 보여줍니다.

AWS CLI

LAG 삭제

다음 예시에서는 지정한 LAG를 삭제합니다.

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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteLag 섹션을 참조하세요.

delete-virtual-interface

다음 코드 예시에서는 delete-virtual-interface 코드를 사용하는 방법을 보여줍니다.

AWS CLI

가상 인터페이스 삭제

다음 delete-virtual-interface 명령은 지정된 가상 인터페이스를 삭제합니다.

```
aws directconnect delete-virtual-interface --virtual-interface-id dxvif-ffhhk74f
```

출력:

```
{
    "virtualInterfaceState": "deleting"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteVirtualInterface 섹션을 참조하세요.

describe-connection-loa

다음 코드 예시에서는 describe-connection-loa 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Linux 또는 Mac OS X를 사용하여 연결에 대한 LOA-CFA를 설명하는 방법

다음 예시에서는 dxcon-fh6ayh1d 연결에 대한 LOA-CFA를 설명합니다. LOA-CFA의 콘텐츠는 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 사용하여 연결에 대한 LOA-CFA를 설명하는 방법

이전 예시에서는 base64 유틸리티를 사용하여 출력을 디코딩해야 합니다. Windows 컴퓨터에서는 대신 certutil을 사용할 수 있습니다. 다음 예시에서 첫 번째 명령은 dxcon-fh6ayh1d 연결용 LOA-CFA를 설명하고 --output 및 --query 파라미터를 사용하여 출력을 제어하고 loaContent 구조의 내용을 myLoaCfa.base64 파일로 추출합니다. 두 번째 명령은 certutil 유틸리티를 사용하여 파일을 디코딩하고 출력을 PDF 파일로 내보냅니다.

```
aws directconnect describe-connection-loa --connection-id dxcon-fh6ayh1d --
output text --query loa.loaContent > myLoaCfa.base64
```

```
certutil -decode myLoaCfa.base64 myLoaCfa.pdf
```

AWS CLI 출력 제어에 대한 자세한 내용은 AWS Command Line Interface 사용 설명서의 AWS Command Line Interface에서 명령 출력 제어를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeConnectionLoa 섹션을 참조하세요.

describe-connections-on-interconnect

다음 코드 예시에서는 describe-connections-on-interconnect 코드를 사용하는 방법을 보여줍니다.

AWS CLI

상호 연결의 연결을 나열하는 방법

다음 describe-connections-on-interconnect 명령은 지정된 인터커넥트에서 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"
            }
        ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeConnectionsOnInterconnect 섹션을 참조하세요.

describe-connections

다음 코드 예시에서는 describe-connections 코드를 사용하는 방법을 보여줍니다.

AWS CLI

현재 리전의 모든 연결을 나열하는 방법

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"
      }
  ]
```

```
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeConnections 섹션을 참조하세요.

describe-direct-connect-gateway-association-proposals

다음 코드 예시에서는 describe-direct-connect-gateway-association-proposals 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Direct Connect 게이트웨이 연결 제안 설명

다음 describe-direct-connect-gateway-association-proposals 예시에서는 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 Associating and Disassociating Transit Gateways를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDirectConnectGatewayAssociationProposals 섹션을 참조하세요.

describe-direct-connect-gateway-associations

다음 코드 예시에서는 `describe-direct-connect-gateway-associations` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Direct Connect 게이트웨이 연결 설명

다음 예시에서는 Direct Connect 게이트웨이 `5f294f92-bafb-4011-916d-9b0bexample`과의 모든 연결을 설명합니다.

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"
      }
  ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDirectConnectGatewayAssociations 섹션을 참조하세요.

`describe-direct-connect-gateway-attachments`

다음 코드 예시에서는 `describe-direct-connect-gateway-attachments` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Direct Connect 게이트웨이 연결 설명

다음 예시에서는 Direct Connect 게이트웨이 `5f294f92-bafb-4011-916d-9b0bexample`에 연결된 가상 인터페이스를 설명합니다.

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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDirectConnectGatewayAttachments 섹션을 참조하세요.

`describe-direct-connect-gateways`

다음 코드 예시에서는 `describe-direct-connect-gateways` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Direct Connect 게이트웨이를 설명하는 방법

다음 예시에서는 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"
      }
  ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDirectConnectGateways 섹션을 참조하세요.

describe-hosted-connections

다음 코드 예시에서는 `describe-hosted-connections` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

상호 연결의 연결을 나열하는 방법

다음 예시에서는 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeHostedConnections 섹션을 참조하세요.

`describe-interconnect-loa`

다음 코드 예시에서는 `describe-interconnect-loa` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Linux 또는 Mac OS X를 사용하여 상호 연결에 대한 LOA-CFA를 설명하는 방법

다음 예시에서는 인터커넥트 `dxcon-fh6ayh1d`에 대한 LOA-CFA를 설명합니다. LOA-CFA의 콘텐츠는 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 사용하여 상호 연결에 대한 LOA-CFA 설명

이전 예시에서는 `base64` 유틸리티를 사용하여 출력을 디코딩해야 합니다. Windows 컴퓨터에서는 대신 `certutil`을 사용할 수 있습니다. 다음 예시에서 첫 번째 명령은 `dxcon-fh6ayh1d` 인터커넥트용 LOA-CFA를 설명하고 `--output` 및 `--query` 파라미터를 사용하여 출력을 제어하고

loaContent 구조의 내용을 myLoaCfa.base64 파일로 추출합니다. 두 번째 명령은 certutil 유틸리티를 사용하여 파일을 디코딩하고 출력을 PDF 파일로 내보냅니다.

```
aws directconnect describe-interconnect-loa --interconnect-id dxcon-fh6ayh1d --output text --query loa.loaContent > myLoaCfa.base64
```

```
certutil -decode myLoaCfa.base64 myLoaCfa.pdf
```

AWS CLI 출력 제어에 대한 자세한 내용은 AWS Command Line Interface 사용 설명서의 AWS Command Line Interface에서 명령 출력 제어를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeInterconnectLoa 섹션을 참조하세요.

describe-interconnects

다음 코드 예시에서는 describe-interconnects 코드를 사용하는 방법을 보여줍니다.

AWS CLI

상호 연결을 나열하는 방법

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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeInterconnects 섹션을 참조하세요.

describe-lags

다음 코드 예시에서는 describe-lags 코드를 사용하는 방법을 보여줍니다.

AWS CLI

LAGs 설명하는 방법

다음 명령은 현재 리전에 대한 모든 LAG를 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"
    }
  ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeLags 섹션을 참조하세요.

describe-loa

다음 코드 예시에서는 describe-loa 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Linux 또는 Mac OS X를 사용하여 연결에 대한 LOA-CFA를 설명하는 방법

다음 예시에서는 dxcon-fh6ayh1d 연결에 대한 LOA-CFA를 설명합니다. LOA-CFA의 콘텐츠는 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 사용하여 연결에 대한 LOA-CFA를 설명하는 방법

이전 예시에서는 base64 유틸리티를 사용하여 출력을 디코딩해야 합니다. Windows 컴퓨터에서는 대신 certutil을 사용할 수 있습니다. 다음 예시에서 첫 번째 명령은 dxcon-fh6ayh1d 연결용 LOA-CFA를 설명하고 --output 및 --query 파라미터를 사용하여 출력을 제어하고 loaContent 구조의 내용을 myLoaCfa.base64 파일로 추출합니다. 두 번째 명령은 certutil 유틸리티를 사용하여 파일을 디코딩하고 출력을 PDF 파일로 내보냅니다.

```
aws directconnect describe-loa --connection-id dxcon-fh6ayh1d --output text --
query loa.loaContent > myLoaCfa.base64
```

```
certutil -decode myLoaCfa.base64 myLoaCfa.pdf
```

AWS CLI 출력 제어에 대한 자세한 내용은 AWS Command Line Interface 사용 설명서의 AWS Command Line Interface에서 명령 출력 제어를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeLoa 섹션을 참조하세요.

describe-locations

다음 코드 예시에서는 describe-locations 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS Direct Connect 파트너 및 위치를 나열하는 방법

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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeLocations 섹션을 참조하세요.

describe-tags

다음 코드 예시에서는 describe-tags 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS Direct Connect 리소스에 대한 태그를 설명하는 방법

다음 명령은 연결 dxcon-abcabc12에 대한 태그를 설명합니다.

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"
              }
          ]
      }
  ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeTags 섹션을 참조하세요.

`describe-virtual-gateways`

다음 코드 예시에서는 `describe-virtual-gateways` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

가상 프라이빗 게이트웨이 나열

다음 `describe-virtual-gateways` 명령은 AWS 계정에서 소유한 가상 프라이빗 게이트웨이를 나열합니다.

```
aws directconnect describe-virtual-gateways
```

출력:

```
{
    "virtualGateways": [
        {
            "virtualGatewayId": "vgw-aba37db6",
            "virtualGatewayState": "available"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeVirtualGateways 섹션을 참조하세요.

describe-virtual-interfaces

다음 코드 예시에서는 `describe-virtual-interfaces` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

모든 가상 인터페이스를 나열하는 방법

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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeVirtualInterfaces 섹션을 참조하세요.

`disassociate-connection-from-lag`

다음 코드 예시에서는 `disassociate-connection-from-lag` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

연결을 LAG에서 연결 해제

다음 예시에서는 지정된 연결을 지정된 LAG와의 연결을 해제합니다.

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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DisassociateConnectionFromLag 섹션을 참조하세요.

`tag-resource`

다음 코드 예시에서는 `tag-resource` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS Direct Connect 리소스에 태그를 추가하는 방법

다음 명령은 키가 `Name`이고 값이 `VAConnection`인 태그를 연결 `dxcon-abcabc12`에 추가합니다. 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"
```

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

untag-resource

다음 코드 예시에서는 `untag-resource` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS Direct Connect 리소스에서 태그를 제거하는 방법

다음 명령은 연결 `dxcon-abcabc12`에서 `Name` 키가 있는 태그를 제거합니다. 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
```

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

update-direct-connect-gateway-association

다음 코드 예시에서는 `update-direct-connect-gateway-association` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Direct Connect 게이트웨이 연결의 지정된 속성 업데이트

다음 `update-direct-connect-gateway-association` 예시에서는 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 AWS Direct Connect 사용 설명서의 Working with Direct Connect Gateways를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateDirectConnectGatewayAssociation 섹션을 참조하세요.

update-lag

다음 코드 예시에서는 `update-lag` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

LAG 업데이트

다음 예시에서는 지정된 LAG의 이름을 변경합니다.

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"
}
```


- API 세부 정보는 AWS CLI 명령 참조의 [UpdateLag](#) 섹션을 참조하세요.

`update-virtual-interface-attributes`

다음 코드 예시에서는 `update-virtual-interface-attributes` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

가상 인터페이스의 MTU를 업데이트하는 방법

다음 `update-virtual-interface-attributes` 예시에서는 지정된 가상 인터페이스의 MTU를 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 Setting Network MTU for Private Virtual Interfaces or Transit Virtual Interfaces를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateVirtualInterfaceAttributes 섹션을 참조하세요.

AWS CLI를 사용한 AWS Directory Service 예시

다음 코드 예시에서는 AWS Directory Service에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

describe-directories

다음 코드 예제에서는 describe-directories의 사용 방법을 보여줍니다.

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"
                        ]
                    }
                }
            }
        ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDirectories를 참조하세요.

describe-trusts

다음 코드 예제에서는 describe-trusts의 사용 방법을 보여줍니다.

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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeTrusts를 참조하세요.

AWS Directory Service를 사용한 AWS CLI 데이터 예제

다음 코드 예제에서는 AWS Directory Service Data와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

작업

`add-group-member`

다음 코드 예시는 `add-group-member`의 사용 방법을 보여 줍니다.

AWS CLI

디렉터리에 그룹 멤버를 추가하려면

다음 `add-group-member` 예제에서는 지정된 디렉터리의 지정된 그룹에 지정된 사용자를 추가합니다.

```
aws ds-data add-group-member \
    --directory-id d-1234567890 \
    --group-name 'sales' \
    --member-name 'john.doe'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Directory Service 관리 안내서의 그룹에 AWS 관리형 Microsoft AD 멤버 추가 또는 제거 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AddGroupMember 섹션을 참조하세요.

create-group

다음 코드 예시는 create-group의 사용 방법을 보여 줍니다.

AWS CLI

사용 가능한 위젯을 나열하는 방법

다음 create-group 예제에서는 지정된 디렉터리에서 그룹을 만듭니다.

```
aws ds-data create-group \
    --directory-id d-1234567890 \
    --sam-account-name "sales"
```

출력:

```
{
    "DirectoryId": "d-1234567890",
    "SAMAccountName": "sales",
    "SID": "S-1-2-34-5567891234-5678912345-67891234567-8912"
}
```

자세한 내용은 AWS Directory Service 관리 안내서의 AWS 관리형 Microsoft AD 그룹 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateGroup을 참조하세요.

create-user

다음 코드 예시는 create-user의 사용 방법을 보여 줍니다.

AWS CLI

사용자 생성

다음 create-user 예제에서는 지정된 디렉터리에서 사용자를 생성합니다.

```
aws ds-data create-user \
    --directory-id d-1234567890 \
    --sam-account-name 'john.doe'
```

출력:

```
{
    "DirectoryId": "d-1234567890",
    "SAMAccountName": "john.doe",
    "SID": "S-1-2-34-5567891234-5678912345-67891234567-8912"
}
```

자세한 내용은 AWS Directory Service 관리 안내서의 AWS 관리형 Microsoft AD 사용자 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateUser를 참조하세요.

delete-group

다음 코드 예시는 delete-group의 사용 방법을 보여 줍니다.

AWS CLI

그룹 삭제

다음 delete-group 예제에서는 지정된 디렉터리에서 지정된 태그를 삭제합니다.

```
aws ds-data delete-group \
    --directory-id d-1234567890 \
    --sam-account-name 'sales'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Directory Service 관리 안내서의 AWS 관리형 Microsoft AD 그룹 삭제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteGroup을 참조하세요.

delete-user

다음 코드 예시는 delete-user의 사용 방법을 보여 줍니다.

AWS CLI

사용자 삭제

다음 delete-user 예제에서는 지정된 디렉터리에서 지정된 사용자를 삭제합니다.

```
aws ds-data delete-user \
    --directory-id d-1234567890 \
    --sam-account-name 'john.doe'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Directory Service 관리 안내서의 AWS 관리형 Microsoft AD 사용자 삭제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteUser를 참조하세요.

describe-group

다음 코드 예시는 describe-group의 사용 방법을 보여 줍니다.

AWS CLI

그룹의 세부 정보를 나열하려면

다음 describe-group 예제에서는 지정된 디렉터리에서 지정된 그룹의 정보를 가져옵니다.

```
aws ds-data describe-group \
    --directory-id d-1234567890 \
    --sam-account-name 'sales'
```

출력:

```
{
    "DirectoryId": "d-1234567890",
    "DistinguishedName": "CN=sales,OU=Users,OU=CORP,DC=corp,DC=example,DC=com",
    "GroupScope": "Global",
    "GroupType": "Security",
    "Realm": "corp.example.com",
    "SAMAccountName": "sales",
    "SID": "S-1-2-34-5567891234-5678912345-67891234567-8912"
}
```

자세한 내용은 AWS Directory Service 관리 안내서의 AWS 관리형 Microsoft AD 그룹 세부 정보 보기 및 업데이트 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeGroup 섹션을 참조하세요.

describe-user

다음 코드 예시는 describe-user의 사용 방법을 보여 줍니다.

AWS CLI

사용자에 대한 정보를 나열하려면

다음 describe-user 예제에서는 지정된 디렉터리에서 지정된 사용자의 정보를 가져옵니다.

```
aws ds-data describe-user command-name \
    --directory-id d-1234567890 \
    --sam-account-name 'john.doe'
```

출력:

```
{
    "DirectoryId": "d-1234567890",
    "DistinguishedName": "CN=john.doe,OU=Users,OU=CORP,DC=corp,DC=example,DC=com",
    "Enabled": false,
    "Realm": "corp.example.com",
    "SAMAccountName": "john.doe",
    "SID": "S-1-2-34-5678901234-5678901234-5678910123-4567",
    "UserPrincipalName": "john.doe@CORP.EXAMPLE.COM"
}
```

자세한 내용은 AWS Directory Service 관리 안내서의 AWS 관리형 Microsoft AD 사용자 보기 및 업데이트 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeUser 섹션을 참조하세요.

disable-directory-data-access

다음 코드 예시는 disable-directory-data-access의 사용 방법을 보여 줍니다.

AWS CLI

디렉터리에 대한 Directory Service Data API를 비활성화하려면

다음 `disable-directory-data-access` 예제에서는 지정된 디렉터리에 대한 Directory Service Data API를 비활성화합니다.

```
aws ds disable-directory-data-access \
    --directory-id d-1234567890
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Directory Service 관리 안내서의 사용자 및 그룹 관리 또는 AWS Directory Service Data 활성화 또는 비활성화를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisableDirectoryDataAccess를 참조하세요.

`disable-user`

다음 코드 예시는 `disable-user`의 사용 방법을 보여 줍니다.

AWS CLI

사용자 비활성화

다음 `disable-user` 예제에서는 지정된 디렉터리에서 지정된 사용자를 비활성화합니다.

```
aws ds-data disable-user \
    --directory-id d-1234567890 \
    --sam-account-name 'john.doe'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Directory Service 관리 안내서의 AWS 관리형 Microsoft AD 사용자 비활성화를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisableUser를 참조하세요.

`enable-directory-data-access`

다음 코드 예시는 `enable-directory-data-access`의 사용 방법을 보여 줍니다.

AWS CLI

디렉터리에 대해 Directory Service Data API를 활성화하려면

다음 `enable-directory-data-access` 예제에서는 지정된 디렉터리에 대해 Directory Service Data API를 활성화합니다.

```
aws ds enable-directory-data-access \
    --directory-id d-1234567890
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Directory Service 관리 안내서의 사용자 및 그룹 관리 또는 AWS Directory Service Data 활성화 또는 비활성화를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 EnableDirectoryDataAccess를 참조하세요.

`list-group-members`

다음 코드 예시는 `list-group-members`의 사용 방법을 보여 줍니다.

AWS CLI

디렉터리의 그룹 멤버를 나열하려면

다음 `list-group-members` 예제에서는 지정된 디렉터리의 지정된 그룹에 대한 그룹 멤버를 나열합니다.

```
aws ds-data list-group-members \
    --directory-id d-1234567890 \
    --sam-account-name 'sales'
```

출력:

```
{
    "Members": [
        {
            "MemberType": "USER",
            "SAMAccountName": "Jane Doe",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4568"
        },
        {
            "MemberType": "USER",
            "SAMAccountName": "John Doe",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4569"
```

```
        }
    ],
    "DirectoryId": "d-1234567890",
    "MemberRealm": "corp.example.com",
    "Realm": "corp.example.com"
}
```

자세한 내용은 AWS Directory Service 관리 안내서의 AWS 관리형 Microsoft AD 그룹 세부 정보 보기 및 업데이트 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListGroupMembers 섹션을 참조하세요.

list-groups-for-member

다음 코드 예시는 `list-groups-for-member`의 사용 방법을 보여 줍니다.

AWS CLI

디렉터리의 그룹 멤버십을 나열하려면

다음 `list-groups-for-member` 예제에서는 지정된 디렉터리의 지정된 사용자에 대한 그룹 멤버십을 나열합니다.

```
aws ds-data list-groups-for-member \
    --directory-id d-1234567890 \
    --sam-account-name 'john.doe'
```

출력:

```
{
    "Groups": [
        {
            "GroupScope": "Global",
            "GroupType": "Security",
            "SAMAccountName": "Domain Users",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4567"
        }
    ],
    "DirectoryId": "d-1234567890",
    "MemberRealm": "corp.example.com",
    "Realm": "corp.example.com"
}
```

자세한 내용은 AWS Directory Service 관리 안내서의 AWS 관리형 Microsoft AD 사용자 보기 및 업데이트 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListGroupsForMember를 참조하세요.

list-groups

다음 코드 예시는 `list-groups`의 사용 방법을 보여 줍니다.

AWS CLI

디렉터리의 그룹을 나열하려면

다음 `list-groups` 예제에서는 지정된 디렉터리의 그룹을 나열합니다.

```
aws ds-data list-groups \
    --directory-id d-1234567890
```

출력:

```
{
    "Groups": [
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Administrators",
            "SID": "S-1-2-33-441"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Users",
            "SID": "S-1-2-33-442"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Guests",
            "SID": "S-1-2-33-443"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
```

```
            "SAMAccountName": "Print Operators",
            "SID": "S-1-2-33-444"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Backup Operators",
            "SID": "S-1-2-33-445"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Replicator",
            "SID": "S-1-2-33-446"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Remote Desktop Users",
            "SID": "S-1-2-33-447"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Network Configuration Operators",
            "SID": "S-1-2-33-448"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Performance Monitor Users",
            "SID": "S-1-2-33-449"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Performance Log Users",
            "SID": "S-1-2-33-450"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Distributed COM Users",
            "SID": "S-1-2-33-451"
```

```
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "IIS_IUSRS",
            "SID": "S-1-2-33-452"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Cryptographic Operators",
            "SID": "S-1-2-33-453"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Event Log Readers",
            "SID": "S-1-2-33-454"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Certificate Service DCOM Access",
            "SID": "S-1-2-33-456"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "RDS Remote Access Servers",
            "SID": "S-1-2-33-457"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "RDS Endpoint Servers",
            "SID": "S-1-2-33-458"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "RDS Management Servers",
            "SID": "S-1-2-33-459"
        },
        {
```

```
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Hyper-V Administrators",
            "SID": "S-1-2-33-460"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Access Control Assistance Operators",
            "SID": "S-1-2-33-461"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Remote Management Users",
            "SID": "S-1-2-33-462"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Storage Replica Administrators",
            "SID": "S-1-2-33-463"
        },
        {
            "GroupScope": "Global",
            "GroupType": "Security",
            "SAMAccountName": "Domain Computers",
            "SID": "S-1-2-34-56789123456-7891012345-6789123486-789"
        },
        {
            "GroupScope": "Global",
            "GroupType": "Security",
            "SAMAccountName": "Domain Controllers",
            "SID": "S-1-2-34-56789123456-7891012345-6789123486-790"
        },
        {
            "GroupScope": "Universal",
            "GroupType": "Security",
            "SAMAccountName": "Schema Admins",
            "SID": "S-1-2-34-56789123456-7891012345-6789123486-791"
        },
        {
            "GroupScope": "Universal",
            "GroupType": "Security",
```

```
                "SAMAccountName": "Enterprise Admins",
                "SID": "S-1-2-34-56789123456-7891012345-6789123486-792"
            },
            {
                "GroupScope": "DomainLocal",
                "GroupType": "Security",
                "SAMAccountName": "Cert Publishers",
                "SID": "S-1-2-34-56789123456-7891012345-6789123486-793"
            },
            {
                "GroupScope": "Global",
                "GroupType": "Security",
                "SAMAccountName": "Domain Admins",
                "SID": "S-1-2-34-56789123456-7891012345-6789123486-794"
            },
            {
                "GroupScope": "Global",
                "GroupType": "Security",
                "SAMAccountName": "Domain Users",
                "SID": "S-1-2-34-56789123456-7891012345-6789123486-795"
            },
            {
                "GroupScope": "Global",
                "GroupType": "Security",
                "SAMAccountName": "Domain Guests",
                "SID": "S-1-2-34-56789123456-7891012345-6789123486-796"
            },
            {
                "GroupScope": "Global",
                "GroupType": "Security",
                "SAMAccountName": "Group Policy Creator Owners",
                "SID": "S-1-2-34-56789123456-7891012345-6789123486-797"
            },
            {
                "GroupScope": "DomainLocal",
                "GroupType": "Security",
                "SAMAccountName": "RAS and IAS Servers",
                "SID": "S-1-2-34-56789123456-7891012345-6789123486-798"
            },
            {
                "GroupScope": "BuiltinLocal",
                "GroupType": "Security",
                "SAMAccountName": "Server Operators",
                "SID": "S-1-2-33-464"
```

```
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Account Operators",
            "SID": "S-1-2-33-465"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Pre-Windows 2000 Compatible Access",
            "SID": "S-1-2-33-466"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Incoming Forest Trust Builders",
            "SID": "S-1-2-33-467"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Windows Authorization Access Group",
            "SID": "S-1-2-33-468"
        },
        {
            "GroupScope": "BuiltinLocal",
            "GroupType": "Security",
            "SAMAccountName": "Terminal Server License Servers",
            "SID": "S-1-2-33-469"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "Allowed RODC Password Replication Group",
            "SID": "S-1-2-34-56789123456-7891012345-6789123486-798"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "Denied RODC Password Replication Group",
            "SID": "S-1-2-34-56789123456-7891012345-6789123486-799"
        },
        {
```

```
                "GroupScope": "Global",
                "GroupType": "Security",
                "SAMAccountName": "Read-only Domain Controllers",
                "SID": "S-1-2-34-56789123456-7891012345-6789123486-800"
            },
            {
                "GroupScope": "Universal",
                "GroupType": "Security",
                "SAMAccountName": "Enterprise Read-only Domain Controllers",
                "SID": "S-1-2-34-56789123456-7891012345-6789123486-801"
            },
            {
                "GroupScope": "Global",
                "GroupType": "Security",
                "SAMAccountName": "Cloneable Domain Controllers",
                "SID": "S-1-2-34-56789123456-7891012345-6789123486-802"
            },
            {
                "GroupScope": "Global",
                "GroupType": "Security",
                "SAMAccountName": "Protected Users",
                "SID": "S-1-2-34-56789123456-7891012345-6789123486-803"
            },
            {
                "GroupScope": "Global",
                "GroupType": "Security",
                "SAMAccountName": "Key Admins",
                "SID": "S-1-2-34-56789123456-7891012345-6789123486-804"
            },
            {
                "GroupScope": "Universal",
                "GroupType": "Security",
                "SAMAccountName": "Enterprise Key Admins",
                "SID": "S-1-2-34-56789123456-7891012345-6789123486-805"
            },
            {
                "GroupScope": "DomainLocal",
                "GroupType": "Security",
                "SAMAccountName": "DnsAdmins",
                "SID": "S-1-2-34-5678901234-5678901234-5678910123-4567"
            },
            {
                "GroupScope": "Global",
                "GroupType": "Security",
```

```
                "SAMAccountName": "DnsUpdateProxy",
                "SID": "S-1-2-34-5678901234-5678901234-5678910123-4568"
            },
            {
                "GroupScope": "DomainLocal",
                "GroupType": "Security",
                "SAMAccountName": "Admins",
                "SID": "S-1-2-34-5678901234-5678901234-5678910123-4569"
            },
            {
                "GroupScope": "DomainLocal",
                "GroupType": "Security",
                "SAMAccountName": "AWSAdministrators",
                "SID": "S-1-2-34-5678901234-5678901234-5678910123-4570"
            },
            {
                "GroupScope": "DomainLocal",
                "GroupType": "Security",
                "SAMAccountName": "AWS Object Management Service Accounts",
                "SID": "S-1-2-34-5678901234-5678901234-5678910123-4571"
            },
            {
                "GroupScope": "DomainLocal",
                "GroupType": "Security",
                "SAMAccountName": "AWS Private CA Connector for AD Delegated Group",
                "SID": "S-1-2-34-5678901234-5678901234-5678910123-4572"
            },
            {
                "GroupScope": "DomainLocal",
                "GroupType": "Security",
                "SAMAccountName": "AWS Application and Service Delegated Group",
                "SID": "S-1-2-34-5678901234-5678901234-5678910123-4573"
            },
            {
                "GroupScope": "DomainLocal",
                "GroupType": "Security",
                "SAMAccountName": "AWS Delegated Administrators",
                "SID": "S-1-2-34-5678901234-5678901234-5678910123-4574"
            },
            {
                "GroupScope": "DomainLocal",
                "GroupType": "Security",
                "SAMAccountName": "AWS Delegated FSx Administrators",
                "SID": "S-1-2-34-5678901234-5678901234-5678910123-4575"
```

```
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Account Operators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4576"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Active Directory Based Activation
 Administrators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4577"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Allowed to Authenticate Objects",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4578"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Allowed to Authenticate to Domain
 Controllers",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4579"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Deleted Object Lifetime
 Administrators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4580"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Distributed File System
 Administrators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4581"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
```

```
            "SAMAccountName": "AWS Delegated Dynamic Host Configuration Protocol 
Administrators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4582"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Enterprise Certificate Authority 
Administrators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4583"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Fine Grained Password Policy 
Administrators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4584"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Group Policy Administrators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4585"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Managed Service Account 
Administrators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4586"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Read Foreign Security Principals",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4587"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Remote Access Service Administrators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4588"
        },
        {
```

```
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Replicate Directory Changes
 Administrators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4588"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Sites and Services Administrators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4589"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated System Management Administrators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4590"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Terminal Server Licensing
 Administrators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4591"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated User Principal Name Suffix
 Administrators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4592"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Add Workstations To Domain Users",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4593"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Domain Name System Administrators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4594"
        },
```

```
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Kerberos Delegation Administrators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4595"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated Server Administrators",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4596"
        },
        {
            "GroupScope": "DomainLocal",
            "GroupType": "Security",
            "SAMAccountName": "AWS Delegated MS-NPRC Non-Compliant Devices",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4597"
        },
        {
            "GroupScope": "Global",
            "GroupType": "Security",
            "SAMAccountName": "Remote Access",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4598"
        },
        {
            "GroupScope": "Global",
            "GroupType": "Security",
            "SAMAccountName": "Accounting",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4599"
        },
        {
            "GroupScope": "Global",
            "GroupType": "Distribution",
            "SAMAccountName": "sales",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4567"
        }
    ],
    "DirectoryId": "d-1234567890",
    "Realm": "corp.example.com"
}
```

자세한 내용은 AWS Directory Service 관리 안내서의 AWS 관리형 Microsoft AD 그룹 세부 정보 보기 및 업데이트 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListGroups를 참조하세요.

list-users

다음 코드 예시는 list-users의 사용 방법을 보여 줍니다.

AWS CLI

디렉터리의 사용자를 나열하려면

다음 list-users 예제에서는 지정된 디렉터리의 사용자를 나열합니다.

```
aws ds-data list-users \
    --directory-id d-1234567890
```

출력:

```
{
    "Users": [
        {
            "Enabled": true,
            "SAMAccountName": "Administrator",
            "SID": "S-1-2-34-5678910123-4567895012-3456789012-345"
        },
        {
            "Enabled": false,
            "SAMAccountName": "Guest",
            "SID": "S-1-2-34-5678910123-4567895012-3456789012-345"
        },
        {
            "Enabled": false,
            "SAMAccountName": "krbtgt",
            "SID": "S-1-2-34-5678910123-4567895012-3456789012-346"
        },
        {
            "Enabled": true,
            "SAMAccountName": "Admin",
            "SID": "S-1-2-34-5678910123-4567895012-3456789012-347"
        },
        {
            "Enabled": true,
            "SAMAccountName": "Richard Roe",
            "SID": "S-1-2-34-5678910123-4567895012-3456789012-348"
```

```
        },
        {
            "Enabled": true,
            "SAMAccountName": "Jane Doe",
            "SID": "S-1-2-34-5678910123-4567895012-3456789012-349"
        },
        {
            "Enabled": true,
            "SAMAccountName": "AWS_WGnzYlN6YyY",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4567"
        },
        {
            "Enabled": true,
            "SAMAccountName": "john.doe",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4568"
        }
    ],
    "DirectoryId": "d-1234567890",
    "Realm": "corp.example.com"
}
```

자세한 내용은 AWS Directory Service 관리 안내서의 AWS 관리형 Microsoft AD 사용자 보기 및 업데이트 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListUsers를 참조하세요.

remove-group-member

다음 코드 예시는 remove-group-member의 사용 방법을 보여 줍니다.

AWS CLI

디렉터리에서 그룹 멤버를 제거하려면

다음 remove-group-member 예제에서는 지정된 디렉터리의 지정된 그룹에서 지정된 그룹 멤버를 제거합니다.

```
aws ds-data remove-group-member \
    --directory-id d-1234567890 \
    --group-name 'sales' \
    --member-name 'john.doe'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Directory Service 관리 안내서의 그룹에 AWS 관리형 Microsoft AD 멤버 추가 및 제거 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RemoveGroupMember를 참조하세요.

reset-user-password

다음 코드 예시는 reset-user-password의 사용 방법을 보여 줍니다.

AWS CLI

디렉터리에서 사용자 암호를 재설정하려면

다음 reset-user-password 예제에서는 지정된 디렉터리에서 지정된 사용자를 재설정하고 활성화합니다.

```
aws ds reset-user-password \
    --directory-id d-1234567890 \
    --user-name 'john.doe' \
    --new-password 'password'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Directory Service 관리 안내서의 AWS 관리형 Microsoft AD 사용자의 암호 재설정 및 활성화를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ResetUserPassword를 참조하세요.

search-groups

다음 코드 예시는 search-groups의 사용 방법을 보여 줍니다.

AWS CLI

디렉터리에서 그룹을 검색하려면

다음 search-groups 예제에서는 지정된 디렉터리에서 지정된 그룹을 검색합니다.

```
aws ds-data search-groups \
    --directory-id d-1234567890 \
    --search-attributes 'SamAccountName' \
    --search-string 'sales'
```

출력:

```
{
    "Groups": [
        {
            "GroupScope": "Global",
            "GroupType": "Distribution",
            "SAMAccountName": "sales",
            "SID": "S-1-2-34-5678901234-5678901234-5678910123-4567"
        }
    ],
    "DirectoryId": "d-1234567890",
    "Realm": "corp.example.com"
}
```

자세한 내용은 AWS Directory Service 관리 안내서의 AWS 관리형 Microsoft AD 그룹 세부 정보 보기 및 업데이트 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 SearchGroups를 참조하세요.

search-users

다음 코드 예시는 search-users의 사용 방법을 보여 줍니다.

AWS CLI

디렉터리에서 사용자를 검색하려면

다음 search-users 예제에서는 지정된 디렉터리에서 지정된 사용자를 검색합니다.

```
aws ds-data search-users \
    --directory-id d-1234567890 \
    --search-attributes 'SamAccountName' \
    --Search-string 'john.doe'
```

출력:

```
{
    "Users": [
        {
            "Enabled": true,
            "SAMAccountName": "john.doe",
```

```
                "SID": "S-1-2-34-5678901234-5678901234-5678910123-4567"
            }
        ],
        "DirectoryId": "d-1234567890",
        "Realm": "corp.example.com"
}
```

자세한 내용은 AWS Directory Service 관리 안내서의 AWS 관리형 Microsoft AD 사용자 보기 및 업데이트 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 SearchUsers를 참조하세요.

update-group

다음 코드 예시는 update-group의 사용 방법을 보여 줍니다.

AWS CLI

디렉터리에서 그룹의 속성을 업데이트하려면

다음 update-group 예제에서는 지정된 디렉터리의 지정된 그룹에 대해 지정된 속성을 업데이트합니다.

```
aws ds-data update-group \
    --directory-id d-1234567890 \
    --sam-account-name 'sales' \
    --update-type 'REPLACE' \
    --group-type 'Distribution'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Directory Service 관리 안내서의 AWS 관리형 Microsoft AD 그룹 세부 정보 보기 및 업데이트 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateGroup을 참조하세요.

update-user

다음 코드 예시는 update-user의 사용 방법을 보여 줍니다.

AWS CLI

디렉터리에서 사용자의 속성을 업데이트하려면

다음 update-user 예제에서는 지정된 디렉터리의 지정된 사용자에 대해 지정된 속성을 업데이트합니다.

```
aws ds-data update-user \
    --directory-id d-1234567890 \
    --sam-account-name 'john.doe' \
    --update-type 'ADD' \
    --email-address 'example.corp.com'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Directory Service 관리 안내서의 AWS 관리형 Microsoft AD 사용자 보기 및 업데이트 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateUser를 참조하세요.

AWS CLI를 사용한 AWS DMS 예시

다음 코드 예시에서는 AWS DMS에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

add-tags-to-resource

다음 코드 예시에서는 add-tags-to-resource 코드를 사용하는 방법을 보여줍니다.

AWS CLI

태그를 리소스에 추가

다음 add-tags-to-resource 예시에서는 복제 인스턴스를 태그에 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 Database Migration Service 사용 설명서의 Tagging Resources을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AddTagsToResource를 참조하세요.

`create-endpoint`

다음 코드 예시에서는 `create-endpoint` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

엔드포인트 생성

다음 `create-endpoint` 예시에서는 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 Database Migration Service 사용 설명서의 Working with AWS DMS Endpoints를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateEndpoint를 참조하세요.

create-event-subscription

다음 코드 예시에서는 `create-event-subscription` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

이벤트 구독을 나열하는 방법

다음 `create-event-subscription` 예시에서는 Amazon SNS 주제(my-sns-topic)에 대한 이벤트 구독을 생성합니다.

```
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 Database Migration Service 사용 설명서의 Working with Events and Notifications를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateEventSubscription을 참조하세요.

`create-replication-instance`

다음 코드 예시에서는 `create-replication-instance` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

복제 인스턴스 생성

다음 `create-replication-instance` 예시에서는 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 Database Migration Service 사용 설명서의 Working with an AWS DMS Replication Instance를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateReplicationInstance 섹션을 참조하세요.

`create-replication-subnet-group`

다음 코드 예시에서는 `create-replication-subnet-group` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

서브넷 그룹 생성

다음 `create-replication-subnet-group` 예시에서는 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 Database Migration Service 사용 설명서의 Setting Up a Network for a Replication Instance를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateReplicationSubnetGroup을 참조하세요.

`create-replication-task`

다음 코드 예시에서는 `create-replication-task` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

복제 태스크 생성

다음 `create-replication-task` 예시에서는 복제 태스크를 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.json`의 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 Database Migration Service 사용 설명서의 Working with AWS DMS Tasks를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateReplicationTask를 참조하세요.

delete-connection

다음 코드 예시에서는 `delete-connection` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

연결 삭제

다음 `delete-connection` 예시에서는 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 Database Migration Service 사용 설명서의 https://docs.aws.amazon.com/dms/latest/userguide/CHAP_Endpoints.Creating.html 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteConnection을 참조하세요.

delete-endpoint

다음 코드 예시에서는 delete-endpoint 코드를 사용하는 방법을 보여줍니다.

AWS CLI

엔드포인트 삭제

다음 delete-endpoint 예시에서는 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 Database Migration Service 사용 설명서의 Working with AWS DMS Endpoints를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteEndpoint 섹션을 참조하세요.

delete-event-subscription

다음 코드 예시에서는 `delete-event-subscription` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

이벤트 구독 삭제

다음 `delete-event-subscription` 예시는 Amazon SNS 주제를 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 Database Migration Service 사용 설명서의 Working with Events and Notifications를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteEventSubscription을 참조하세요.

`delete-replication-instance`

다음 코드 예시에서는 `delete-replication-instance` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

복제 인스턴스 삭제

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 Database Migration Service 사용 설명서의 Working with an AWS DMS Replication Instance를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteReplicationInstance를 참조하세요.

delete-replication-subnet-group

다음 코드 예시에서는 delete-replication-subnet-group 코드를 사용하는 방법을 보여줍니다.

AWS CLI

서브넷 그룹 삭제

다음 delete-replication-subnet-group 예시에서는 서브넷 그룹을 삭제합니다.

```
aws dms delete-replication-subnet-group \
--replication-subnet-group-identifier my-subnet-group
```

출력:

```
(none)
```

자세한 내용은 AWS Database Migration Service 사용 설명서의 Setting Up a Network for a Replication Instance를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteReplicationSubnetGroup](#)를 참조하세요.

delete-replication-task

다음 코드 예시에서는 delete-replication-task 코드를 사용하는 방법을 보여줍니다.

AWS CLI

복제 태스크를 삭제하는 방법

다음 delete-replication-task 예시에서는 복제 태스크를 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 Database Migration Service 사용 설명서의 [Working with AWS DMS Tasks](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteReplicationTask](#)을 참조하세요.

`describe-account-attributes`

다음 코드 예시에서는 `describe-account-attributes` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

계정 속성 설명

다음 `describe-account-attributes` 예시에서는 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeAccountAttributes를 참조하세요.

`describe-certificates`

다음 코드 예시에서는 `describe-certificates` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용 가능한 인증서를 나열하는 방법

다음 `describe-certificates` 예시에서는 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 Database Migration Service 사용 설명서의 Using SSL을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeCertificates를 참조하세요.

`describe-connections`

다음 코드 예시에서는 `describe-connections` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

연결 삭제

다음 `describe-connections` 예시에는 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 Database Migration Servic 사용 설명서의 Creating Source and Target Endpoints를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeConnections을 참조하세요.

`describe-endpoint-types`

다음 코드 예시에서는 `describe-endpoint-types` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용 가능한 엔드포인트 유형 나열

다음 `describe-endpoint-types` 예시에서는 사용 가능한 MySQL 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 Database Migration Servic 사용 설명서의 Working with AWS DMS Endpoints <https://docs.aws.amazon.com/dms/latest/userguide/CHAP_Endpoints.html>`__을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeEndpointTypes를 참조하세요.

`describe-endpoints`

다음 코드 예시에서는 `describe-endpoints` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

엔드포인트 설명

다음 `describe-endpoints` 예시에서는 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 Database Migration Service 사용 설명서의 Working with AWS DMS Endpoints를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeEndpoints 섹션을 참조하세요.

`describe-event-categories`

다음 코드 예시에서는 `describe-event-categories` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

이벤트 카테고리 설명

다음 `describe-event-categories` 예시에서는 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 Database Migration Service 사용 설명서의 Working with Events and Notifications를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeEventCategories를 참조하세요.

describe-event-subscriptions

다음 코드 예시에서는 `describe-event-subscriptions` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

이벤트 구독 설명

다음 `describe-event-subscriptions` 예시에서는 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 Database Migration Service 사용 설명서의 Working with Events and Notifications를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeEventSubscriptions를 참조하세요.

describe-events

다음 코드 예시에서는 `describe-events` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

DMS 이벤트를 나열하는 방법

다음 `describe-events` 예시에서는 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 Database Migration Service 사용 설명서의 Working with Events and Notifications를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeEvents 섹션을 참조하세요.

describe-orderable-replication-instances

다음 코드 예시에서는 describe-orderable-replication-instances 코드를 사용하는 방법을 보여줍니다.

AWS CLI

주문 가능한 복제 인스턴스를 설명하는 방법

다음 describe-orderable-replication-instances 예시에서는 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 Database Migration Service 사용 설명서의 Working with an AWS DMS Replication Instance를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeOrderableReplicationInstances를 참조하세요.

describe-refresh-schemas-status

다음 코드 예시에서는 describe-refresh-schemas-status 코드를 사용하는 방법을 보여줍니다.

AWS CLI

엔드포인트의 새로 고침 상태를 나열하는 방법

다음 describe-refresh-schemas-status 예시에서는 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
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeRefreshSchemasStatus를 참조하세요.

`describe-replication-instances`

다음 코드 예시에서는 `describe-replication-instances` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

복제 인스턴스를 설명하는 방법

다음 `describe-replication-instances` 예시에서는 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 Database Migration Service 사용 설명서의 Working with an AWS DMS Replication Instance를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeReplicationInstances를 참조하세요.

`describe-replication-subnet-groups`

다음 코드 예시에서는 `describe-replication-subnet-groups` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용 가능한 서브넷 그룹을 표시하는 방법

다음 describe-replication-subnet-groups 예시에서는 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 Database Migration Service 사용 설명서의 Setting Up a Network for a Replication Instance를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeReplicationSubnetGroups를 참조하세요.

`describe-replication-task-assessment-results`

다음 코드 예시에서는 `describe-replication-task-assessment-results` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

복제 태스크 평가 결과를 나열하는 방법

다음 `describe-replication-task-assessment-results` 예시에서는 이전 태스크 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 Database Migration Servic 사용 설명서의 Creating a Task Assessment Report를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeReplicationTaskAssessmentResults를 참조하세요.

describe-replication-tasks

다음 코드 예시에서는 describe-replication-tasks 코드를 사용하는 방법을 보여줍니다.

AWS CLI

복제 태스크를 설명하는 방법

다음 describe-replication-tasks 예시에서는 현재 복제 태스크를 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 Database Migration Service 사용 설명서의 Working with AWS DMS Tasks를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeReplicationTasks를 참조하세요.

`describe-schemas`

다음 코드 예시에서는 `describe-schemas` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

데이터베이스 스키마를 설명하는 방법

다음 `describe-schemas` 예시에서는 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 Database Migration Service 사용 설명서의 주제 제목을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeSchemas를 참조하세요.

`list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

리소스에 대한 태그 나열

다음 `list-tags-for-resource` 예시에서는 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 Database Migration Service 사용 설명서의 Tagging Resources을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource를 참조하세요.

`modify-endpoint`

다음 코드 예시에서는 `modify-endpoint` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

엔드포인트 수정

다음 `modify-endpoint` 예시에서는 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 Database Migration Servic 사용 설명서의 Working with AWS DMS Endpoints <https://docs.aws.amazon.com/dms/latest/userguide/CHAP_Endpoints.html>`__을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyEndpoint를 참조하세요.

modify-event-subscription

다음 코드 예시에서는 modify-event-subscription 코드를 사용하는 방법을 보여줍니다.

AWS CLI

이벤트 구독 수정

다음 modify-event-subscription 예시에서는 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 Database Migration Service 사용 설명서의 Working with Events and Notifications를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyEventSubscription을 참조하세요.

modify-replication-instance

다음 코드 예시에서는 modify-replication-instance 코드를 사용하는 방법을 보여줍니다.

AWS CLI

복제 인스턴스 수정

다음 modify-replication-instance 예시에서는 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 Database Migration Service 사용 설명서의 Working with an AWS DMS Replication Instance를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyReplicationInstance을 참조하세요.

modify-replication-subnet-group

다음 코드 예시에서는 modify-replication-subnet-group 코드를 사용하는 방법을 보여줍니다.

AWS CLI

서브넷 그룹 수정

다음 modify-replication-subnet-group 예시에서는 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 Database Migration Service 사용 설명서의 Setting Up a Network for a Replication Instance를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyReplicationSubnetGroup을 참조하세요.

modify-replication-task

다음 코드 예시에서는 `modify-replication-task` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

복제 태스크를 수정하는 방법

다음 `modify-replication-task` 예시에서는 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.json`의 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 Database Migration Service 사용 설명서의 [Working with AWS DMS Tasks](https://docs.aws.amazon.com/)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyReplicationTask를 참조하세요.

reboot-replication-instance

다음 코드 예시에서는 `reboot-replication-instance` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

복제 인스턴스 재부팅

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 Database Migration Service 사용 설명서의 Working with an AWS DMS Replication Instance를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RebootReplicationInstance을 참조하세요.

refresh-schemas

다음 코드 예시에서는 `refresh-schemas` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

데이터베이스 스키마를 새로 고치려면

다음 `refresh-schemas` 예시에서는 AWS DMS가 엔드포인트에서 스키마 목록을 새로 고치도록 요청합니다.

```
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
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [RefreshSchemas](#)를 참조하세요.

`reload-tables`

다음 코드 예시에서는 `reload-tables` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

엔드포인트에서 사용할 수 있는 테이블 목록을 새로 고치려면

다음 `reload-tables` 예시에서는 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [ReloadTables](#)를 참조하세요.

`remove-tags-from-resource`

다음 코드 예시에서는 `remove-tags-from-resource` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

복제 인스턴스에서 태그를 제거하는 방법

다음 `remove-tags-from-resource` 예시에서는 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 Database Migration Service 사용 설명서의 [Tagging Resources](#)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [RemoveTagsFromResource](#)를 참조하세요.

`start-replication-task-assessment`

다음 코드 예시에서는 `start-replication-task-assessment` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

태스크 평가를 시작하는 방법

다음 `start-replication-task-assessment` 예시에서는 복제 태스크 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 Database Migration Servic 사용 설명서의 Creating a Task Assessment Report를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartReplicationTaskAssessment을 참조하세요.

start-replication-task

다음 코드 예시에서는 start-replication-task 코드를 사용하는 방법을 보여줍니다.

AWS CLI

복제 태스크 시작

다음 command-name 예시에서는 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 Database Migration Service 사용 설명서의 Working with AWS DMS Tasks를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartReplicationTask을 참조하세요.

stop-replication-task

다음 코드 예시에서는 stop-replication-task 코드를 사용하는 방법을 보여줍니다.

AWS CLI

작업을 중지하려면

다음 stop-replication-task 예시에서는 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 Database Migration Service 사용 설명서의 Working with AWS DMS Tasks를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StopReplicationTask를 참조하세요.

test-connection

다음 코드 예시에서는 test-connection 코드를 사용하는 방법을 보여줍니다.

AWS CLI

엔드포인트에 대한 연결을 테스트하는 방법

다음 test-connection 예시에서는 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 Database Migration Servic 사용 설명서의 Creating source and target endpoints를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TestConnection을 참조하세요.

AWS CLI를 사용한 Amazon DocumentDB 예시

다음 코드 예시에서는 Amazon DocumentDB에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`add-tags-to-resource`

다음 코드 예시에서는 `add-tags-to-resource` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

지정된 리소스에 하나 이상의 태그 추가

다음 add-tags-to-resource 예시에서는 sample-cluster에 3개의 태그를 추가합니다. 키 이름은 있지만 값이 없는 태그(CropB)가 있습니다.

```
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 Tagging Amazon DocumentDB Resources를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AddTagsToResource를 참조하세요.

apply-pending-maintenance-action

다음 코드 예시에서는 apply-pending-maintenance-action 코드를 사용하는 방법을 보여줍니다.

AWS CLI

다음 유지 관리 기간 동안 보류 중인 유지 관리 작업이 수행되도록 하는 방법

다음 apply-pending-maintenance-action 예시에서는 다음 예정된 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 Applying Amazon DocumentDB Updates를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ApplyPendingMaintenanceAction 섹션을 참조하세요.

copy-db-cluster-parameter-group

다음 코드 예시에서는 copy-db-cluster-parameter-group 코드를 사용하는 방법을 보여줍니다.

AWS CLI

기존 DB 클러스터 파라미터 그룹을 복제하는 방법

다음 copy-db-cluster-parameter-group 예시에서는 custom-docdb3-6-copy라는 이름으로 파라미터 그룹 custom-docdb3-6의 복사본을 생성합니다. 복사할 때 새 파라미터 그룹에 태그가 추가됩니다.

```
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 [Copying an Amazon DocumentDB Cluster Parameter Group]을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [CopyDbClusterParameterGroup] 섹션을 참조하세요.

copy-db-cluster-snapshot

다음 코드 예시에서는 copy-db-cluster-snapshot 코드를 사용하는 방법을 보여줍니다.

AWS CLI

스냅샷의 사본 생성

다음 copy-db-cluster-snapshot 예시에서는 sample-cluster-snapshot-copy라는 sample-cluster-snapshot의 사본을 생성합니다. 복사본에는 원본의 모든 태그에 키 이름이 CopyNumber인 새 태그가 추가됩니다.

```
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 Copying a Cluster Snapshot을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CopyDbClusterSnapshot 섹션을 참조하세요.

create-db-cluster-parameter-group

다음 코드 예시에서는 create-db-cluster-parameter-group 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Amazon DocumentDB 클러스터 파라미터 그룹 생성

다음 create-db-cluster-parameter-group 예시에서는 docdb3.6 패밀리를 사용하여 DB 클러스터 파라미터 그룹 sample-parameter-group을 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 Creating an Amazon DocumentDB Cluster Parameter Group을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateDbClusterParameterGroup 섹션을 참조하세요.

`create-db-cluster-snapshot`

다음 코드 예시에서는 `create-db-cluster-snapshot` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

수동 Amazon DocumentDB 클러스터 스냅샷을 생성하는 방법

다음 `create-db-cluster-snapshot` 예시에서는 sample-cluster-snapshot이라는 이름의 Amazon DB 클러스터 스냅샷을 생성합니다.

```
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 Creating a Manual Cluster Snapshot을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateDbClusterSnapshot 섹션을 참조하세요.

create-db-cluster

다음 코드 예시에서는 `create-db-cluster` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Amazon DocumentDB 클러스터 생성

다음 `create-db-cluster` 예시에서는 일요일 20:30~11:00을 기본 유지 관리 시간으로 설정한 `sample-cluster`라는 이름의 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 Creating an Amazon DocumentDB Cluster를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateDbCluster 섹션을 참조하세요.

`create-db-instance`

다음 코드 예시에서는 `create-db-instance` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Amazon DocumentDB 클러스터 인스턴스 생성

다음 `create-db-instance` 예시 코드는 Amazon DocumentDB 클러스터의 인스턴스 `sample-cluster-instance-2`를 나열합니다.

```
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 Adding an Amazon DocumentDB Instance to a Cluster를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [CreateDbInstance](#) 섹션을 참조하세요.

`create-db-subnet-group`

다음 코드 예시에서는 `create-db-subnet-group` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Amazon DocumentDB 서브넷 그룹 생성

다음 `create-db-subnet-group` 예시에서는 `sample-subnet-group`이라는 이름의 Amazon DocumentDB 서브넷 그룹을 생성합니다.

```
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 Creating an Amazon DocumentDB Subnet Group을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateDbSubnetGroup 섹션을 참조하세요.

delete-db-cluster-parameter-group

다음 코드 예시에서는 delete-db-cluster-parameter-group 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Amazon DocumentDB 클러스터 파라미터 그룹 삭제

다음 delete-db-cluster-parameter-group 예시에서는 Amazon DocumentDB 파라미터 그룹 sample-parameter-group을 삭제합니다.

```
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 Deleting an Amazon DocumentDB Cluster Parameter Group을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteDbClusterParameterGroup 섹션을 참조하세요.

delete-db-cluster-snapshot

다음 코드 예시에서는 delete-db-cluster-snapshot 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Amazon DocumentDB 클러스터 스냅샷 삭제

다음 `delete-db-cluster-snapshot` 예시에서는 Amazon DocumentDB 클러스터 스냅샷 `sample-cluster-snapshot`을 삭제합니다.

```
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 Deleting a Cluster Snapshot을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteDbClusterSnapshot 섹션을 참조하세요.

`delete-db-cluster`

다음 코드 예시에서는 `delete-db-cluster` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Amazon DocumentDB 클러스터 삭제

다음 `delete-db-cluster` 예시에서는 Amazon DocumentDB 클러스터 `sample-cluster`를 삭제합니다. 클러스터를 삭제하기 전에 클러스터를 백업하지 않습니다 참고: 클러스터를 삭제하려면 먼저 클러스터와 연결된 모든 인스턴스를 삭제해야 합니다.

```
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 Deleting an Amazon DocumentDB Cluster를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteDbCluster 섹션을 참조하세요.

delete-db-instance

다음 코드 예시에서는 `delete-db-instance` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Amazon DocumentDB 인스턴스 삭제

다음 `delete-db-instance` 예시에서는 Amazon DocumentDB 인스턴스 `sample-cluster-instance-2`를 삭제합니다.

```
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 Deleting an Amazon DocumentDB Instance를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteDbInstance 섹션을 참조하세요.

`delete-db-subnet-group`

다음 코드 예시에서는 `delete-db-subnet-group` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Amazon DocumentDB 서브넷 그룹 삭제

다음 `delete-db-subnet-group` 예시에서는 Amazon DocumentDB 서브넷 그룹 `sample-subnet-group`을 삭제합니다.

```
aws docdb delete-db-subnet-group \
    --db-subnet-group-name sample-subnet-group
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon DocumentDB 개발자 안내서의 Deleting an Amazon DocumentDB Subnet Group을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteDbSubnetGroup 섹션을 참조하세요.

describe-db-cluster-parameter-groups

다음 코드 예시에서는 describe-db-cluster-parameter-groups 코드를 사용하는 방법을 보여줍니다.

AWS CLI

하나 이상의 Amazon DocumentDB 클러스터 파라미터 그룹의 세부 정보를 보는 방법

다음 describe-db-cluster-parameter-groups 예시에서는 Amazon DocumentDB 클러스터 파라미터 그룹 custom3-6-param-grp에 대한 세부 정보를 표시합니다.

```
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 Viewing Amazon DocumentDB Cluster Parameter Groups를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDbClusterParameterGroups 섹션을 참조하세요.

describe-db-cluster-parameters

다음 코드 예시에서는 describe-db-cluster-parameters 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Amazon DocumentDB 클러스터 파라미터 그룹의 세부 파라미터 목록 보기

다음 describe-db-cluster-parameters 예시에서는 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 Viewing Amazon DocumentDB Cluster Parameters를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDbClusterParameters 섹션을 참조하세요.

describe-db-cluster-snapshot-attributes

다음 코드 예시에서는 describe-db-cluster-snapshot-attributes 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Amazon DocumentDB 스냅샷 속성 이름 및 값을 나열하는 방법

다음 describe-db-cluster-snapshot-attributes 예시에서는 Amazon DocumentDB 스냅샷 sample-cluster-snapshot의 속성 이름과 값을 나열합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDbClusterSnapshotAttributes 섹션을 참조하세요.

describe-db-cluster-snapshots

다음 코드 예시에서는 describe-db-cluster-snapshots 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Amazon DocumentDB 스냅샷을 설명하는 방법

다음 describe-db-cluster-snapshots 예시에서는 Amazon DocumentDB 스냅샷 sample-cluster-snapshot에 대한 세부 정보를 표시합니다.

```
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 DescribeDBClusterSnapshots를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDbClusterSnapshots 섹션을 참조하세요.

describe-db-clusters

다음 코드 예시에서는 `describe-db-clusters` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

하나 이상의 Amazon DocumentDB 클러스터에 대한 자세한 정보 가져오기

다음 `describe-db-clusters` 예시에서는 Amazon DocumentDB 클러스터 `sample-cluster`에 대한 세부 정보를 표시합니다. `--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 Describing Amazon DocumentDB Clusters를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDbClusters 섹션을 참조하세요.

describe-db-engine-versions

다음 코드 예시에서는 describe-db-engine-versions 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용 가능한 Amazon DocumentDB 엔진 버전을 나열하는 방법

다음 describe-db-engine-versions 예시에서는 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 DescribeDBEngineVersions를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDbEngineVersions 섹션을 참조하세요.

describe-db-instances

다음 코드 예시에서는 `describe-db-instances` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

프로비저닝된 Amazon DocumentDB 인스턴스에 대한 정보 찾기

다음 `describe-db-instances` 예시에서는 Amazon DocumentDB 인스턴스 `sample-cluster-instance`에 대한 세부 정보를 표시합니다. `--db-instance-identifier` 파라미터가 생략되면 최대 100개의 인스턴스에 대한 정보를 받게 됩니다.

```
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 Describing Amazon DocumentDB Instances를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDbInstances 섹션을 참조하세요.

`describe-db-subnet-groups`

다음 코드 예시에서는 `describe-db-subnet-groups` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Amazon DocumentDB 서브넷 설명 목록을 검색하는 방법

다음 `describe-db-subnet-groups` 예시에서는 이름이 `default`인 Amazon DocumentDB 서브넷에 대한 세부 정보를 설명합니다.

```
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 Describing Subnet Groups를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDbSubnetGroups 섹션을 참조하세요.

describe-engine-default-cluster-parameters

다음 코드 예시에서는 `describe-engine-default-cluster-parameters` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Amazon DocumentDB의 기본 엔진 및 시스템 파라미터 정보를 설명하는 방법

다음 `describe-engine-default-cluster-parameters` 예시에서는 Amazon DocumentDB 파라미터 그룹 `docdb3.6`의 기본 엔진 및 시스템 파라미터 정보에 대한 세부 정보를 표시합니다.

```
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 안내서의 DescribeEngineDefaultClusterParameters를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeEngineDefaultClusterParameters 섹션을 참조하세요.

describe-event-categories

다음 코드 예시에서는 `describe-event-categories` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

모든 Amazon DocumentDB 이벤트 범주를 설명하는 방법

다음 `describe-event-categories` 예시에서는 Amazon DocumentDB 이벤트 소스 유형 db-`instance`의 모든 범주를 나열합니다.

```
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 Viewing Event Categories를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeEventCategories 섹션을 참조하세요.

describe-events

다음 코드 예시에서는 describe-events 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Amazon DocumentDB 이벤트를 나열하는 방법

다음 describe-events 예시에서는 지난 24시간(1,440분) 동안 발생한 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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeEvents 섹션을 참조하세요.

describe-orderable-db-instance-options

다음 코드 예시에서는 describe-orderable-db-instance-options 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Amazon DocumentDB 인스턴스 옵션 찾기

다음 describe-orderable-db-instance-options 예시에서는 리전에 대한 Amazon DocumentDB 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 Adding an Amazon DocumentDB Instance to a Cluster를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeOrderableDBInstanceOptions 섹션을 참조하세요.

`describe-pending-maintenance-actions`

다음 코드 예시에서는 `describe-pending-maintenance-actions` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

보류 중인 Amazon DocumentDB 유지 관리 작업 나열

다음 `describe-pending-maintenance-actions` 예시에서는 보류 중인 모든 Amazon DocumentDB 유지 관리 작업을 나열합니다.

```
aws docdb describe-pending-maintenance-actions
```

출력:

```
{
```

```
    "PendingMaintenanceActions": []
}
```

자세한 내용은 Amazon DocumentDB 개발자 안내서의 Maintaining Amazon DocumentDB를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribePendingMaintenanceActions 섹션을 참조하세요.

failover-db-cluster

다음 코드 예시에서는 `failover-db-cluster` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Amazon DocumentDB 클러스터를 복제본으로 장애 조치하도록 강제하는 방법

다음 `failover-db-cluster` 예시에서는 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 Failover를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 FailoverDbCluster 섹션을 참조하세요.

list-tags-for-resource

다음 코드 예시에서는 list-tags-for-resource 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Amazon DocumentDB 리소스의 모든 태그 나열

다음 list-tags-for-resource 예시에서는 Amazon DocumentDB 클러스터 sample-cluster의 모든 태그를 나열합니다.

```
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 Listing Tags on an Amazon DocumentDB Resource를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource를 참조하세요.

modify-db-cluster-parameter-group

다음 코드 예시에서는 modify-db-cluster-parameter-group 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Amazon DocumentDB 클러스터 파라미터 그룹 수정

다음 modify-db-cluster-parameter-group 예시에서는 두 파라미터 audit_logs 및 ttl_monitor를 활성화로 설정하여 Amazon DocumentDB 클러스터 파라미터 그룹 custom3-6-param-grp를 수정합니다. 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 Modifying an Amazon DocumentDB Cluster Parameter Group을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyDbClusterParameterGroup 섹션을 참조하세요.

modify-db-cluster-snapshot-attribute

다음 코드 예시에서는 modify-db-cluster-snapshot-attribute 코드를 사용하는 방법을 보여 줍니다.

AWS CLI

예시 1: Amazon DocumentDB 스냅샷에 속성을 추가하는 방법

다음 modify-db-cluster-snapshot-attribute 예시에서는 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"
    }
}
```

예시 2: Amazon DocumentDB 스냅샷에서 속성을 제거하는 방법

다음 `modify-db-cluster-snapshot-attribute` 예시에서는 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 ModifyDBClusterSnapshotAttribute를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyDbClusterSnapshotAttribute 섹션을 참조하세요.

modify-db-cluster

다음 코드 예시에서는 `modify-db-cluster` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Amazon DocumentDB 클러스터 수정

다음 `modify-db-cluster` 예시에서는 자동 백업의 보존 기간을 7일로 설정하고 백업 및 유지 관리의 기본 기간을 변경하여 Amazon DocumentDB 클러스터 `sample-cluster`를 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 Modifying an Amazon DocumentDB Cluster를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyDbCluster 섹션을 참조하세요.

modify-db-instance

다음 코드 예시에서는 modify-db-instance 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Amazon DocumentDB 인스턴스 수정

다음 modify-db-instance 예시는 인스턴스 클래스를 db.r4.4xlarge로 변경하고 프로모션 계층을 5로 변경하여 Amazon DocumentDB 인스턴스 sample-cluster2를 수정합니다. 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 Modifying an Amazon DocumentDB Instance를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyDbInstance 섹션을 참조하세요.

modify-db-subnet-group

다음 코드 예시에서는 modify-db-subnet-group 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Amazon DocumentDB 서브넷 그룹 수정

다음 modify-db-subnet-group 예시에서는 지정된 서브넷과 새 설명을 추가하여 서브넷 그룹 sample-subnet-group를 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 Modifying an Amazon DocumentDB Subnet Group을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyDbSubnetGroup 섹션을 참조하세요.

reboot-db-instance

다음 코드 예시에서는 `reboot-db-instance` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Amazon DocumentDB 인스턴스 재부팅

다음 `reboot-db-instance` 예시에서는 Amazon DocumentDB 인스턴스 `sample-cluster2`를 재부팅합니다.

```
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 Rebooting an Amazon DocumentDB ILnstance를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RebootDbInstance 섹션을 참조하세요.

remove-tags-from-resource

다음 코드 예시에서는 remove-tags-from-resource 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Amazon DocumentDB 리소스에서 태그 제거

다음 remove-tags-from-resource 예시에서는 Amazon DocumentDB 클러스터 sample-cluster에서 이름이 B인 키가 있는 태그를 제거합니다.

```
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 Removing Tags from an Amazon DocumentDBResource를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RemoveTagsFromResource를 참조하세요.

reset-db-cluster-parameter-group

다음 코드 예시에서는 reset-db-cluster-parameter-group 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Amazon DocumentDB 파라미터 그룹에서 지정된 파라미터 값을 기본값으로 재설정하는 방법

다음 reset-db-cluster-parameter-group 예시에서는 Amazon DocumentDB 파라미터 그룹 custom3-6-param-grp의 ttl_monitor 파라미터를 기본값으로 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 재설정하는 방법

다음 `reset-db-cluster-parameter-group` Amazon DocumentDB 파라미터 그룹 `custom3-6-param-grp`의 모든 파라미터를 기본값으로 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 Resetting an Amazon DocumentDB Cluster Parameter Group을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ResetDbClusterParameterGroup 섹션을 참조하세요.

`restore-db-cluster-from-snapshot`

다음 코드 예시에서는 `restore-db-cluster-from-snapshot` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

자동 또는 수동 스냅샷에서 Amazon DocumentDB 클러스터 복원

다음 `restore-db-cluster-from-snapshot` 예시에서는 스냅샷 `rds:sample-cluster-2019-03-16-00-01`에서 `sample-cluster-2019-03-16-00-01-restored`라는 새 Amazon DocumentDB 클러스터를 생성합니다.

```
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 Restoring from a Cluster Snapshot을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RestoreDbClusterFromSnapshot 섹션을 참조하세요.

restore-db-cluster-to-point-in-time

다음 코드 예시에서는 restore-db-cluster-to-point-in-time 코드를 사용하는 방법을 보여줍니다.

AWS CLI

수동 스냅샷에서 Amazon DocumentDB 클러스터를 특정 시점으로 복원하는 방법

다음 restore-db-cluster-to-point-in-time 예시에서는 sample-cluster-snapshot을 사용하여 최신 복원 가능 시간을 사용하여 새 Amazon DocumentDB 클러스터 sample-cluster-pit를 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 Restoring a Snapshot to a Point in Time을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RestoreDbClusterToPointInTime 섹션을 참조하세요.

start-db-cluster

다음 코드 예시에서는 `start-db-cluster` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

중지된 Amazon DocumentDB 클러스터 시작

다음 `start-db-cluster` 예시에서는 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 Stopping and Starting an Amazon DocumentDB Cluster를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [StartDbCluster](https://docs.aws.amazon.com/goto/aws-cli/docdb-2014-10-31/StartDbCluster) 섹션을 참조하세요.

stop-db-cluster

다음 코드 예시에서는 `stop-db-cluster` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

실행 중인 Amazon DocumentDB 클러스터 중지

다음 `stop-db-cluster` 예시에서는 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 Stopping and Starting an Amazon DocumentDB Cluster를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StopDbCluster 섹션을 참조하세요.

AWS CLI를 사용한 DynamoDB 예제

다음 코드 예는 DynamoDB와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

batch-get-item

다음 코드 예시는 `batch-get-item`의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 BatchGetItem을 참조하세요.

batch-write-item

다음 코드 예시는 batch-write-item의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 BatchWriteItem을 참조하세요.

create-backup

다음 코드 예시는 `create-backup`의 사용 방법을 보여 줍니다.

AWS CLI

기존 DynamoDB 테이블에 대한 백업 생성

다음 `create-backup` 예시에서는 `MusicCollection` 테이블에서 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 온디맨드 DynamoDB 백업 및 복원 사용 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateBackup을 참조하세요.

`create-global-table`

다음 코드 예시는 `create-global-table`의 사용 방법을 보여 줍니다.

AWS CLI

글로벌 테이블 생성

다음 `create-global-table` 예시에서는 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 테이블 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateGlobalTable을 참조하세요.

`create-table`

다음 코드 예시는 `create-table`의 사용 방법을 보여 줍니다.

AWS CLI

예 1: 태그가 포함된 테이블을 생성하는 방법

다음 `create-table` 예시에서는 지정된 속성과 키 스키마를 사용하여 이름이 `MusicCollection`인 테이블을 생성합니다. 이 테이블은 프로비저닝된 처리량을 사용하며, 기본 AWS 소유 CMK를 사용하여 저장 시 암호화됩니다. 이 명령은 또한 키가 `Owner`이고 값이 `blueTeam`인 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.

예 3: 고객 관리형 CMK로 테이블을 생성하고 암호화하는 방법

다음 예시에서는 이름이 `MusicCollection`인 테이블을 만들고 고객 관리형 CMK를 사용하여 이를 암호화합니다.

```
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 테이블은 파티션 키가 UserId이고 정렬 키가 GameTitle이므로 특정 게임의 개별 사용자 최고 점수를 효율적으로 찾을 수 있는 반면 GSI는 파티션 키가 GameTitle이고 정렬 키가 TopScore이므로 특정 게임의 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 `GameScores`인 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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateTable을 참조하세요.

delete-backup

다음 코드 예시는 delete-backup의 사용 방법을 보여 줍니다.

AWS CLI

기존 DynamoDB 백업 삭제 방법

다음 delete-backup 예시에서는 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 온디맨드 DynamoDB 백업 및 복원 사용 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteBackup을 참조하세요.

`delete-item`

다음 코드 예시는 `delete-item`의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteItem을 참조하세요.

delete-table

다음 코드 예시는 delete-table의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteTable](#)을 참조하세요.

describe-backup

다음 코드 예시는 describe-backup의 사용 방법을 보여 줍니다.

AWS CLI

테이블의 기존 백업에 대한 정보를 가져오는 방법

다음 describe-backup 예시는 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 [온디맨드 DynamoDB 백업 및 복원 사용](#) 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeBackup](#)을 참조하세요.

`describe-continuous-backups`

다음 코드 예시는 `describe-continuous-backups`의 사용 방법을 보여 줍니다.

AWS CLI

DynamoDB 테이블의 연속 백업에 대한 정보를 가져오는 방법

다음 `describe-continuous-backups` 예시에서는 `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 DynamoDB 개발자 안내서의 DynamoDB의 시점 백업 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeContinuousBackups를 참조하세요.

describe-contributor-insights

다음 코드 예시는 describe-contributor-insights의 사용 방법을 보여 줍니다.

AWS CLI

DynamoDB 테이블에 대한 Contributor Insights 설정을 보는 방법

다음 describe-contributor-insights 예시에서는 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 분석 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeContributorInsights를 참조하세요.

`describe-endpoints`

다음 코드 예시는 `describe-endpoints`의 사용 방법을 보여 줍니다.

AWS CLI

리전 엔드포인트 정보를 보는 방법

다음 `describe-endpoints` 예시에서는 현재 AWS 리전의 엔드포인트에 대한 세부 정보를 표시합니다.

```
aws dynamodb describe-endpoints
```

출력:

```
{
    "Endpoints": [
        {
            "Address": "dynamodb.us-west-2.amazonaws.com",
            "CachePeriodInMinutes": 1440
        }
    ]
}
```

자세한 내용을 알아보려면 AWS 일반 참조의 [Amazon DynamoDB 엔드포인트 및 할당량]을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeEndpoints]를 참조하세요.

`describe-global-table-settings`

다음 코드 예시는 `describe-global-table-settings`의 사용 방법을 보여 줍니다.

AWS CLI

DynamoDB 글로벌 테이블 설정에 대한 정보를 가져오는 방법

다음 `describe-global-table-settings` 예시에서는 `MusicCollection` 글로벌 테이블의 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 테이블 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeGlobalTableSettings를 참조하세요.

describe-global-table

다음 코드 예시는 `describe-global-table`의 사용 방법을 보여 줍니다.

AWS CLI

DynamoDB 글로벌 테이블에 대한 정보를 표시하는 방법

다음 describe-global-table 예시에서는 MusicCollection 글로벌 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 테이블 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeGlobalTable을 참조하세요.

describe-limits

다음 코드 예시는 describe-limits의 사용 방법을 보여 줍니다.

AWS CLI

프로비저닝된 용량 제한을 보는 방법

다음 describe-limits 예시에서는 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 DynamoDB 내의 제한 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeLimits를 참조하세요.

describe-table-replica-auto-scaling

다음 코드 예시는 describe-table-replica-auto-scaling의 사용 방법을 보여 줍니다.

AWS CLI

글로벌 테이블의 복제본 간에 오토 스케일링 설정을 보는 방법

다음 describe-table-replica-auto-scaling 예시에서는 MusicCollection 글로벌 테이블의 복제본에 대한 오토 스케일링 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 테이블 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeTableReplicaAutoScaling을 참조하세요.

describe-table

다음 코드 예시는 describe-table의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeTable](#)을 참조하세요.

describe-time-to-live

다음 코드 예시는 `describe-time-to-live`의 사용 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 Live](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeTimeToLive](#)를 참조하세요.

get-item

다음 코드 예시는 `get-item`의 사용 방법을 보여 줍니다.

AWS CLI

예 1: 테이블의 항목을 읽는 방법

다음 `get-item` 예시에서는 `MusicCollection` 테이블에서 항목을 검색합니다. 테이블에는 해시 및 범위 프라이머리 키(`Artist` 및 `SongTitle`)가 있으므로 이 두 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.json의 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.json의 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.

- API 세부 정보는 AWS CLI 명령 참조의 GetItem을 참조하세요.

list-backups

다음 코드 예시는 list-backups의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 기존 DynamoDB 백업 모두 나열하는 방법

다음 list-backups 예시에서는 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 온디맨드 DynamoDB 백업 및 복원 사용 섹션을 참조하세요.

예시 2: 특정 시간 범위에서 사용자가 생성한 백업 나열을 나열하는 방법

다음 예시에서는 생성 날짜가 2020년 1월 1일에서 2020년 3월 1일 사이인 사용자가 생성한 `MusicCollection` 테이블의 백업만 나열합니다(DynamoDB에서 자동으로 생성한 백업이 아님).

```
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 온디맨드 DynamoDB 백업 및 복원 사용 섹션을 참조하세요.

예 3: 페이지 크기를 제한하는 방법

다음 예시에서는 모든 기존 백업의 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-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 [온디맨드 DynamoDB 백업 및 복원 사용](https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/BackupRestore.html) 섹션을 참조하세요.

예 4: 반환되는 항목 수를 제한하는 방법

다음 예시에서는 반환되는 항목 수를 1개로 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 [온디맨드 DynamoDB 백업 및 복원 사용] 섹션을 참조하세요.

예 5: 결과의 다음 페이지를 검색하는 방법

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 온디맨드 DynamoDB 백업 및 복원 사용 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListBackups를 참조하세요.

`list-contributor-insights`

다음 코드 예시는 `list-contributor-insights`의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: Contributor Insights 요약 목록 보기

다음 `list-contributor-insights` 예시에서는 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 분석 섹션을 참조하세요.

예 2: 반환되는 항목 수를 제한하는 방법

다음 예시에서는 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 분석 섹션을 참조하세요.

예 3: 결과의 다음 페이지를 검색하는 방법

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 분석 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListContributorInsights를 참조하세요.

`list-global-tables`

다음 코드 예시는 `list-global-tables`의 사용 방법을 보여 줍니다.

AWS CLI

기존 DynamoDB 글로벌 테이블을 나열하는 방법

다음 `list-global-tables` 예시에서는 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 테이블 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListGlobalTables를 참조하세요.

`list-tables`

다음 코드 예시는 `list-tables`의 사용 방법을 보여 줍니다.

AWS CLI

예 1: 테이블을 나열하는 방법

다음 `list-tables` 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListTables를 참조하세요.

`list-tags-of-resource`

다음 코드 예시는 `list-tags-of-resource`의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: DynamoDB 리소스의 태그 나열

다음 `list-tags-of-resource` 예시에서는 `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 DynamoDB에서 태그 지정 섹션을 참조하세요.

예시 2: 반환되는 태그 수 제한

다음 예시에서는 반환되는 태그 수를 1개로 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 DynamoDB에서 태그 지정 섹션을 참조하세요.

예 3: 결과의 다음 페이지를 검색하는 방법

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 DynamoDB에서 태그 지정 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsOfResource를 참조하세요.

put-item

다음 코드 예시는 `put-item`의 사용 방법을 보여 줍니다.

AWS CLI

예 1: 테이블에 항목을 추가하는 방법

다음 `put-item` 예시에서는 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.json`의 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.

- API 세부 정보는 AWS CLI 명령 참조의 PutItem을 참조하세요.

query

다음 코드 예시는 query의 사용 방법을 보여 줍니다.

AWS CLI

예 1: 테이블을 쿼리하는 방법

다음 query 예시에서는 MusicCollection 테이블의 항목을 쿼리합니다. 테이블에는 해시 및 범위 프라이머리 키(Artist 및 SongTitle)가 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 작업](#)을 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.json`의 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 작업]을 참조하세요.

예 3: 특정 결과를 필터링하는 방법

다음 예시에서는 `MusicCollection`을 쿼리하되 `AlbumTitle` 속성에 특정 값이 있는 결과를 제외합니다. 항목을 읽은 후에 필터가 적용되므로 `ScannedCount` 또는 `ConsumedCapacity`에는 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.json`의 콘텐츠:

```
{
    ":v1": {"S": "No One You Know"},
    ":v2": {"S": "Blue Sky Blues"},
    ":v3": {"S": "Greatest Hits"}
}
```

`names.json`의 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 작업](#)을 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.json`의 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 `AlbumTitleIndex`를 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.json`의 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.

- API 세부 정보는 AWS CLI 명령 참조의 Query를 참조하세요.

`restore-table-from-backup`

다음 코드 예시는 `restore-table-from-backup`의 사용 방법을 보여 줍니다.

AWS CLI

기존 백업에서 DynamoDB 테이블을 복원하는 방법

다음 `restore-table-from-backup` 예시에서는 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 온디맨드 DynamoDB 백업 및 복원 사용 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RestoreTableFromBackup을 참조하세요.

`restore-table-to-point-in-time`

다음 코드 예시는 `restore-table-to-point-in-time`의 사용 방법을 보여 줍니다.

AWS CLI

DynamoDB 테이블을 특정 시점으로 복원하려면

다음 `restore-table-to-point-in-time` 예시에서는 `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 DynamoDB 개발자 안내서의 DynamoDB의 시점 백업 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RestoreTableToPointInTime을 참조하세요.

scan

다음 코드 예시는 scan의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 Scan을 참조하세요.

tag-resource

다음 코드 예시는 tag-resource의 사용 방법을 보여 줍니다.

AWS CLI

태그를 DynamoDB 리소스에 추가하는 방법

다음 tag-resource 예시에서는 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 DynamoDB에서 태그 지정 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

transact-get-items

다음 코드 예시는 transact-get-items의 사용 방법을 보여 줍니다.

AWS CLI

하나 이상의 테이블에서 여러 항목을 원자적으로 검색하는 방법

다음 transact-get-items 예시에서는 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 Transactions를 사용하여 복잡한 워크플로 관리 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TransactGetItems를 참조하세요.

transact-write-items

다음 코드 예시는 transact-write-items의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 항목을 하나 이상의 테이블에 원자적으로 쓰기

다음 transact-write-items 예시에서는 한 항목을 업데이트하고 다른 항목을 삭제합니다. 두 작업 중 하나라도 실패하거나 두 항목 중 하나에 Rating 속성이 포함되어 있으면 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 파일의 콘텐츠:

```
[
    {
        "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 Transactions를 사용하여 복잡한 워크플로 관리](#) 섹션을 참조하세요.

예시 2: 클라이언트 요청 토큰을 사용하여 항목을 원자적으로 쓰기

다음 명령은 클라이언트 요청 토큰을 사용하여 `transact-write-items`에 대한 호출에 멱등성을 부여하므로 여러 번의 호출이 한 번의 호출과 동일한 효과를 갖습니다.

```
aws dynamodb transact-write-items \
    --transact-items file://transact-items.json \
    --client-request-token abc123
```

`transact-items.json` 파일의 콘텐츠:

```
[
    {
        "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 Transactions를 사용하여 복잡한 워크플로 관리 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TransactWriteItems를 참조하세요.

untag-resource

다음 코드 예시는 untag-resource의 사용 방법을 보여 줍니다.

AWS CLI

DynamoDB 리소스에서 태그 제거

다음 untag-resource 예시에서는 MusicCollection 테이블에서 Owner 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 DynamoDB에서 태그 지정 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

update-continuous-backups

다음 코드 예시는 update-continuous-backups의 사용 방법을 보여 줍니다.

AWS CLI

DynamoDB 테이블에 대한 연속 백업 설정을 업데이트하려면

다음 `update-continuous-backups` 예시에서는 `MusicCollection` 테이블에 대한 특정 시점 복구를 활성화합니다.

```
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 DynamoDB 개발자 안내서의 DynamoDB의 시점 백업 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateContinuousBackups를 참조하세요.

`update-contributor-insights`

다음 코드 예시는 `update-contributor-insights`의 사용 방법을 보여 줍니다.

AWS CLI

테이블에 Contributor Insights 활성화

다음 `update-contributor-insights` 예시에서는 `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 분석 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateContributorInsights를 참조하세요.

`update-global-table-settings`

다음 코드 예시는 `update-global-table-settings`의 사용 방법을 보여 줍니다.

AWS CLI

DynamoDB 글로벌 테이블에서 프로비저닝된 쓰기 용량 설정을 업데이트하는 방법

다음 `update-global-table-settings` 예시에서는 `MusicCollection` 글로벌 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 테이블 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateGlobalTableSettings를 참조하세요.

`update-global-table`

다음 코드 예시는 `update-global-table`의 사용 방법을 보여 줍니다.

AWS CLI

DynamoDB 글로벌 테이블을 업데이트하는 방법

다음 update-global-table 예시에서는 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 테이블 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateGlobalTable을 참조하세요.

update-item

다음 코드 예시는 update-item의 사용 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.json`의 콘텐츠:

```
{
    "Artist": {"S": "Acme Band"},
    "SongTitle": {"S": "Happy Day"}
}
```

`expression-attribute-names.json`의 콘텐츠:

```
{
    "#Y":"Year", "#AT":"AlbumTitle"
}
```

`expression-attribute-values.json`의 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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateItem을 참조하세요.

update-table-replica-auto-scaling

다음 코드 예시는 update-table-replica-auto-scaling의 사용 방법을 보여 줍니다.

AWS CLI

글로벌 테이블의 복제본 간에 오토 스케일링 설정을 업데이트하는 방법

다음 update-table-replica-auto-scaling 예시에서는 지정된 글로벌 테이블의 복제본에 걸쳐 쓰기 용량 오토 스케일링 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 테이블 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateTableReplicaAutoScaling을 참조하세요.

update-table

다음 코드 예시는 update-table의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateTable을 참조하세요.

`update-time-to-live`

다음 코드 예시는 `update-time-to-live`의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateTimeToLive를 참조하세요.

AWS CLI를 사용한 DynamoDB Streams 예제

다음 코드 예제에서는 DynamoDB Streams에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여 줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`describe-stream`

다음 코드 예제에서는 `describe-stream`의 사용 방법을 보여줍니다.

AWS CLI

DynamoDB 테이블에 대한 정보를 가져오려면

다음 `describe-stream` 명령은 특정 DynamoDB Streams에 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": "4000000000000642977831",
                    "EndingSequenceNumber": "4000000000000642977831"
                }
            },
            {
                "ShardId": "shardId-00000001571780995058-40810d86",
                "SequenceNumberRange": {
                    "StartingSequenceNumber": "757400000000005655171150"
                },
                "ParentShardId": "shardId-00000001571767321804-697ce3d2"
            }
        ]
```

```
    }
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DynamoDB Streams를 사용하여 Table Activity 캡처를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeStream을 참조하세요.

get-records

다음 코드 예제에서는 `get-records`의 사용 방법을 보여줍니다.

AWS CLI

Dynamodb Streams에서 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 Streams를 사용하여 Table Activity 캡처](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [GetRecords](#)를 참조하세요.

`get-shard-iterator`

다음 코드 예제에서는 `get-shard-iterator`의 사용 방법을 보여줍니다.

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 Streams를 사용하여 Table Activity 캡처]를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [GetShardIterator]를 참조하세요.

`list-streams`

다음 코드 예제에서는 `list-streams`의 사용 방법을 보여줍니다.

AWS CLI

DynamoDB Streams를 나열하려면

다음 `list-streams` 명령은 기본 AWS 리전 내의 기존 Amazon DynamoDB Streams를 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 Streams를 사용하여 Table Activity 캡처를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListStreams를 참조하세요.

AWS CLI를 사용한 Amazon EC2 예제

다음 코드 예제는 Amazon EC2와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

accept-address-transfer

다음 코드 예시는 `accept-address-transfer`의 사용 방법을 보여 줍니다.

AWS CLI

탄력적 IP 주소가 계정으로 전송되는 것을 허용

다음 accept-address-transfer 예시에서는 지정된 탄력적 IP 주소가 계정으로 전송되는 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.

- API 세부 정보는 AWS CLI 명령 참조의 AcceptAddressTransfer 섹션을 참조하세요.

accept-reserved-instances-exchange-quote

다음 코드 예시는 accept-reserved-instances-exchange-quote의 사용 방법을 보여 줍니다.

AWS CLI

전환형 예약 인스턴스 교환 수행

이 예시에서는 지정된 전환형 예약 인스턴스의 교환을 수행합니다.

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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 AcceptReservedInstancesExchangeQuote 섹션을 참조하세요.

accept-transit-gateway-peering-attachment

다음 코드 예시는 accept-transit-gateway-peering-attachment의 사용 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 피어링 연결 허용

다음 accept-transit-gateway-peering-attachment 예시에서는 지정된 전송 게이트웨이의 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 설명서의 Transit Gateway Peering Attachments 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AcceptTransitGatewayPeeringAttachment 섹션을 참조하세요.

accept-transit-gateway-vpc-attachment

다음 코드 예시는 `accept-transit-gateway-vpc-attachment`의 사용 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이에 VPC 연결 요청 수락

다음 `accept-transit-gateway-vpc-attachment` 예시에서는 특정 연결에 대한 요청을 허용합니다.

```
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 설명서의 [Transit Gateway Attachments to a VPC](#) 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [AcceptTransitGatewayVpcAttachment](#) 섹션을 참조하세요.

`accept-vpc-endpoint-connections`

다음 코드 예시는 `accept-vpc-endpoint-connections`의 사용 방법을 보여 줍니다.

AWS CLI

인터페이스 엔드포인트 연결 요청 수락

이 예시에서는 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": []
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [AcceptVpcEndpointConnections](#) 섹션을 참조하세요.

`accept-vpc-peering-connection`

다음 코드 예시는 `accept-vpc-peering-connection`의 사용 방법을 보여 줍니다.

AWS CLI

VPC 피어링 연결 허용

이 예시에서는 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"
    }
  }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 AcceptVpcPeeringConnection 섹션을 참조하세요.

advertise-byoip-cidr

다음 코드 예시는 advertise-byoip-cidr의 사용 방법을 보여 줍니다.

AWS CLI

주소 범위 광고

다음 advertise-byoip-cidr 예시에서는 지정된 퍼블릭 IPv4 주소 범위를 알립니다.

```
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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 AdvertiseByoipCidr 섹션을 참조하세요.

allocate-address

다음 코드 예시는 allocate-address의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: Amazon 주소 풀에서 탄력적 IP 주소를 할당하는 방법

다음 allocate-address 예제는 탄력적 IP 주소를 할당합니다. Amazon EC2는 Amazon 주소 풀에서 주소를 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 설명서에서 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 설명서에서 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. Amazon EC2는 주소 풀에서 주소를 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 설명서에서 탄력적 IP 주소를 참조하세요.

예제 4: IPAM 풀에서 탄력적 IP 주소 할당

다음 `allocate-address` 예제에서는 Amazon VPC IP 주소 관리자(IPAM) 풀에서 특정 /32 탄력적 IP 주소를 할당합니다.

```
aws ec2 allocate-address \
    --region us-east-1 \
    --ipam-pool-id ipam-pool-1234567890abcdef0 \
    --address 192.0.2.0
```

출력:

```
{
    "PublicIp": "192.0.2.0",
    "AllocationId": "eipalloc-abcdef01234567890",
    "PublicIpv4Pool": "ipam-pool-1234567890abcdef0",
    "NetworkBorderGroup": "us-east-1",
    "Domain": "vpc"
}
```

자세한 내용은 Amazon VPC IPAM 사용 설명서의 IPAM 풀에서 순차적 탄력적 IP 주소 할당을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AllocateAddress 섹션을 참조하세요.

allocate-hosts

다음 코드 예시는 allocate-hosts의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 전용 호스트 할당

다음 allocate-hosts 예시에서는 m5.large 인스턴스를 시작할 수 있는 eu-west-1a 가용 영역에 단일 전용 호스트를 할당합니다. 기본적으로 전용 호스트는 대상 인스턴스 시작만 허용하며 호스트 복구는 지원하지 않습니다.

```
aws ec2 allocate-hosts \
    --instance-type m5.large \
    --availability-zone eu-west-1a \
    --quantity 1
```

출력:

```
{
    "HostIds": [
        "h-07879acf49EXAMPLE"
    ]
}
```

예시 2: 자동 배치 및 호스트 복구가 활성화된 전용 호스트 할당

다음 allocate-hosts 예시에서는 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"
     ]
}
```

예시 3: 태그가 있는 전용 호스트 할당

다음 allocate-hosts 예시에서는 단일 전용 호스트를 할당하고 키 이름이 purpose, 값이 production인 태그를 적용합니다

```
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 Amazon EC2 사용 설명서의 전용 호스트 할당을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AllocateHosts 섹션을 참조하세요.

`allocate-ipam-pool-cidr`

다음 코드 예시는 `allocate-ipam-pool-cidr`의 사용 방법을 보여 줍니다.

AWS CLI

IPAM 풀에서 CIDR 할당

다음 `allocate-ipam-pool-cidr` 예시에서는 IPAM 풀에서 CIDR을 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 [풀에 CIDR을 수동으로 할당하여 IP 주소 공간 예약] 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [AllocateIpamPoolCidr] 섹션을 참조하세요.

`apply-security-groups-to-client-vpn-target-network`

다음 코드 예시는 `apply-security-groups-to-client-vpn-target-network`의 사용 방법을 보여 줍니다.

AWS CLI

Client VPN 엔드포인트의 대상 네트워크에 보안 그룹 적용

다음 `apply-security-groups-to-client-vpn-target-network` 예시에서는 대상 네트워크와 Client VPN 엔드포인트 간 연결에 `sg-01f6e627a89f4db32` 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 Client VPN 관리자 안내서의 Target Networks 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ApplySecurityGroupsToClientVpnTargetNetwork 섹션을 참조하세요.

`assign-ipv6-addresses`

다음 코드 예시는 `assign-ipv6-addresses`의 사용 방법을 보여 줍니다.

AWS CLI

네트워크 인터페이스에 특정 IPv6 주소 할당

이 예시에서는 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 할당

이 예시에서는 지정된 네트워크 인터페이스에 두 개의 IPv6 주소를 할당합니다. Amazon은 서브넷의 IPv6 CIDR 블록 범위에서 사용 가능한 IPv6 주소 중에서 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 세부 정보는 AWS CLI 명령 참조의 AssignIpv6Addresses 섹션을 참조하세요.

`assign-private-ip-addresses`

다음 코드 예시는 `assign-private-ip-addresses`의 사용 방법을 보여 줍니다.

AWS CLI

특정 보조 프라이빗 IP 주소에 네트워크 인터페이스 할당

이 예시에서는 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
```

Amazon EC2가 선택한 보조 프라이빗 IP 주소를 네트워크 인터페이스 할당

이 예시에서는 지정된 네트워크 인터페이스에 두 개의 보조 프라이빗 IP 주소를 할당합니다. Amazon EC2는 네트워크 인터페이스가 연결된 서브넷의 CIDR 블록 범위에서 사용 가능한 IP 주소 중에서 이러한 IP 주소를 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
```

- API 세부 정보는 AWS CLI 명령 참조의 AssignPrivateIpAddresses 섹션을 참조하세요.

`assign-private-nat-gateway-address`

다음 코드 예시는 `assign-private-nat-gateway-address`의 사용 방법을 보여 줍니다.

AWS CLI

프라이빗 NAT 게이트웨이에 프라이빗 IP 주소 할당

다음 `assign-private-nat-gateway-address` 예시에서는 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 정보는 Amazon VPC 사용 설명서의 NAT 게이트웨이 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AssignPrivateNatGatewayAddress 섹션을 참조하세요.

associate-address

다음 코드 예시는 `associate-address`의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 인스턴스와 탄력적 IP 주소 연결

다음 `associate-address` 예제에서는 지정된 EC2 인스턴스와 탄력적 IP 주소를 연결합니다.

```
aws ec2 associate-address \
    --instance-id i-0b263919b6498b123 \
    --allocation-id eipalloc-64d5890a
```

출력:

```
{
    "AssociationId": "eipassoc-2bebb745"
}
```

예제 2: 네트워크 인터페이스와 탄력적 IP 주소 연결

다음 `associate-address` 예제에서는 지정된 탄력적 IP 주소와 지정된 네트워크 인터페이스를 연결합니다.

```
aws ec2 associate-address
    --allocation-id eipalloc-64d5890a \
    --network-interface-id eni-1a2b3c4d
```

출력:

```
{
    "AssociationId": "eipassoc-2bebb745"
}
```

예제 3: 탄력적 IP 주소와 프라이빗 IP 주소 연결

다음 `associate-address` 예제에서는 지정된 탄력적 IP 주소를 지정된 네트워크 인터페이스의 지정된 프라이빗 IP 주소와 연결합니다.

```
aws ec2 associate-address \
    --allocation-id eipalloc-64d5890a \
    --network-interface-id eni-1a2b3c4d \
    --private-ip-address 10.0.0.85
```

출력:

```
{
    "AssociationId": "eipassoc-2bebb745"
}
```

자세한 내용은 Amazon EC2 사용 설명서에서 탄력적 IP 주소를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AssociateAddress 섹션을 참조하세요.

`associate-client-vpn-target-network`

다음 코드 예시는 `associate-client-vpn-target-network`의 사용 방법을 보여 줍니다.

AWS CLI

클라이언트 VPN 엔드포인트와 대상 네트워크 연결

다음 `associate-client-vpn-target-network` 예시에서는 지정된 클라이언트 VPN 엔드포인트와 서브넷을 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 Client VPN 관리자 안내서의 Target Networks 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AssociateClientVpnTargetNetwork 섹션을 참조하세요.

associate-dhcp-options

다음 코드 예시는 `associate-dhcp-options`의 사용 방법을 보여 줍니다.

AWS CLI

VPC와 DHCP 옵션 세트 연결

이 예시에서는 지정된 DHCP 옵션 세트를 지정된 VPC와 연결합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 associate-dhcp-options --dhcp-options-id dopt-d9070ebb --vpc-id vpc-a01106c2
```

VPC와 기본 DHCP 옵션 세트 연결

이 예시에서는 기본 DHCP 옵션 설정을 지정된 VPC와 연결합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 associate-dhcp-options --dhcp-options-id default --vpc-id vpc-a01106c2
```

- API 세부 정보는 AWS CLI 명령 참조의 AssociateDhcpOptions 섹션을 참조하세요.

associate-iam-instance-profile

다음 코드 예시는 `associate-iam-instance-profile`의 사용 방법을 보여 줍니다.

AWS CLI

인스턴스와 IAM 인스턴스 프로파일 연결

이 예시에서는 IAM 인스턴스 프로파일 admin-role을 인스턴스 i-123456789abcde123과 연결합니다.

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"
      }
  }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 AssociateIamInstanceProfile 섹션을 참조하세요.

associate-instance-event-window

다음 코드 예시는 associate-instance-event-window의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 이벤트 기간에 하나 이상의 인스턴스 연결

다음 associate-instance-event-window 예시에서는 하나 이상의 인스턴스를 이벤트 기간과 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 사항 섹션을 참조하세요.

예시 2: 이벤트 기간에 인스턴스 태그 연결

다음 `associate-instance-event-window` 예시에서는 인스턴스 태그를 이벤트 기간과 연결합니다. `instance-event-window-id` 파라미터를 입력하여 이벤트 기간을 지정합니다. 인스턴스 태그를 연결하려면 `association-target` 파라미터를 지정하고 파라미터 값으로 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 사항 섹션을 참조하세요.

예시 3: 이벤트 기간에 전용 호스트 연결

다음 `associate-instance-event-window` 예시에서는 전용 호스트를 이벤트 기간 연결합니다. `instance-event-window-id` 파라미터를 입력하여 이벤트 기간을 지정합니다. 전용 호스트를 연결하려면 `--association-target` 파라미터를 지정하고 파라미터 값에 대해 전용 호스트 ID 중 하나를 지정합니다.

```
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 사항 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AssociateInstanceEventWindow 섹션을 참조하세요.

associate-ipam-resource-discovery

다음 코드 예시는 associate-ipam-resource-discovery의 사용 방법을 보여 줍니다.

AWS CLI

IPAM과 리소스 검색 연결

이 예시에서는 IPAM 위임된 관리자인 사용자가 다른 AWS 계정에서 리소스 검색을 만들어 공유했으므로 다른 계정에서 소유한 리소스 CIDR을 관리하고 모니터링하는 데 IPAM을 사용할 수 있습니다.

Note

이 요청을 완료하려면 describe-ipam-resource-discoverie로 얻을 수 있는 리소스 검색 ID와 describe-ipams로 얻을 수 있는 IPAM ID가 필요하며, 연결하려는 리소스 검색은 먼저 AWS RAM을 사용하여 계정과 공유된 것이어야 합니다. 입력하는 --region은 연결하려는 IPAM의 홈 리전과 일치해야 합니다.

다음 associate-ipam-resource-discovery 예시에서는 리소스 검색을 IPAM과 연결합니다.

```
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 연결한 후에는 다른 계정에서 만든 리소스의 IP 주소를 모니터링 및/또는 관리할 수 있습니다. 자세한 내용은 Amazon VPC IPAM 사용 설명서의 IPAM을 조직 외부 계정과 통합 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AssociateIpamResourceDiscovery 섹션을 참조하세요.

associate-nat-gateway-address

다음 코드 예시는 `associate-nat-gateway-address`의 사용 방법을 보여 줍니다.

AWS CLI

퍼블릭 NAT 게이트웨이와 탄력적 IP 주소 연결

다음 `associate-nat-gateway-address` 예시에서는 지정된 탄력적 IP 주소를 지정된 퍼블릭 NAT 게이트웨이와 연결합니다. AWS는 자동으로 보조 프라이빗 IPv4 주소를 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 정보는 Amazon VPC 사용 설명서의 NAT 게이트웨이 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AssociateNatGatewayAddress 섹션을 참조하세요.

associate-route-table

다음 코드 예시는 `associate-route-table`의 사용 방법을 보여 줍니다.

AWS CLI

서브넷에 라우팅 테이블 연결

이 예시에서는 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 AssociateRouteTable 섹션을 참조하세요.

associate-security-group-vpc

다음 코드 예시는 associate-security-group-vpc의 사용 방법을 보여 줍니다.

AWS CLI

보안 그룹을 다른 VPC와 연결

다음 associate-security-group-vpc 예제에서는 지정된 VPC를 지정된 보안 그룹에 연결합니다.

```
aws ec2 associate-security-group-vpc \
    --group-id sg-04dbb43907d3f8a78 \
    --vpc-id vpc-0bf4c2739bc05a694
```

출력:

```
{
    "State": "associating"
}
```

자세한 내용은 Amazon VPC 사용 설명서의 보안 그룹을 여러 VPC와 연결을 참조하시기 바랍니다.

- API 세부 정보는 AWS CLI 명령 참조의 AssociateSecurityGroupVpc를 참조하시기 바랍니다.

associate-subnet-cidr-block

다음 코드 예시는 associate-subnet-cidr-block의 사용 방법을 보여 줍니다.

AWS CLI

서브넷에 IPv6 CIDR 블록 연결

이 예시에서는 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"
      }
  }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 AssociateSubnetCidrBlock 섹션을 참조하세요.

associate-transit-gateway-multicast-domain

다음 코드 예시는 associate-transit-gateway-multicast-domain의 사용 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이를 멀티캐스트 도메인에 연결

다음 associate-transit-gateway-multicast-domain 예시에서는 지정된 서브넷 및 첨부 파일을 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 설명서의 멀티캐스트 도메인 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AssociateTransitGatewayMulticastDomain 섹션을 참조하세요.

associate-transit-gateway-route-table

다음 코드 예시는 `associate-transit-gateway-route-table`의 사용 방법을 보여 줍니다.

AWS CLI

Transit Gateway Attachment에 전송 게이트웨이 라우팅 테이블 연결

다음 예시에서는 지정한 VPC 연결과 지정한 전송 게이트웨이 라우팅 테이블을 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 AWS Transit Gateways 설명서의 Associate a Transit Gateway Route Table 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AssociateTransitGatewayRouteTable 섹션을 참조하세요.

associate-vpc-cidr-block

다음 코드 예시는 associate-vpc-cidr-block의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: Amazon에서 제공하는 IPv6 CIDR 블록을 VPC와 연결

다음 associate-vpc-cidr-block 예시에서는 IPv6 CIDR 블록을 지정된 VPC와 연결합니다.

```
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"
}
```

예시 2:추가 IPv4 CIDR 블록을 VPC와 연결

다음 associate-vpc-cidr-block 예시에서는 IPv4 CIDR 블록 10.2.0.0/16을 지정된 VPC와 연결합니다.

```
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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 AssociateVpcCidrBlock 섹션을 참조하세요.

`attach-classic-link-vpc`

다음 코드 예시는 `attach-classic-link-vpc`의 사용 방법을 보여 줍니다.

AWS CLI

EC2-Classic 인스턴스를 VPC에 연결

이 예시에서는 인스턴스 i-1234567890abcdef0을 VPC 보안 그룹 sg-12312312을 통해 VPC vpc-88888888에 연결합니다.

명령:

```
aws ec2 attach-classic-link-vpc --instance-id  i-1234567890abcdef0 --vpc-id vpc-88888888 --groups sg-12312312
```

출력:

```
{
  "Return": true
}
```

- API 세부 정보는 AWS CLI 명령 참조의 AttachClassicLinkVpc 섹션을 참조하세요.

`attach-internet-gateway`

다음 코드 예시는 `attach-internet-gateway`의 사용 방법을 보여 줍니다.

AWS CLI

VPC에 인터넷 게이트웨이 연결

다음 `attach-internet-gateway` 예시에서는 지정된 인터넷 게이트웨이를 특정 VPC에 연결합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 AttachInternetGateway 섹션을 참조하세요.

`attach-network-interface`

다음 코드 예시는 `attach-network-interface`의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 인스턴스에 네트워크 인터페이스 연결

다음 `attach-network-interface` 예시에서는 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.

예시 2: 여러 네트워크 카드가 있는 인스턴스에 네트워크 인터페이스 연결

다음 `attach-network-interface` 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 AttachNetworkInterface 섹션을 참조하세요.

attach-verified-access-trust-provider

다음 코드 예시는 `attach-verified-access-trust-provider`의 사용 방법을 보여 줍니다.

AWS CLI

인스턴스에 신뢰 공급자 연결

다음 `attach-verified-access-trust-provider` 예시에서는 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 AWS Verified Access 사용 설명서의 Verified Access instances 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AttachVerifiedAccessTrustProvider 섹션을 참조하세요.

`attach-volume`

다음 코드 예시는 `attach-volume`의 사용 방법을 보여 줍니다.

AWS CLI

인스턴스에 볼륨 연결

이 예시에서는 볼륨(`vol-1234567890abcdef0`)을 인스턴스(`i-01474ef662b89480`)에 `/dev/sdf`로 첨부합니다.

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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 AttachVolume 섹션을 참조하세요.

attach-vpn-gateway

다음 코드 예시는 attach-vpn-gateway의 사용 방법을 보여 줍니다.

AWS CLI

VPC에 가상 프라이빗 게이트웨이 연결

다음 attach-vpn-gateway 예시에서는 지정된 가상 프라이빗 게이트웨이를 지정된 VPC에 연결합니다.

```
aws ec2 attach-vpn-gateway \
    --vpn-gateway-id vgw-9a4cacf3 \
    --vpc-id vpc-a01106c2
```

출력:

```
{
    "VpcAttachment": {
        "State": "attaching",
        "VpcId": "vpc-a01106c2"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 AttachVpnGateway 섹션을 참조하세요.

authorize-client-vpn-ingress

다음 코드 예시는 authorize-client-vpn-ingress의 사용 방법을 보여 줍니다.

AWS CLI

Client VPN 엔드포인트에 권한 부여 규칙 추가

다음 `authorize-client-vpn-ingress` 예시에서는 모든 클라이언트가 인터넷(`0.0.0.0/0`)에 액세스할 수 있도록 허용하는 수신 권한 부여 규칙을 추가합니다.

```
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 Client VPN 관리자 안내서의 [권한 부여 규칙]을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [AuthorizeClientVpnIngress] 섹션을 참조하세요.

`authorize-security-group-egress`

다음 코드 예시는 `authorize-security-group-egress`의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 아웃바운드 트래픽을 특정 주소 범위로 허용하는 규칙 추가

다음 `authorize-security-group-egress` 예제에서는 TCP 포트 80에서 지정된 주소 범위에 대한 액세스 권한을 부여하는 규칙을 추가합니다.

```
aws ec2 authorize-security-group-egress \
    --group-id sg-1234567890abcdef0 \
    --ip-permissions
 'IpProtocol=tcp,FromPort=80,ToPort=80,IpRanges=[{CidrIp=10.0.0.0/16}]'
```

출력:

```
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-0b15794cdb17bf29c",
            "GroupId": "sg-1234567890abcdef0",
            "GroupOwnerId": "123456789012",
            "IsEgress": true,
            "IpProtocol": "tcp",
            "FromPort": 80,
            "ToPort": 80,
            "CidrIpv4": "10.0.0.0/16"
        }
    ]
}
```

예제 2: 특정 보안 그룹에 아웃바운드 트래픽을 허용하는 규칙 추가

다음 `authorize-security-group-egress` 예제에서는 TCP 포트 80에서 지정된 보안 그룹에 대한 액세스 권한을 부여하는 규칙을 추가합니다.

```
aws ec2 authorize-security-group-egress \
    --group-id sg-1234567890abcdef0 \
    --ip-permissions
 'IpProtocol=tcp,FromPort=80,ToPort=80,UserIdGroupPairs=[{GroupId=sg-0aad1c26bbeec5c22}]'
```

출력:

```
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-0b5dd815afcea9cc3",
            "GroupId": "sg-1234567890abcdef0",
            "GroupOwnerId": "123456789012",
            "IsEgress": true,
            "IpProtocol": "tcp",
            "FromPort": 80,
            "ToPort": 80,
            "ReferencedGroupInfo": {
                "GroupId": "sg-0aad1c26bbeec5c22",
                "UserId": "123456789012"
```

```
            }
        }
    ]
}
```

자세한 내용을 알아보려면 Amazon VPC 사용 설명서의 보안 그룹을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AuthorizeSecurityGroupEgress 섹션을 참조하세요.

authorize-security-group-ingress

다음 코드 예시는 authorize-security-group-ingress의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 인바운드 SSH 트래픽을 허용하는 규칙을 추가하는 방법

다음 authorize-security-group-ingress 예제에서는 TCP 포트 22(SSH)의 인바운드 트래픽을 허용하는 규칙을 추가합니다.

```
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 추가하는 방법

다음 `authorize-security-group-ingress` 예제에서는 소스 보안 그룹 `sg-1a2b3c4d`에서 TCP 포트 80의 인바운드 액세스를 허용하는 규칙을 추가합니다. 보안 그룹은 동일한 VPC 또는 피어 VPC에 있어야 합니다(VPC 피어링 연결이 필요함). 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)의 인바운드 액세스를 허용하는 하나의 인바운드 규칙과 Ping/ICMP를 허용하는 다른 인바운드 규칙(총 2개)을 추가합니다.

```
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 추가하는 방법

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 규칙을 추가하는 방법

다음 `authorize-security-group-ingress` 예제에서는 `ip-permissions` 파라미터를 사용하여 IPv6 범위 `2001:db8:1234:1a00::/64`에서 SSH 액세스(포트 22)를 허용하는 인바운드 규칙을 추가합니다.

```
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 추가하는 방법

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 범위에서 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
 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 목록에 있는 CIDR 범위의 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 내용을 알아보려면 Amazon VPC 사용 설명서의 보안 그룹을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AuthorizeSecurityGroupIngress를 참조하세요.

bundle-instance

다음 코드 예시는 bundle-instance의 사용 방법을 보여 줍니다.

AWS CLI

인스턴스 번들링

이 예시에서는 i-1234567890abcdef0 인스턴스를 bundletasks 버킷에 번들로 묶습니다. 액세스 키 ID의 값을 지정하기 전에 AWS 액세스 키 관리를 위한 모범 사례의 지침을 검토하고 따르세요.

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"
  }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 BundleInstance 섹션을 참조하세요.

cancel-bundle-task

다음 코드 예시는 cancel-bundle-task의 사용 방법을 보여 줍니다.

AWS CLI

번들 작업 취소

이 예시에서는 번들 태스크 bun-2a4e041c를 취소합니다.

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"
  }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CancelBundleTask 섹션을 참조하세요.

`cancel-capacity-reservation-fleets`

다음 코드 예시는 `cancel-capacity-reservation-fleets`의 사용 방법을 보여 줍니다.

AWS CLI

용량 예약 플릿 취소

다음 `cancel-capacity-reservation-fleets` 예시에서는 지정된 용량 예약 플릿과 해당 플릿이 예약한 용량을 취소합니다. 플릿을 취소하면 `cancelled` 상태로 바뀌고 더 이상 새 용량 예약을 생성할 수 없습니다. 또한 플릿의 모든 개별 용량 예약이 취소되고 이전에 예약 용량에서 실행 중이었던 인스턴스는 공유 용량에서 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 예약에 대한 자세한 내용은 Amazon EC2 사용 설명서의 용량 예약 플릿을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CancelCapacityReservationFleets 섹션을 참조하세요.

`cancel-capacity-reservation`

다음 코드 예시는 `cancel-capacity-reservation`의 사용 방법을 보여 줍니다.

AWS CLI

용량 예약 취소

다음 `cancel-capacity-reservation` 예시에서는 지정된 용량 예약을 취소합니다.

```
aws ec2 cancel-capacity-reservation \
    --capacity-reservation-id cr-1234abcd56EXAMPLE
```

출력:

```
{
    "Return": true
}
```

자세한 내용은 Amazon EC2 사용 설명서의 용량 예약 취소를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CancelCapacityReservation 섹션을 참조하세요.

cancel-conversion-task

다음 코드 예시는 cancel-conversion-task의 사용 방법을 보여 줍니다.

AWS CLI

인스턴스 또는 볼륨의 활성 변환 취소

이 예시에서는 태스크 ID import-i-fh95npoc과 관련된 업로드를 취소합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 cancel-conversion-task --conversion-task-id import-i-fh95npoc
```

- API 세부 정보는 AWS CLI 명령 참조의 CancelConversionTask 섹션을 참조하세요.

cancel-export-task

다음 코드 예시는 cancel-export-task의 사용 방법을 보여 줍니다.

AWS CLI

활성 내보내기 태스크 취소

이 예시에서는 태스크 ID가 export-i-fgelt0i7인 활성 내보내기 작업을 취소합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 cancel-export-task --export-task-id export-i-fgelt0i7
```

- API 세부 정보는 AWS CLI 명령 참조의 CancelExportTask 섹션을 참조하세요.

cancel-image-launch-permission

다음 코드 예시는 cancel-image-launch-permission의 사용 방법을 보여 줍니다.

AWS CLI

Amazon Web Services 계정과 공유된 AMI 취소

다음 cancel-image-launch-permission 예시에서는 지정된 AMI의 시작 권한에서 계정을 제거합니다.

```
aws ec2 cancel-image-launch-permission \
    --image-id ami-0123456789example \
    --region us-east-1
```

출력:

```
{
    "Return": true
}
```

자세한 내용은 Amazon EC2 사용 설명서의 Amazon Web Services 계정과 AMI 공유 취소를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CancelImageLaunchPermission 섹션을 참조하세요.

cancel-import-task

다음 코드 예시는 cancel-import-task의 사용 방법을 보여 줍니다.

AWS CLI

가져오기 태스크 취소

다음 cancel-import-task 예시에서는 지정된 이미지 가져오기 태스크를 취소합니다.

```
aws ec2 cancel-import-task \
    --import-task-id import-ami-1234567890abcdef0
```

출력:

```
{
    "ImportTaskId": "import-ami-1234567890abcdef0",
    "PreviousState": "active",
    "State": "deleting"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CancelImportTask 섹션을 참조하세요.

cancel-reserved-instances-listing

다음 코드 예시는 cancel-reserved-instances-listing의 사용 방법을 보여 줍니다.

AWS CLI

예약 인스턴스 목록 취소

다음 cancel-reserved-instances-listing 예시에서는 지정된 예약 인스턴스 목록을 취소합니다.

```
aws ec2 cancel-reserved-instances-listing \
    --reserved-instances-listing-id 5ec28771-05ff-4b9b-aa31-9e57dexample
```

- API 세부 정보는 AWS CLI 명령 참조의 CancelReservedInstancesListing 섹션을 참조하세요.

cancel-spot-fleet-requests

다음 코드 예시는 cancel-spot-fleet-requests의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 스팟 플릿 요청을 취소하고 연결된 인스턴스 종료

다음 cancel-spot-fleet-requests 예시에서는 스팟 플릿 요청을 취소하고 관련된 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": []
}
```

예시 2: 관련 인스턴스를 종료하지 않고 스팟 플릿 요청 취소

다음 `cancel-spot-fleet-requests` 예시에서는 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 Amazon EC2 사용 설명서의 스팟 플릿 요청 취소를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CancelSpotFleetRequests 섹션을 참조하세요.

`cancel-spot-instance-requests`

다음 코드 예시는 `cancel-spot-instance-requests`의 사용 방법을 보여 줍니다.

AWS CLI

스팟 인스턴스 요청 취소

이 예시에서는 스팟 인스턴스 요청을 취소합니다.

명령:

```
aws ec2 cancel-spot-instance-requests --spot-instance-request-ids sir-08b93456
```

출력:

```
{
    "CancelledSpotInstanceRequests": [
        {
            "State": "cancelled",
            "SpotInstanceRequestId": "sir-08b93456"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [CancelSpotInstanceRequests](#) 섹션을 참조하세요.

`confirm-product-instance`

다음 코드 예시는 `confirm-product-instance`의 사용 방법을 보여 줍니다.

AWS CLI

제품 인스턴스 확인

이 예시에서는 지정된 제품 코드가 지정된 인스턴스와 연관되어 있는지 여부를 확인합니다.

명령:

```
aws ec2 confirm-product-instance --product-code 774F4FF8 --instance-id i-1234567890abcdef0
```

출력:

```
{
  "OwnerId": "123456789012"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ConfirmProductInstance 섹션을 참조하세요.

copy-fpga-image

다음 코드 예시는 copy-fpga-image의 사용 방법을 보여 줍니다.

AWS CLI

Amazon FPGA 이미지 복사

이 예시에서는 us-east-1 리전에서 현재 리전(eu-west-1)으로 지정된 AFI를 복사합니다.

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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CopyFpgaImage 섹션을 참조하세요.

copy-image

다음 코드 예시는 copy-image의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 다른 리전에 AMI 복사

다음 copy-image 명령은 지정된 AMI를 us-west-2 리전에서 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 AMI 복사를 참조하세요.

예시 2: 다른 리전에 AMI 복사 및 백업 스냅샷 암호화

다음 `copy-image` 명령은 `us-west-2` 리전에서 현재 리전으로 지정한 AMI를 복사하고 지정한 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 AMI 복사를 참조하세요.

예시 3: AMI를 복사할 때 사용자 정의 AMI 태그 포함

다음 `copy-image` 명령은 AMI를 복사할 때 `--copy-image-tags` 파라미터를 사용하여 사용자 정의 AMI 태그를 복사합니다.

```
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 AMI 복사를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CopyImage 섹션을 참조하세요.

copy-snapshot

다음 코드 예시는 copy-snapshot의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 다른 리전에 스냅샷 복사

다음 copy-snapshot 예시 명령은 us-west-2 리전에서 us-east-1 리전으로 지정된 스냅샷을 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"
}
```

예시 2: 암호화되지 않은 스냅샷 복사 및 새 스냅샷 암호화

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 EBS 사용 설명서의 Amazon EBS 스냅샷 복사를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CopySnapshot 섹션을 참조하세요.

`create-capacity-reservation-fleet`

다음 코드 예시는 `create-capacity-reservation-fleet`의 사용 방법을 보여 줍니다.

AWS CLI

용량 예약 플릿 생성

다음 `create-capacity-reservation-fleet` 예시에서는 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 예약에 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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateCapacityReservationFleet 섹션을 참조하세요.

create-capacity-reservation

다음 코드 예시는 create-capacity-reservation의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 용량 예약 생성

다음 `create-capacity-reservation` 예시에서는 `eu-west-1a` 가용 영역에 용량 예약을 생성하여 Linux/Unix 운영 체제를 실행하는 `t2.medium` 인스턴스 3개를 시작할 수 있습니다. 기본적으로 용량 예약은 오픈 인스턴스 매칭 기준으로 생성되며 임시 스토리지는 지원되지 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"
    }
}
```

예시 2: 지정된 날짜/시간에 자동으로 종료되는 용량 예약 생성

다음 `create-capacity-reservation` 예시에서는 `eu-west-1a` 가용 영역에 용량 예약을 생성하여 Linux/Unix 운영 체제를 실행하는 `m5.large` 인스턴스 3개를 시작할 수 있습니다. 이 용량 예약은 2019/8/31 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"
    }
}
```

예시 3: 대상 인스턴스 시작만 허용하는 용량 예약 생성

다음 `create-capacity-reservation` 예시에서는 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 Amazon EC2 사용 설명서의 용량 예약 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateCapacityReservation 섹션을 참조하세요.

`create-carrier-gateway`

다음 코드 예시는 `create-carrier-gateway`의 사용 방법을 보여 줍니다.

AWS CLI

통신 사업자 게이트웨이 생성

다음 `create-carrier-gateway` 예시에서는 지정된 VPC에 대한 통신 사업자 게이트웨이를 생성합니다.

```
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 통신 사업자 게이트웨이를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateCarrierGateway 섹션을 참조하세요.

create-client-vpn-endpoint

다음 코드 예시는 create-client-vpn-endpoint의 사용 방법을 보여 줍니다.

AWS CLI

Client VPN 엔드포인트 생성

다음 create-client-vpn-endpoint 예시에서는 상호 인증을 사용하는 클라이언트 VPN 엔드포인트를 생성하고 클라이언트 CIDR 블록의 값을 지정합니다.

```
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 AWS Client VPN 관리자 안내서의 Client VPN 엔드포인트를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateClientVpnEndpoint 섹션을 참조하세요.

create-client-vpn-route

다음 코드 예시는 create-client-vpn-route의 사용 방법을 보여 줍니다.

AWS CLI

Client VPN 엔드포인트 경로 생성

다음 create-client-vpn-route 예시에서는 클라이언트 VPN 엔드포인트의 지정된 서브넷에 대한 인터넷(0.0.0.0/0)으로의 경로를 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 Client VPN 관리자 안내서의 라우팅을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateClientVpnRoute 섹션을 참조하세요.

create-coip-cidr

다음 코드 예시는 create-coip-cidr의 사용 방법을 보여 줍니다.

AWS CLI

다양한 고객 소유 IP(CoIP) 주소 생성

다음 create-coip-cidr 예시에서는 지정된 CoIP 풀에서 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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateCoipCidr 섹션을 참조하세요.

create-coip-pool

다음 코드 예시는 `create-coip-pool`의 사용 방법을 보여 줍니다.

AWS CLI

고객 소유 IP(CoIP) 주소 풀 생성

다음 `create-coip-pool` 예시에서는 지정된 로컬 게이트웨이 라우팅 테이블에 CoIP 주소에 대한 CoIP 풀을 만듭니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateCoipPool 섹션을 참조하세요.

create-customer-gateway

다음 코드 예시는 `create-customer-gateway`의 사용 방법을 보여 줍니다.

AWS CLI

고객 게이트웨이 생성

이 예시에서는 외부 인터페이스에 대해 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateCustomerGateway 섹션을 참조하세요.

create-default-subnet

다음 코드 예시는 `create-default-subnet`의 사용 방법을 보여 줍니다.

AWS CLI

기본 서브넷 생성

이 예시에서는 `us-east-2a` 가용 영역에 기본 서브넷을 생성합니다.

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
   }
 }
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateDefaultSubnet 섹션을 참조하세요.

create-default-vpc

다음 코드 예시는 `create-default-vpc`의 사용 방법을 보여 줍니다.

AWS CLI

기본 VPC 생성

이 예시에서는 기본 VPC를 생성합니다.

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
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateDefaultVpc 섹션을 참조하세요.

`create-dhcp-options`

다음 코드 예시는 `create-dhcp-options`의 사용 방법을 보여 줍니다.

AWS CLI

DHCP 옵션 세트 생성

다음 `create-dhcp-options` 예시에서는 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateDhcpOptions 섹션을 참조하세요.

`create-egress-only-internet-gateway`

다음 코드 예시는 `create-egress-only-internet-gateway`의 사용 방법을 보여 줍니다.

AWS CLI

외부 전용 인터넷 게이트웨이 생성

이 예시에서는 지정된 VPC에 대한 외부 전용 인터넷 게이트웨이를 만듭니다.

명령:

```
aws ec2 create-egress-only-internet-gateway --vpc-id vpc-0c62a468
```

출력:

```
{
```

```
    "EgressOnlyInternetGateway": {
        "EgressOnlyInternetGatewayId": "eigw-015e0e244e24dfe8a",
        "Attachments": [
            {
                "State": "attached",
                "VpcId": "vpc-0c62a468"
            }
        ]
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateEgressOnlyInternetGateway 섹션을 참조하세요.

`create-fleet`

다음 코드 예시는 `create-fleet`의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 스팟 인스턴스를 기본 구매 모델로 시작하는 EC2 플릿 생성

다음 `create-fleet` 예시에서는 플릿을 시작하는 데 필요한 최소 파라미터인 시작 템플릿, 목표 용량, 기본 구매 모델을 사용하여 EC2 플릿을 생성합니다. 시작 템플릿은 시작 템플릿 ID와 버전 번호로 식별됩니다. 플릿의 목표 용량은 인스턴스 2개이고 기본 구입 모델은 `spot`이므로 플릿이 스팟 인스턴스 2개를 시작합니다.

EC2 집합을 생성할 때 JSON 파일을 사용하여 시작할 인스턴스에 대한 정보를 지정하세요.

```
aws ec2 create-fleet \
    --cli-input-json file://file_name.json
```

file_name.json의 콘텐츠:

```
{
    "LaunchTemplateConfigs": [
    {
        "LaunchTemplateSpecification": {
        "LaunchTemplateId": "lt-0e8c754449b27161c",
        "Version": "1"
        }
    }
```

```
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 2,
        "DefaultTargetCapacityType": "spot"
    }
}
```

출력:

```
{
    "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE"
}
```

예시 2: 온디맨드 인스턴스를 기본 구매 모델로 시작하는 EC2 플릿 생성

다음 `create-fleet` 예시에서는 플릿을 시작하는 데 필요한 최소 파라미터인 시작 템플릿, 목표 용량, 기본 구매 모델을 사용하여 EC2 플릿을 생성합니다. 시작 템플릿은 시작 템플릿 ID와 버전 번호로 식별됩니다. 플릿의 목표 용량은 인스턴스 2개이고 기본 구입 모델 `on-demand`이므로 플릿이 온디맨드 인스턴스 2개를 시작합니다.

EC2 집합을 생성할 때 JSON 파일을 사용하여 시작할 인스턴스에 대한 정보를 지정하세요.

```
aws ec2 create-fleet \
    --cli-input-json file://file_name.json
```

file_name.json의 콘텐츠:

```
{
    "LaunchTemplateConfigs": [
    {
        "LaunchTemplateSpecification": {
        "LaunchTemplateId": "lt-0e8c754449b27161c",
        "Version": "1"
        }
    }
    ],
    "TargetCapacitySpecification": {
    "TotalTargetCapacity": 2,
    "DefaultTargetCapacityType": "on-demand"
    }
}
```

출력:

```
{
    "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE"
}
```

예시 3: 온디맨드 인스턴스를 기본 용량으로 시작하는 EC2 플릿 생성

다음 `create-fleet` 예시에서는 플릿의 총 목표 용량은 인스턴스 2개, 목표 용량은 온디맨드 인스턴스 1개로 지정하는 EC2 플릿을 생성합니다. 기본 구매 모델은 `spot`입니다. 지정한 대로 플릿은 온디맨드 인스턴스 1개를 시작하지만 총 목표 용량을 충족하려면 인스턴스를 하나 더 시작해야 합니다. 차이에 대한 구매 모델이 `TotalTargetCapacity` – `OnDemandTargetCapacity` = `DefaultTargetCapacityType`으로 계산되므로 플릿에서 스팟 인스턴스 1개를 시작합니다.

EC2 집합을 생성할 때 JSON 파일을 사용하여 시작할 인스턴스에 대한 정보를 지정하세요.

```
aws ec2 create-fleet \
    --cli-input-json file://file_name.json
```

file_name.json의 콘텐츠:

```
{
    "LaunchTemplateConfigs": [
    {
        "LaunchTemplateSpecification": {
        "LaunchTemplateId": "lt-0e8c754449b27161c",
        "Version": "1"
        }
    }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 2,
        "OnDemandTargetCapacity":1,
        "DefaultTargetCapacityType": "spot"
    }
}
```

출력:

```
{
```

```
    "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE"
}
```

예시 4: 최저 가격 할당 전략을 사용하여 스팟 인스턴스를 시작하는 EC2 플릿 생성

스팟 인스턴스의 할당 전략이 지정되어 있지 않으면 기본 할당 전략인 `lowest-price`가 사용됩니다. 다음 `create-fleet` 예시에서는 `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 `spot`입니다. EC2 집합는 최저 가격이 지정된 시작 사양의 인스턴스 유형을 사용하여 스팟 인스턴스 2개를 시작합니다.

EC2 집합을 생성할 때 JSON 파일을 사용하여 시작할 인스턴스에 대한 정보를 지정하세요.

```
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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateFleet 섹션을 참조하세요.

create-flow-logs

다음 코드 예시는 `create-flow-logs`의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 흐름 로그 생성

다음 `create-flow-logs` 예시에서는 지정된 네트워크 인터페이스에 대해 거부된 모든 트래픽을 캡처하는 흐름 로그를 생성합니다. 흐름 로그는 지정된 IAM 역할의 권한을 사용하여 CloudWatch Logs의 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.

예시 2: 사용자 지정 형식으로 흐름 로그 생성

다음 `create-flow-logs` 예시에서는 지정된 VPC의 모든 트래픽을 캡처하고 Amazon S3 버킷에 흐름 로그를 전송하는 흐름 로그를 생성합니다. `--log-format` 파라미터는 흐름 로그 레코드의 사용자 지정 형식을 지정합니다. Windows에서 이 명령을 실행하려면 작은따옴표(')를 큰따옴표(")로 변경하세요.

```
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.

예시 3: 최대 집계 간격이 1분인 흐름 로그 생성

다음 `create-flow-logs` 예시에서는 지정된 VPC의 모든 트래픽을 캡처하고 Amazon S3 버킷에 흐름 로그를 전송하는 흐름 로그를 생성합니다. `--max-aggregation-interval` 파라미터는 최대 60초(1분)의 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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateFlowLogs 섹션을 참조하세요.

`create-fpga-image`

다음 코드 예시는 `create-fpga-image`의 사용 방법을 보여 줍니다.

AWS CLI

Amazon FPGA 이미지 생성

이 예시에서는 지정된 버킷의 지정된 tarball에서 AFI를 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [CreateFpgaImage](https://docs.aws.amazon.com/goto/aws-cli/ec2-2016-11-15/CreateFpgaImage) 섹션을 참조하세요.

`create-image`

다음 코드 예시는 `create-image`의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: Amazon EBS 지원 인스턴스에서 AMI 생성

다음 `create-image` 예시에서는 지정된 인스턴스에서 AMI를 생성합니다.

```
aws ec2 create-image \
    --instance-id i-1234567890abcdef0 \
    --name "My server" \
    --description "An AMI for my server"
```

출력:

```
{
    "ImageId": "ami-abcdef01234567890"
}
```

AMI에 대한 블록 디바이스 매핑을 지정하는 방법에 대한 자세한 내용은 Amazon EC2 사용 설명서의 AMI에 대한 블록 디바이스 매핑 지정을 참조하세요.

예시 2: 재부팅 없이 Amazon EBS 지원 인스턴스에서 AMI 생성

다음 `create-image` 예시에서는 이미지가 생성되기 전에 인스턴스가 재부팅되지 않도록 AMI를 생성하고 --no-reboot 파라미터를 설정합니다.

```
aws ec2 create-image \
    --instance-id i-1234567890abcdef0 \
    --name "My server" \
    --no-reboot
```

출력:

```
{
    "ImageId": "ami-abcdef01234567890"
}
```

AMI에 대한 블록 디바이스 매핑을 지정하는 방법에 대한 자세한 내용은 Amazon EC2 사용 설명서의 AMI에 대한 블록 디바이스 매핑 지정을 참조하세요.

예시 3: 생성 시 AMI 및 스냅샷에 태그 지정

다음 `create-image` 예시에서는 AMI를 생성하고, AMI와 스냅샷에 동일한 태그로 `cost-center=cc123` 태그를 지정합니다.

```
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 리소스에 태그를 지정하는 방법에 대한 자세한 내용은 Amazon EC2 사용 설명서의 리소스 생성에 태그 추가를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateImage 섹션을 참조하세요.

create-instance-connect-endpoint

다음 코드 예시는 create-instance-connect-endpoint의 사용 방법을 보여 줍니다.

AWS CLI

EC2 Instance Connect 엔드포인트 생성

다음 create-instance-connect-endpoint 예시에서는 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 생성](#)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [CreateInstanceConnectEndpoint](#) 섹션을 참조하세요.

`create-instance-event-window`

다음 코드 예시는 `create-instance-event-window`의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 시간 범위로 이벤트 기간 생성

다음 `create-instance-event-window` 예시에서는 시간 범위가 있는 이벤트 기간을 만듭니다. `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 사항 섹션을 참조하세요.

예시 2: cron 표현식으로 이벤트 기간 생성

다음 `create-instance-event-window` 예시에서는 cron 표현식을 사용하여 이벤트 기간을 만듭니다. `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 사항 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateInstanceEventWindow 섹션을 참조하세요.

create-instance-export-task

다음 코드 예시는 create-instance-export-task의 사용 방법을 보여 줍니다.

AWS CLI

인스턴스 내보내기

이 예시에서는 인스턴스 i-1234567890abcdef0을 Amazon S3 버킷 myexportbucket으로 내보내는 태스크를 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateInstanceExportTask 섹션 섹션을 참조하세요.

create-internet-gateway

다음 코드 예시는 create-internet-gateway의 사용 방법을 보여 줍니다.

AWS CLI

인터넷 게이트웨이 생성

다음 create-internet-gateway 예시에서는 태그가 Name=my-igw인 인터넷 게이트웨이를 생성합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateInternetGateway 섹션 섹션을 참조하세요.

create-ipam-pool

다음 코드 예시는 create-ipam-pool의 사용 방법을 보여 줍니다.

AWS CLI

IPAM 풀 생성

다음 `create-ipam-pool` 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateIpamPool 섹션을 참조하세요.

create-ipam-resource-discovery

다음 코드 예시는 `create-ipam-resource-discovery`의 사용 방법을 보여 줍니다.

AWS CLI

리소스 검색 생성

이 예시에서는 다른 조직의 관리자가 내 조직에 있는 리소스의 IP 주소를 관리하고 모니터링할 수 있도록 리소스 검색을 만들어 다른 AWS Organization의 IPAM 관리자와 공유하려는 위임받은 IPAM 관리자의 경우를 가정합니다.

중요

이 예시에서는 `--region` 및 `--operating-regions` 옵션을 모두 포함하는데, 이는 선택 사항이지만 리소스 검색을 IPAM과 성공적으로 통합하려면 특정 방식으로 구성해야 하기 때문입니다. * `--operating-regions`는 IPAM이 검색하려는 리소스가 있는 리전과 일치해야 합니다. 규정 준

수 등의 이유로 IPAM이 IP 주소를 관리하지 않으려는 리전이 있는 경우 해당 리전을 포함하지 마세요. * --region은 연결하려는 IPAM의 홈 리전과 일치해야 합니다. 리소스 검색은 IPAM이 생성된 리전과 동일한 리전에서 생성해야 합니다. 예를 들어, 연결하려는 IPAM이 us-east-1에서 생성된 경우 요청에 --region us-east-1이 포함됩니다. --region 및 --operating-regions 옵션은 지정하지 않으면 기본적으로 명령을 실행하는 리전으로 설정됩니다.

이 예시에서 통합하는 IPAM의 운영 리전에는 us-west-1, us-west-2, ap-south-1가 포함됩니다. 리소스 검색을 생성할 때 IPAM이 us-west-1과 us-west-2에서 리소스 IP 주소를 검색하지만 ap-south-1은 검색하지 않기를 원합니다. 따라서 요청에 --operating-regions RegionName='us-west-1' RegionName='us-west-2'만 포함시키고 있습니다.

다음 create-ipam-resource-discovery 예시에서는 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 만든 후에는 다른 IPAM 위임된 관리자와 공유할 수 있으며, 이 경우 create-resource-share를 사용하여 공유할 수 있습니다. 자세한 내용은 Amazon VPC IPAM 사용 설명서의 IPAM을 조직 외부 계정과 통합 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateIpamResourceDiscovery 섹션을 참조하세요.

`create-ipam-scope`

다음 코드 예시는 `create-ipam-scope`의 사용 방법을 보여 줍니다.

AWS CLI

IPAM 범위 생성

다음 `create-ipam-scope` 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateIpamScope 섹션을 참조하세요.

`create-ipam`

다음 코드 예시는 `create-ipam`의 사용 방법을 보여 줍니다.

AWS CLI

IPAM 생성

다음 `create-ipam` 예시에서는 IPAM을 생성합니다.

(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 IPM 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateIpam 섹션을 참조하세요.

create-key-pair

다음 코드 예시는 `create-key-pair`의 사용 방법을 보여 줍니다.

AWS CLI

키 페어 생성

이 예제에서는 이름이 `MyKeyPair`인 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.

- API 세부 정보는 AWS CLI 명령 참조의 [CreateKeyPair](https://docs.aws.amazon.com/goto/aws-cli/ec2-2016-11-15/CreateKeyPair)를 참조하세요.

`create-launch-template-version`

다음 코드 예시는 `create-launch-template-version`의 사용 방법을 보여 줍니다.

AWS CLI

시작 템플릿 버전 생성

이 예시에서는 실행 템플릿 버전 1을 기반으로 새 실행 템플릿 버전을 만들고 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateLaunchTemplateVersion 섹션을 참조하세요.

create-launch-template

다음 코드 예시는 create-launch-template의 사용 방법을 보여 줍니다.

AWS CLI

예 1: 시작 템플릿을 생성하는 방법

다음 create-launch-template 예제에서는 인스턴스를 시작하고 인스턴스에 퍼블릭 IP 주소 및 IPv6 주소를 할당하며 인스턴스에 대한 태그를 생성할 서브넷을 지정하는 시작 템플릿을 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. JSON 형식 파라미터에서 따옴표 사용에 대한 자세한 내용은 AWS Command Line Interface 사용 설명서의 문자열에 따옴표 사용을 참조하세요.

예제 2: Amazon EC2 Auto Scaling에 대한 시작 템플릿을 생성하는 방법

다음 `create-launch-template` 예제에서는 인스턴스를 시작할 때 추가 EBS 볼륨을 지정하도록 여러 태그 및 블록 디바이스 매핑을 사용하는 시작 템플릿을 생성합니다. Auto Scaling이 인스턴스를 시작하는 VPC의 보안 그룹에 해당하는 `Groups`에 대한 값을 지정합니다. Auto Scaling의 속성으로 VPC 및 서브넷을 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 EC2 Auto Scaling 사용 설명서의 Auto Scaling 그룹에 대한 시작 템플릿 생성을 참조하세요. JSON 형식 파라미터에서 따옴표 사용에 대한 자세한 내용은 AWS Command Line Interface 사용 설명서의 문자열에 따옴표 사용을 참조하세요.

예제 3: EBS 볼륨의 암호화를 지정하는 시작 템플릿을 생성하는 방법

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 CMK를 지정하는 경우 기본적으로 암호화가 활성화되어 있더라도 `"Encrypted"` 옵션도 지정해야 합니다.

```
aws ec2 create-launch-template \
  --launch-template-name TemplateForEncryption \
  --launch-template-data file://config.json
```

`config.json`의 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 암호화 기본 제공을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조에서 CreateLaunchTemplate을 참조하세요.

`create-local-gateway-route-table-virtual-interface-group-association`

다음 코드 예시는 `create-local-gateway-route-table-virtual-interface-group-association`의 사용 방법을 보여 줍니다.

AWS CLI

로컬 게이트웨이 라우팅 테이블을 가상 인터페이스(VIF) 그룹과 연결

다음 `create-local-gateway-route-table-virtual-interface-group-association` 예시에서는 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 AWS Outposts 사용 설명서의 VF 그룹 연결을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateLocalGatewayRouteTableVirtualInterfaceGroupAssociation 섹션을 참조하세요.

create-local-gateway-route-table-vpc-association

다음 코드 예시는 create-local-gateway-route-table-vpc-association의 사용 방법을 보여 줍니다.

AWS CLI

라우팅 테이블과 VPC 연결

다음 create-local-gateway-route-table-vpc-association 예시에서는 지정한 VPC를 지정한 로컬 게이트웨이 라우팅 테이블과 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateLocalGatewayRouteTableVpcAssociation 섹션을 참조하세요.

create-local-gateway-route-table

다음 코드 예시는 `create-local-gateway-route-table`의 사용 방법을 보여 줍니다.

AWS CLI

로컬 게이트웨이 라우팅 테이블 생성

다음 `create-local-gateway-route-table` 예시에서는 직접 VPC 라우팅 모드로 로컬 게이트웨이 라우팅 테이블을 만듭니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateLocalGatewayRouteTable 섹션을 참조하세요.

`create-local-gateway-route`

다음 코드 예시는 `create-local-gateway-route`의 사용 방법을 보여 줍니다.

AWS CLI

로컬 게이트웨이 라우팅 테이블에 정적 라우팅 생성

다음 `create-local-gateway-route` 예시에서는 지정된 로컬 게이트웨이 라우팅 테이블에 지정된 경로를 만듭니다.

```
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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateLocalGatewayRoute 섹션을 참조하세요.

`create-managed-prefix-list`

다음 코드 예시는 `create-managed-prefix-list`의 사용 방법을 보여 줍니다.

AWS CLI

접두사 목록 생성

다음 `create-managed-prefix-list` 예시에서는 최대 10개의 항목이 포함된 IPv4 접두사 목록을 생성하고 접두사 목록에 두 개의 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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateManagedPrefixList 섹션을 참조하세요.

`create-nat-gateway`

다음 코드 예시는 `create-nat-gateway`의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 퍼블릭 NAT 게이트웨이 생성

다음 `create-nat-gateway` 예시에서는 지정된 서브넷에 퍼블릭 NAT 게이트웨이를 생성하고 탄력적 IP 주소를 지정된 할당 ID와 연결합니다. 퍼블릭 NAT 게이트웨이를 만들 때는 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 정보는 Amazon VPC 사용 설명서의 NAT 게이트웨이 섹션을 참조하세요.

예시 2: 프라이빗 NAT 게이트웨이 생성

다음 `create-nat-gateway` 예시에서는 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 정보는 Amazon VPC 사용 설명서의 NAT 게이트웨이 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateNatGateway 섹션을 참조하세요.

create-network-acl-entry

다음 코드 예시는 create-network-acl-entry의 사용 방법을 보여 줍니다.

AWS CLI

네트워크 ACL 항목 생성

이 예시에서는 지정된 네트워크 ACL에 대한 항목을 만듭니다. 이 규칙은 UDP 포트 53(DNS)의 모든 IPv4 주소(0.0.0.0/0)에서 연결된 모든 서브넷으로 유입되는 트래픽을 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 예시에서는 TCP 포트 80(HTTP)의 모든 IPv6 주소(::/0)에서 들어오는 트래픽을 허용하는 지정된 네트워크 ACL에 대한 규칙을 만듭니다.

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
```

- API 세부 정보는 AWS CLI 명령 참조의 [CreateNetworkAclEntry](#) 섹션을 참조하세요.

create-network-acl

다음 코드 예시는 `create-network-acl`의 사용 방법을 보여 줍니다.

AWS CLI

네트워크 ACL 생성

이 예시에서는 지정된 VPC에 대한 네트워크 ACL을 만듭니다.

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
    }
```

```
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateNetworkAcl 섹션을 참조하세요.

create-network-insights-access-scope

다음 코드 예시는 create-network-insights-access-scope의 사용 방법을 보여 줍니다.

AWS CLI

네트워크 액세스 범위 생성

다음 create-network-insights-access-scope 예시에서는 네트워크 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:


```
{
    "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 설명서의 AWS CLI를 사용하여 Network Access Analyzer 시작하기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateNetworkInsightsAccessScope 섹션을 참조하세요.

create-network-insights-path

다음 코드 예시는 create-network-insights-path의 사용 방법을 보여 줍니다.

AWS CLI

경로 생성

다음 create-network-insights-path 예시에서는 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 설명서의 AWS CLI를 사용하여 시작하기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateNetworkInsightsPath 섹션을 참조하세요.

create-network-interface-permission

다음 코드 예시는 create-network-interface-permission의 사용 방법을 보여 줍니다.

AWS CLI

네트워크 인터페이스 권한 생성

이 예시에서는 인스턴스에 네트워크 인터페이스 eni-1a2b3c4d를 연결할 수 있는 권한을 123456789012 계정에 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"
  }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateNetworkInterfacePermission 섹션을 참조하세요.

create-network-interface

다음 코드 예시는 create-network-interface의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 네트워크 인터페이스의 IPv4 주소 지정

다음 create-network-interface 예시에서는 지정된 기본 IPv4 주소로 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"
    }
}
```


예시 2: IPv4 주소와 IPv6 주소로 네트워크 인터페이스 생성

다음 `create-network-interface` 예시에서는 Amazon EC2에서 선택한 IPv4 주소와 IPv6 주소로 지정된 서브넷에 대한 네트워크 인터페이스를 생성합니다.

```
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"
    }
}
```

예시 3: 연결 추적 구성 옵션을 사용하여 네트워크 인터페이스 생성

다음 `create-network-interface` 예시에서는 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"
    }
}
```

예시 4: 탄력적 패브릭 어댑터 생성

다음 `create-network-interface` 예시에서는 EFA를 생성합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateNetworkInterface 섹션을 참조하세요.

`create-placement-group`

다음 코드 예시는 `create-placement-group`의 사용 방법을 보여 줍니다.

AWS CLI

배치 그룹 생성

이 예시에서는 지정된 이름으로 배치 그룹을 생성합니다.

명령:

```
aws ec2 create-placement-group --group-name my-cluster --strategy cluster
```

파티션 배치 그룹 생성

이 예시에서는 5개의 파티션이 있는 `HDFS-Group-A` 파티션 배치 그룹을 만듭니다.

명령:

```
aws ec2 create-placement-group --group-name HDFS-Group-A --strategy partition --
partition-count 5
```

- API 세부 정보는 AWS CLI 명령 참조의 CreatePlacementGroup 섹션을 참조하세요.

`create-replace-root-volume-task`

다음 코드 예시는 `create-replace-root-volume-task`의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 루트 볼륨을 초기 시작 상태 복원

다음 `create-replace-root-volume-task` 예시에서는 인스턴스 i-0123456789abcdefa의 루트 볼륨을 초기 실행 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": []
    }
}
```

예시 2: 루트 볼륨을 특정 스냅샷으로 복원

다음 `create-replace-root-volume-task` 예시에서는 인스턴스 i-0123456789abcdefa의 루트 볼륨을 스냅샷 snap-0abcdef1234567890으로 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 EC2 사용 설명서의 루트 볼륨 교체를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateReplaceRootVolumeTask 섹션을 참조하세요.

create-reserved-instances-listing

다음 코드 예시는 create-reserved-instances-listing의 사용 방법을 보여 줍니다.

AWS CLI

예약 인스턴스 Marketplace의 예약 인스턴스 나열

다음 create-reserved-instances-listing 예시에서는 예약 인스턴스 마켓플레이스에서 지정된 예약 인스턴스에 대한 리스팅을 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
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateReservedInstancesListing 섹션을 참조하세요.

create-restore-image-task

다음 코드 예시는 create-restore-image-task의 사용 방법을 보여 줍니다.

AWS CLI

S3 버킷에서 AMI 복원

다음 create-restore-image-task 예시에서는 S3 버킷에서 AMI를 복원합니다. describe-store-image-tasks 출력에서 S3ObjectKey `` and ``Bucket 값을 사용하고, AMI의 객체 키와 AMI가 복사된 S3 버킷의 이름을 지정한 다음, 복원된 AMI의 이름을 지정합니다. 이름은 이 계정의 리전 내 AMI에 대해 고유해야 합니다. 복원된 AMI는 새 AMI ID를 받게 됩니다.

```
aws ec2 create-restore-image-task \
    --object-key ami-1234567890abcdef0.bin \
    --bucket my-ami-bucket \
    --name 'New AMI Name'
```

출력:

```
{
    "ImageId": "ami-0eab20fe36f83e1a8"
}
```

자세한 내용은 Amazon EC2 사용 설명서의 S3를 사용하여 AMI 저장 및 복원을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateRestoreImageTask 섹션을 참조하세요.

create-route-table

다음 코드 예시는 `create-route-table`의 사용 방법을 보여 줍니다.

AWS CLI

라우팅 테이블을 생성하는 방법

이 예제에서는 지정된 VPC에 대한 라우팅 테이블을 생성합니다.

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"
                }
            ]
        }
}
```

- API 세부 정보는 AWS CLI 명령 참조에서 [CreateRouteTable](#)을 참조하세요.

`create-route`

다음 코드 예시는 `create-route`의 사용 방법을 보여 줍니다.

AWS CLI

라우팅 생성

이 예시에서는 지정된 라우팅 테이블에 대한 경로를 생성합니다. 이 경로는 모든 IPv4 트래픽 (`0.0.0.0/0`)을 일치시켜 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 예시에서는 라우팅 테이블 rtb-g8ff4ea2에 경로를 만듭니다. 이 경로는 IPv4 CIDR 블록 10.0.0.0/16에 대한 트래픽을 일치시키고 이를 VPC 피어링 연결인 pcx-111aaa22로 라우팅합니다. 이 라우팅을 사용하면 VPC 피어링 연결에서 트래픽을 피어 VPC로 보낼 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 예시에서는 지정된 라우팅 테이블에 모든 IPv6 트래픽(`::/0`)과 일치하는 경로를 생성하여 지정된 외부 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
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateRoute 섹션을 참조하세요.

create-security-group

다음 코드 예시는 `create-security-group`의 사용 방법을 보여 줍니다.

AWS CLI

EC2-Classic에 대한 보안 그룹을 생성하는 방법

이 예제에서는 이름이 `MySecurityGroup`인 보안 그룹을 생성합니다.

명령:

```
aws ec2 create-security-group --group-name MySecurityGroup --description "My security group"
```

출력:

```
{
    "GroupId": "sg-903004f8"
}
```

EC2-VPC에 대한 보안 그룹을 생성하는 방법

이 예제에서는 지정된 VPC에 대해 이름이 `MySecurityGroup`인 보안 그룹을 생성합니다.

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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateSecurityGroup을 참조하세요.

create-snapshot

다음 코드 예시는 `create-snapshot`의 사용 방법을 보여 줍니다.

AWS CLI

스냅샷 생성

이 예시에서는 볼륨 ID가 `vol-1234567890abcdef0`의 볼륨 스냅샷과 스냅샷을 식별할 수 있는 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 스냅샷 생성

이 예시에서는 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"
 }
```

- API 세부 정보는 AWS CLI 명령 참조의 [CreateSnapshot](#) 섹션을 참조하세요.

`create-snapshots`

다음 코드 예시는 `create-snapshots`의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 다중 볼륨 스냅샷 생성

다음 `create-snapshots` 예시에서는 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"
        },
        ...
    ]
}
```

예시 2: 소스 볼륨의 태그를 사용하여 다중 볼륨 스냅샷 생성

다음 `create-snapshots` 예시에서는 지정된 인스턴스에 연결된 모든 볼륨의 스냅샷을 생성하고 각 볼륨의 태그를 해당 스냅샷에 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"
        }
        ...
    ]
}
```

예시 3: 루트 볼륨을 포함하지 않는 다중 볼륨 스냅샷 생성

다음 `create-snapshots` 예시에서는 루트 볼륨을 제외한 지정된 인스턴스에 연결된 모든 볼륨의 스냅샷을 생성합니다.

```
aws ec2 create-snapshots \
    --instance-specification InstanceId=i-1234567890abcdef0,ExcludeBootVolume=true
```

샘플 출력은 예 1을 참조하세요.

예시 4: 다중 볼륨 스냅샷 생성 및 태그 추가

다음 `create-snapshots` 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateSnapshots 섹션을 참조하세요.

create-spot-datafeed-subscription

다음 코드 예시는 `create-spot-datafeed-subscription`의 사용 방법을 보여 줍니다.

AWS CLI

스팟 인스턴스 데이터 피드 생성

다음 `create-spot-datafeed-subscription` 예시에서는 스팟 인스턴스 데이터 피드를 생성합니다.

```
aws ec2 create-spot-datafeed-subscription \
    --bucket amzn-s3-demo-bucket \
    --prefix spot-data-feed
```

출력:

```
{
    "SpotDatafeedSubscription": {
        "Bucket": "amzn-s3-demo-bucket",
        "OwnerId": "123456789012",
        "Prefix": "spot-data-feed",
        "State": "Active"
    }
}
```

데이터 피드는 지정한 Amazon S3 버킷에 저장됩니다. 이 데이터 피드의 파일 이름은 다음과 같은 형식을 따릅니다.

```
amzn-s3-demo-bucket.s3.amazonaws.com/spot-data-feed/123456789012.YYYY-MM-DD-
HH.n.abcd1234.gz
```

자세한 내용은 Amazon EC2 사용 설명서의 스팟 인스턴스 데이터 피드를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [CreateSpotDatafeedSubscription](#) 섹션을 참조하세요.

`create-store-image-task`

다음 코드 예시는 `create-store-image-task`의 사용 방법을 보여 줍니다.

AWS CLI

S3 버킷에 AMI 저장

다음 `create-store-image-task` 예시에서는 AMI를 S3 버킷에 저장합니다. AMI의 ID와 AMI를 저장할 S3 버킷의 이름을 지정합니다.

```
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 사용하여 AMI 저장 및 복원](#)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [CreateStoreImageTask](#) 섹션을 참조하세요.

`create-subnet-cidr-reservation`

다음 코드 예시는 `create-subnet-cidr-reservation`의 사용 방법을 보여 줍니다.

AWS CLI

서브넷 CIDR 예약 생성

다음 `create-subnet-cidr-reservation` 예시에서는 지정된 서브넷과 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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateSubnetCidrReservation 섹션을 참조하세요.

`create-subnet`

다음 코드 예시는 `create-subnet`의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: IPv4 CIDR 블록만 사용하여 서브넷을 생성하는 방법

다음 `create-subnet` 예제에서는 지정된 IPv4 CIDR 블록을 사용하여 지정된 VPC에서 서브넷을 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 서브넷을 생성하는 방법

다음 `create-subnet` 예제에서는 지정된 IPv4 및 IPv6 CIDR 블록을 사용하여 지정된 VPC에서 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 서브넷을 생성하는 방법

다음 create-subnet 예제에서는 지정된 IPv6 CIDR 블록을 사용하여 지정된 VPC에서 서브넷을 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 VPC 및 서브넷을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조에서 CreateSubnet을 참조하세요.

create-tags

다음 코드 예시는 create-tags의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 리소스에 태그 추가

다음 create-tags 예제에서는 지정된 이미지에 Stack=production 태그를 추가하거나 태그 키가 Stack인 AMI의 기존 태그를 덮어씁니다.

```
aws ec2 create-tags \
    --resources ami-1234567890abcdef0 \
    --tags Key=Stack,Value=production
```

이 명령은 출력을 생성하지 않습니다.

예시 2: 여러 리소스에 태그 추가

다음 `create-tags` 예제에서는 AMI와 인스턴스에 대해 두 개의 태그를 추가하거나 덮어씁니다. 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
```

이 명령은 출력을 생성하지 않습니다.

예시 3: 특수 문자가 포함된 태그 추가

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 PowerShell을 사용하는 경우 다음과 같이 특수 문자가 있는 값을 큰따옴표(")로 묶고 각 큰따옴표 문자 앞에 백슬래시(\)를 붙인 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 Amazon EC2 사용 설명서의 Amazon EC2 리소스에 태그 지정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateTags를 참조하세요.

create-traffic-mirror-filter-rule

다음 코드 예시는 create-traffic-mirror-filter-rule의 사용 방법을 보여 줍니다.

AWS CLI

수신 TCP 트래픽에 대한 필터 규칙 생성

다음 create-traffic-mirror-filter-rule 예시에서는 모든 수신 TCP 트래픽을 미러링하는 데 사용할 수 있는 규칙을 생성합니다. 이 명령을 실행하기 전에 create-traffic-mirror-filter를 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 안내서의 트래픽 미러 필터 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateTrafficMirrorFilterRule 섹션을 참조하세요.

`create-traffic-mirror-filter`

다음 코드 예시는 `create-traffic-mirror-filter`의 사용 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-rule`을 사용하여 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 안내서의 트래픽 미러 필터 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateTrafficMirrorFilter 섹션을 참조하세요.

`create-traffic-mirror-session`

다음 코드 예시는 `create-traffic-mirror-session`의 사용 방법을 보여 줍니다.

AWS CLI

트래픽 미러 세션을 생성하려면

다음 `create-traffic-mirror-session` 명령은 지정된 소스 및 대상에 대해 25바이트의 패킷에 대한 트래픽 미러 세션을 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 안내서의 트래픽 미러 세션 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateTrafficMirrorSession 섹션을 참조하세요.

`create-traffic-mirror-target`

다음 코드 예시는 `create-traffic-mirror-target`의 사용 방법을 보여 줍니다.

AWS CLI

Network Load Balancer 트래픽 미러 대상을 생성하려면

다음 `create-traffic-mirror-target` 예제에서는 Network Load Balancer 트래픽 미러 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` 예시에서는 네트워크 인터페이스 트래픽 미러 대상을 생성합니다.

```
aws ec2 create-traffic-mirror-target \
    --description 'Network interface target' \
    --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 안내서의 트래픽 미러 대상 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateTrafficMirrorTarget 섹션을 참조하세요.

create-transit-gateway-connect-peer

다음 코드 예시는 `create-transit-gateway-connect-peer`의 사용 방법을 보여 줍니다.

AWS CLI

Transit Gateway Connect 피어 생성

다음 `create-transit-gateway-connect-peer` 예시에서는 연결 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 설명서의 [전송 게이트웨이 Connect 연결 및 Transit Gateway Connect 피어]를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [CreateTransitGatewayConnectPeer] 섹션을 참조하세요.

`create-transit-gateway-connect`

다음 코드 예시는 `create-transit-gateway-connect`의 사용 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 Connect 연결 생성

다음 `create-transit-gateway-connect` 예시에서는 지정된 연결에 대해 'gre' 프로토콜을 사용하여 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 설명서의 전송 게이트웨이 Connect 연결 및 Transit Gateway Connect 피어를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateTransitGatewayConnect 섹션을 참조하세요.

`create-transit-gateway-multicast-domain`

다음 코드 예시는 `create-transit-gateway-multicast-domain`의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: IGMP 멀티캐스트 도메인 생성

다음 `create-transit-gateway-multicast-domain` 예시에서는 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"
    }
}
```

예시 2: 정적 멀티캐스트 도메인 생성

다음 `create-transit-gateway-multicast-domain` 예시에서는 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 설명서의 Managing multicast domains 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateTransitGatewayMulticastDomain 섹션을 참조하세요.

create-transit-gateway-peering-attachment

다음 코드 예시는 `create-transit-gateway-peering-attachment`의 사용 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 피어링 연결 생성

다음 `create-transit-gateway-peering-attachment` 예시에서는 지정된 두 개의 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 설명서의 Transit Gateway Peering Attachments 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateTransitGatewayPeeringAttachment 섹션을 참조하세요.

`create-transit-gateway-policy-table`

다음 코드 예시는 `create-transit-gateway-policy-table`의 사용 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 정책 테이블 생성

다음 `create-transit-gateway-policy-table` 예시에서는 지정된 전송 게이트웨이에 대한 전송 게이트웨이 정책 테이블을 만듭니다.

```
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 전송 게이트웨이 정책 테이블을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateTransitGatewayPolicyTable 섹션을 참조하세요.

`create-transit-gateway-prefix-list-reference`

다음 코드 예시는 `create-transit-gateway-prefix-list-reference`의 사용 방법을 보여 줍니다.

AWS CLI

접두사 목록에 대한 참조 생성

다음 `create-transit-gateway-prefix-list-reference` 예시에서는 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-11111122222222333 \
    --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 설명서의 접두사 목록 참조 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateTransitGatewayPrefixListReference 섹션을 참조하세요.

create-transit-gateway-route-table

다음 코드 예시는 `create-transit-gateway-route-table`의 사용 방법을 보여 줍니다.

AWS CLI

Transit Gateway 라우팅 테이블 생성

다음 `create-transit-gateway-route-table` 예시에서는 지정된 전송 게이트웨이에 대한 라우팅 테이블을 만듭니다.

```
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 설명서의 전송 게이트웨이 라우팅 테이블 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateTransitGatewayRouteTable 섹션을 참조하세요.

`create-transit-gateway-route`

다음 코드 예시는 `create-transit-gateway-route`의 사용 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 라우팅 생성

다음 `create-transit-gateway-route` 예시에서는 지정된 라우팅 테이블에 대해 대상이 지정된 경로를 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 설명서의 전송 게이트웨이 라우팅 테이블을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateTransitGatewayRoute 섹션을 참조하세요.

`create-transit-gateway-vpc-attachment`

다음 코드 예시는 `create-transit-gateway-vpc-attachment`의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 전송 게이트웨이를 VPC에 연결

다음 `create-transit-gateway-vpc-attachment` 예시에서는 지정된 VPC에 전송 게이트웨이 연결을 생성합니다.

```
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 설명서의 [VPC에 대한 전송 게이트웨이 연결 생성](#)을 참조하세요.

예시 2: 전송 게이트웨이를 VPC의 여러 서브넷에 연결

다음 `create-transit-gateway-vpc-attachment` 예시에서는 지정된 VPC 및 서브넷에 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 설명서의 [VPC에 대한 전송 게이트웨이 연결 생성](#)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [CreateTransitGatewayVpcAttachment](#) 섹션을 참조하세요.

`create-transit-gateway`

다음 코드 예시는 `create-transit-gateway`의 사용 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 생성

다음 `create-transit-gateway` 예시에서는 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 설명서의 전송 게이트웨이 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateTransitGateway 섹션을 참조하세요.

create-verified-access-endpoint

다음 코드 예시는 create-verified-access-endpoint의 사용 방법을 보여 줍니다.

AWS CLI

Verified·Access 엔드포인트 생성

다음 create-verified-access-endpoint 지정된 Verified Access 그룹에 대한 Verified Access 엔드포인트를 생성합니다. 지정된 네트워크 인터페이스와 보안 그룹은 동일한 VPC에 속해야 합니다.

```
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 AWS Verified Access 사용 설명서의 Verified Access endpoints 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateVerifiedAccessEndpoint 섹션을 참조하세요.

create-verified-access-group

다음 코드 예시는 `create-verified-access-group`의 사용 방법을 보여 줍니다.

AWS CLI

Verified·Access 그룹 생성

다음 `create-verified-access-group` 예시에서는 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 AWS Verified Access 사용 설명서의 Verified Access groups 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateVerifiedAccessGroup 섹션을 참조하세요.

`create-verified-access-instance`

다음 코드 예시는 `create-verified-access-instance`의 사용 방법을 보여 줍니다.

AWS CLI

Verified·Access 인스턴스 생성

다음 `create-verified-access-instance` 예시에서는 Name 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 AWS Verified Access 사용 설명서의 Verified Access instances 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateVerifiedAccessInstance 섹션을 참조하세요.

`create-verified-access-trust-provider`

다음 코드 예시는 `create-verified-access-trust-provider`의 사용 방법을 보여 줍니다.

AWS CLI

Verified·Access 신뢰 공급자 생성

다음 `create-verified-access-trust-provider` 예시에서는 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 AWS Verified Access 사용 설명서의 Trust providers for Verified Access 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateVerifiedAccessTrustProvider 섹션을 참조하세요.

create-volume

다음 코드 예시는 create-volume의 사용 방법을 보여 줍니다.

AWS CLI

비어 있는 범용 SSD(gp2) 볼륨 생성

다음 create-volume 지정된 가용 영역에 80GiB 범용 SSD(gp2) 볼륨을 생성합니다. 현재 리전은 반드시 us-east-1이거나 --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 gp2입니다.

```
aws ec2 create-volume \
    --size 80 \
    --availability-zone us-east-1a
```

예시 2: 스냅샷에서 프로비저닝된 IOPS SSD(io1) 볼륨 생성

다음 `create-volume` 예시에서는 지정된 스냅샷을 사용하여 지정된 가용 영역에 1000개의 프로비저닝된 IOPS를 가진 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
}
```

예시 3: 암호화된 볼륨 생성

다음 create-volume 예시에서는 EBS 암호화를 위한 기본 CMK를 사용하여 암호화된 볼륨을 생성합니다. 기본적으로 암호화가 비활성화되어 있는 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 예시에서는 --encrypted 파라미터가 없어도 암호화된 볼륨을 생성합니다.

```
aws ec2 create-volume \
    --size 80 \
    --availability-zone us-east-1a
```

--kms-key-id 파라미터를 사용하여 고객 관리형 CMK를 지정하는 경우 기본적으로 암호화가 활성화되어 있더라도 --encrypted 파라미터도 지정해야 합니다.

```
aws ec2 create-volume \
    --volume-type gp2 \
    --size 80 \
    --encrypted \
    --kms-key-id 0ea3fef3-80a7-4778-9d8c-1c0c6EXAMPLE \
    --availability-zone us-east-1a
```

예시 4: 태그를 사용하여 볼륨 생성

다음 `create-volume` 예시에서는 볼륨을 만들고 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}]'
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateVolume 섹션을 참조하세요.

`create-vpc-endpoint-connection-notification`

다음 코드 예시는 `create-vpc-endpoint-connection-notification`의 사용 방법을 보여 줍니다.

AWS CLI

엔드포인트 연결 알림 생성

이 예시에서는 특정 엔드포인트 서비스에 대한 알림을 만들어 인터페이스 엔드포인트가 서비스에 연결되었을 때와 엔드포인트가 서비스에 대해 수락되었을 때를 알려줍니다.

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"
    }
 }
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateVpcEndpointConnectionNotification 섹션을 참조하세요.

`create-vpc-endpoint-service-configuration`

다음 코드 예시는 `create-vpc-endpoint-service-configuration`의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 인터페이스 엔드포인트에 대한 엔드포인트 서비스 구성 생성

다음 `create-vpc-endpoint-service-configuration` 예시에서는 Network Load Balancer `nlb-vpce`를 사용하여 VPC 엔드포인트 서비스 구성을 생성합니다. 이 예시에서는 인터페이스 엔드포인트를 통해 서비스에 연결하라는 요청을 수락해야 한다고 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"
        ]
    }
}
```

자세한 내용은 AWS PrivateLink 사용 설명서의 엔드포인트 생성을 참조하세요.

예시 2: Gateway Load Balancer 엔드포인트에 대한 엔드포인트 서비스 구성 생성

다음 `create-vpc-endpoint-service-configuration` 예시에서는 Gateway Load Balancer `GWLBService`를 사용하여 VPC 엔드포인트 서비스 구성을 생성합니다. Gateway Load Balancer 엔드포인트를 통한 서비스 연결 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 AWS PrivateLink 사용 설명서의 Gateway Load Balancer 엔드포인트 서비스 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateVpcEndpointServiceConfiguration 섹션을 참조하세요.

`create-vpc-endpoint`

다음 코드 예시는 `create-vpc-endpoint`의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 게이트웨이 엔드포인트 생성

다음 `create-vpc-endpoint` 예시에서는 VPC `vpc-1a2b3c4d`와 `us-east-1` 리전의 Amazon S3 사이에 게이트웨이 VPC 엔드포인트를 생성하고 라우팅 테이블 `rtb-11aa22bb`를 엔드포인트와 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 설명서의 게이트웨이 엔드포인트 생성을 참조하세요.

예시 2: 인터페이스 엔드포인트 생성

다음 `create-vpc-endpoint` 예시에서는 `us-east-1` 리전의 VPC `vpc-1a2b3c4d`와 Amazon S3 사이에 인터페이스 VPC 엔드포인트를 생성합니다. 이 명령은 `subnet-1a2b3c4d` 서브넷에 엔드포인트를 만들고 `sg-1a2b3c4d` 보안 그룹에 연결한 다음 키가 'Service'이고 값이 'S3'인 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 AWS PrivateLink 사용 설명서의 인터페이스 VPC 엔드포인트 생성을 참조하세요.

예시 3: Gateway Load Balancer 엔드포인트 생성

다음 `create-vpc-endpoint` 예시에서는 VPC `vpc-111122223333aabbc`와 Gateway Load Balancer를 사용하여 구성된 서비스 사이에 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 AWS PrivateLink 사용 설명서의 Gateway Load Balancer 엔드포인트를 참조하세요.

예제 4: 리소스 엔드포인트 만들기

다음 `create-vpc-endpoint` 예제에서는 리소스 엔드포인트를 만듭니다.

```
aws ec2 create-vpc-endpoint \
    --vpc-endpoint-type Resource \
    --vpc-id vpc-111122223333aabbc \
    --subnet-ids subnet-0011aabbcc2233445 \
    --resource-configuration-arn arn:aws:vpc-lattice-us-
east-1:123456789012:resourceconfiguration/rcfg-0123abcde98765432
```

출력:

```
{
    "VpcEndpoint": {
        "VpcEndpointId": "vpce-00939a7ed9EXAMPLE",
        "VpcEndpointType": "Resource",
        "VpcId": "vpc-111122223333aabbc",
        "State": "Pending",
        "SubnetIds": [
            "subnet-0011aabbcc2233445"
        ],
        "Groups": [
            {
```

```
                "GroupId": "sg-03e2f15fbfc09b000",
                "GroupName": "default"
            }
        ],
        "IpAddressType": "IPV4",
        "PrivateDnsEnabled": false,
        "CreationTimestamp": "2025-02-06T23:38:49.525000+00:00",
        "Tags": [],
        "OwnerId": "123456789012",
        "ResourceConfigurationArn": "arn:aws:vpc-lattice:us-
east-1:123456789012:resourceconfiguration/rcfg-0123abcde98765432"
    }
}
```

자세한 내용은 AWS PrivateLink 사용 설명서의 리소스 엔드포인트를 참조하시기 바랍니다.

예제 5: 서비스 네트워크 엔드포인트 만들기

다음 `create-vpc-endpoint` 예제에서는 서비스 네트워크 엔드포인트를 만듭니다.

```
aws ec2 create-vpc-endpoint \
    --vpc-endpoint-type ServiceNetwork \
    --vpc-id vpc-111122223333aabbc \
    --subnet-ids subnet-0011aabbcc2233445 \
    --service-network-arn arn:aws:vpc-lattice:us-east-1:123456789012:servicenetwork/
sn-0101abcd5432abcd0 \
    --security-group-ids sg-0123456789012abcd
```

출력:

```
{
    "VpcEndpoint": {
        "VpcEndpointId": "vpce-0f00567fa8EXAMPLE",
        "VpcEndpointType": "ServiceNetwork",
        "VpcId": "vpc-111122223333aabbc",
        "State": "Pending",
        "SubnetIds": [
            "subnet-0011aabbcc2233445"
        ],
        "Groups": [
            {
                "GroupId": "sg-0123456789012abcd",
                "GroupName": "my-security-group"
```

```
            }
        ],
        "IpAddressType": "IPV4",
        "PrivateDnsEnabled": false,
        "CreationTimestamp": "2025-02-06T23:44:20.449000+00:00",
        "Tags": [],
        "OwnerId": "123456789012",
        "ServiceNetworkArn": "arn:aws:vpc-lattice:us-
east-1:123456789012:servicenetwork/sn-0101abcd5432abcd0"
    }
}
```

자세한 내용은 AWS PrivateLink 사용 설명서의 Service network endpoints를 참조하시기 바랍니다.

- API 세부 정보는 AWS CLI 명령 참조의 CreateVpcEndpoint 섹션을 참조하세요.

`create-vpc-peering-connection`

다음 코드 예시는 `create-vpc-peering-connection`의 사용 방법을 보여 줍니다.

AWS CLI

VPC 간에 VPC 피어링 연결 생성

이 예시에서는 VPC인 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"
        }
    }
}
```

또 다른 계정에 있는 VPC와의 VPC 피어링 연결 생성

이 예시에서는 VPC(vpc-1a2b3c4d)와 123456789012 AWS 계정에 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 리전의 VPC와 VPC 피어링 연결 생성

이 예시에서는 현재 리전 내 VPC(vpc-1a2b3c4d)와 리전 내 `us-west-2` 계정에 있는 VPC(vpc-11122233) 간의 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 예시에서는 현재 리전 내 VPC(vpc-1a2b3c4d)와 `us-west-2` 리전 내 123456789012 AWS 계정에 속한 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
```

- API 세부 정보는 AWS CLI 명령 참조의 [CreateVpcPeeringConnection](#) 섹션을 참조하세요.

create-vpc

다음 코드 예시는 create-vpc의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: VPC를 생성하는 방법

다음 create-vpc 예제에서는 지정된 IPv4 CIDR 블록과 이름 태그를 사용하여 VPC를 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 테넌시를 사용하여 VPC를 생성하는 방법

다음 `create-vpc` 예제에서는 지정된 IPv4 CIDR 블록과 전용 테넌시를 사용하여 VPC를 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 블록을 사용하여 VPC를 생성하는 방법

다음 `create-vpc` 예제에서는 Amazon에서 제공하는 IPv6 CIDR 블록을 사용하여 VPC를 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 풀에서 CIDR을 사용하여 VPC를 생성하는 방법

다음 `create-vpc` 예제에서는 Amazon VPC IP 주소 관리자(IPAM) 풀에서 CIDR을 사용하여 VPC를 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 설명서에서 IPAM 풀 CIDR을 사용하는 VPC 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조에서 CreateVpc를 참조하세요.

create-vpn-connection-route

다음 코드 예시는 create-vpn-connection-route의 사용 방법을 보여 줍니다.

AWS CLI

VPN 연결을 위한 정적 경로 생성

이 예시에서는 지정된 VPN 연결에 대한 고정 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
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateVpnConnectionRoute 섹션을 참조하세요.

create-vpn-connection

다음 코드 예시는 create-vpn-connection의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 동적 라우팅을 사용하여 VPN 연결 생성

다음 create-vpn-connection 예시에서는 지정된 가상 프라이빗 게이트웨이와 지정된 고객 게이트웨이 사이에 VPN 연결을 만들고 VPN 연결에 태그를 적용합니다. 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 \
    --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 AWS Site-to-Site VPN 작동 방식을 참조하세요.

예시 2: 정적 라우팅을 사용하여 VPN 연결 생성

다음 `create-vpn-connection` 예시에서는 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 AWS Site-to-Site VPN 작동 방식을 참조하세요.

예시 3: VPN 연결 생성 및 내부 CIDR 및 사전 공유 키 지정

다음 `create-vpn-connection` 예시에서는 VPN 연결을 생성하고 각 터널에 대해 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 AWS Site-to-Site VPN 작동 방식을 참조하세요.

예시 4: IPv6 트래픽을 지원하는 VPN 연결 생성

다음 `create-vpn-connection` 예시에서는 지정된 전송 게이트웨이와 지정된 고객 게이트웨이 간에 IPv6 트래픽을 지원하는 VPN 연결을 생성합니다. 두 터널의 터널 옵션은 AWS가 IKE 협상을 시작하도록 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 AWS Site-to-Site VPN 작동 방식을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateVpnConnection 섹션을 참조하세요.

create-vpn-gateway

다음 코드 예시는 `create-vpn-gateway`의 사용 방법을 보여 줍니다.

AWS CLI

가상 프라이빗 게이트웨이 생성

이 예시에서는 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 측 ASN을 사용하여 가상 프라이빗 게이트웨이 생성

이 예시에서는 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": []
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateVpnGateway 섹션을 참조하세요.

delete-carrier-gateway

다음 코드 예시는 delete-carrier-gateway의 사용 방법을 보여 줍니다.

AWS CLI

통신 사업자 게이트웨이 삭제

다음 delete-carrier-gateway 예시에서는 지정된 통신 사업자 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.

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteCarrierGateway] 섹션을 참조하세요.

delete-client-vpn-endpoint

다음 코드 예시는 delete-client-vpn-endpoint의 사용 방법을 보여 줍니다.

AWS CLI

Client VPN 엔드포인트 삭제

다음 delete-client-vpn-endpoint 예시에서는 지정된 클라이언트 VPN 엔드포인트를 삭제합니다.

```
aws ec2 delete-client-vpn-endpoint \
    --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde
```

출력:

```
{
    "Status": {
        "Code": "deleting"
    }
}
```

자세한 내용은 AWS Client VPN 관리자 안내서의 [Client VPN 엔드포인트]를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteClientVpnEndpoint] 섹션을 참조하세요.

delete-client-vpn-route

다음 코드 예시는 delete-client-vpn-route의 사용 방법을 보여 줍니다.

AWS CLI

Client VPN 엔드포인트 경로 삭제

다음 delete-client-vpn-route 예시에서는 Client 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 Client VPN 관리자 안내서의 라우팅을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteClientVpnRoute 섹션을 참조하세요.

delete-coip-cidr

다음 코드 예시는 delete-coip-cidr의 사용 방법을 보여 줍니다.

AWS CLI

다양한 고객 소유 IP(CoIP) 주소 삭제

다음 delete-coip-cidr 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteCoipCidr](#) 섹션을 참조하세요.

delete-coip-pool

다음 코드 예시는 `delete-coip-pool`의 사용 방법을 보여 줍니다.

AWS CLI

고객 소유 IP(CoIP) 주소 풀 삭제

다음 `delete-coip-pool` 예시에서는 CoIP 주소의 CoIP 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 주소](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteCoipPool](#) 섹션을 참조하세요.

delete-customer-gateway

다음 코드 예시는 `delete-customer-gateway`의 사용 방법을 보여 줍니다.

AWS CLI

고객 게이트웨이 삭제

이 예시에서는 지정된 고객 게이트웨이를 삭제합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 delete-customer-gateway --customer-gateway-id cgw-0e11f167
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteCustomerGateway 섹션을 참조하세요.

delete-dhcp-options

다음 코드 예시는 delete-dhcp-options의 사용 방법을 보여 줍니다.

AWS CLI

DHCP 옵션 세트 삭제

이 예시에서는 지정된 DHCP 옵션 세트를 삭제합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 delete-dhcp-options --dhcp-options-id dopt-d9070ebb
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteDhcpOptions 섹션을 참조하세요.

delete-egress-only-internet-gateway

다음 코드 예시는 delete-egress-only-internet-gateway의 사용 방법을 보여 줍니다.

AWS CLI

외부 전용 인터넷 게이트웨이 삭제

이 예시에서는 지정된 외부 전용 인터넷 게이트웨이를 삭제합니다.

명령:

```
aws ec2 delete-egress-only-internet-gateway --egress-only-internet-gateway-
id eigw-01eadbd45ecd7943f
```

출력:

```
{
```

```
  "ReturnCode": true
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteEgressOnlyInternetGateway 섹션을 참조하세요.

delete-fleets

다음 코드 예시는 delete-fleets의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: EC2 플릿을 삭제하고 연결된 인스턴스 종료

다음 delete-fleets 예시에서는 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.

예시 2: 연결된 인스턴스의 종료 없이 EC2 플릿 삭제

다음 delete-fleets 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteFleets 섹션을 참조하세요.

delete-flow-logs

다음 코드 예시는 `delete-flow-logs`의 사용 방법을 보여 줍니다.

AWS CLI

흐름 로그 삭제

다음 `delete-flow-logs` 예시에서는 지정된 흐름 로그를 삭제합니다.

```
aws ec2 delete-flow-logs --flow-log-id fl-11223344556677889
```

출력:

```
{
    "Unsuccessful": []
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteFlowLogs 섹션을 참조하세요.

delete-fpga-image

다음 코드 예시는 delete-fpga-image의 사용 방법을 보여 줍니다.

AWS CLI

Amazon FPGA 이미지 삭제

이 예시에서는 지정된 AFI를 삭제합니다.

명령:

```
aws ec2 delete-fpga-image --fpga-image-id afi-06b12350a123fbabc
```

출력:

```
{
  "Return": true
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteFpgaImage 섹션을 참조하세요.

delete-instance-connect-endpoint

다음 코드 예시는 delete-instance-connect-endpoint의 사용 방법을 보여 줍니다.

AWS CLI

EC2 Instance Connect 엔드포인트 삭제

다음 delete-instance-connect-endpoint 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteInstanceConnectEndpoint 섹션을 참조하세요.

delete-instance-event-window

다음 코드 예시는 delete-instance-event-window의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 이벤트 기간 삭제

다음 delete-instance-event-window 예시에서는 이벤트 기간을 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 사항 섹션을 참조하세요.

예시 2: 이벤트 기간 강제 삭제

다음 delete-instance-event-window 예시에서는 이벤트 기간이 현재 타겟과 연결되어 있는 경우 이벤트 기간을 강제로 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 사항 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteInstanceEventWindow 섹션을 참조하세요.

delete-internet-gateway

다음 코드 예시는 delete-internet-gateway의 사용 방법을 보여 줍니다.

AWS CLI

인터넷 게이트웨이 삭제

다음 delete-internet-gateway 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteInternetGateway 섹션을 참조하세요.

`delete-ipam-pool`

다음 코드 예시는 `delete-ipam-pool`의 사용 방법을 보여 줍니다.

AWS CLI

IPAM 풀 삭제

이 예시에서는 더 이상 필요하지 않은 IPAM 풀을 삭제하려는 IPAM 위임된 관리자인데 해당 풀에 CIDR이 프로비저닝되어 있다고 가정합니다. 풀에 CIDR이 프로비저닝된 경우 `--cascade` 옵션을 사용하지 않는 한 풀을 삭제할 수 없으므로 `--cascade`를 사용하게 됩니다.

이 요청을 완료하는 방법:

[describe-ipam-pools](https://docs.aws.amazon.com/cli/latest/reference/ec2/describe-ipam-pools.html)로 얻을 수 있는 IPAM 풀 ID가 필요합니다. `--region`는 IPAM 홈 리전이어야 합니다.

다음 `delete-ipam-pool` 예시에서는 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 VPC IPv6 풀을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteIpamPool 섹션을 참조하세요.

`delete-ipam-resource-discovery`

다음 코드 예시는 `delete-ipam-resource-discovery`의 사용 방법을 보여 줍니다.

AWS CLI

리소스 검색 삭제

이 예시에서는 IPAM을 조직 외부의 계정과 통합하는 과정에서 다른 IPAM 관리자와 공유하기 위해 만든 기본값이 아닌 리소스 검색을 삭제하려는 IPAM 위임된 관리자를 가정합니다.

이 요청을 완료하는 방법:

`--region`은 리소스 검색을 생성한 리전이어야 하며, `"IsDefault": true`인 경우 기본 리소스 검색을 삭제할 수 없습니다. 기본 리소스 검색은 IPAM을 생성하는 계정에서 자동으로 생성되는 리소스 검색입니다. 기본 리소스 검색을 삭제하려면 IPAM을 삭제해야 합니다.

다음 `delete-ipam-resource-discovery` 예시에서는 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 검색으로 작업](https://docs.aws.amazon.com/vpc/latest/ipam/res-disc-work-with.html)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteIpamResourceDiscovery](https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ec2/delete-ipam-resource-discovery.html) 섹션을 참조하세요.

delete-ipam-scope

다음 코드 예시는 `delete-ipam-scope`의 사용 방법을 보여 줍니다.

AWS CLI

IPAM 범위 삭제

다음 `delete-ipam-scope` 예시에서는 IPAM을 삭제합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteIpamScope 섹션을 참조하세요.

delete-ipam

다음 코드 예시는 delete-ipam의 사용 방법을 보여 줍니다.

AWS CLI

IPAM 삭제

다음 delete-ipam 예시에서는 IPAM을 삭제합니다.

```
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 IPAM 삭제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteIpam 섹션을 참조하세요.

delete-key-pair

다음 코드 예시는 delete-key-pair의 사용 방법을 보여 줍니다.

AWS CLI

키 페어 삭제

다음 delete-key-pair 예시에서는 지정된 키 페어를 삭제합니다.

```
aws ec2 delete-key-pair \
    --key-name my-key-pair
```

출력:

```
{
    "Return": true,
    "KeyPairId": "key-03c8d3aceb53b507"
}
```

자세한 내용은 AWS Command Line Interface 사용 설명서의 키 페어 사용 및 삭제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteKeyPair를 참조하세요.

delete-launch-template-versions

다음 코드 예시는 delete-launch-template-versions의 사용 방법을 보여 줍니다.

AWS CLI

시작 템플릿 버전 삭제

이 예시에서는 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"
      }
  ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteLaunchTemplateVersions 섹션을 참조하세요.

delete-launch-template

다음 코드 예시는 delete-launch-template의 사용 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조에서 DeleteLaunchTemplate을 참조하세요.

delete-local-gateway-route-table-virtual-interface-group-association

다음 코드 예시는 delete-local-gateway-route-table-virtual-interface-group-association의 사용 방법을 보여 줍니다.

AWS CLI

로컬 게이트웨이 라우팅 테이블을 가상 인터페이스(VIFs) 그룹에서 연결 해제

다음 delete-local-gateway-route-table-virtual-interface-group-association 예시에서는 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 AWS Outposts 사용 설명서의 VF 그룹 연결을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteLocalGatewayRouteTableVirtualInterfaceGroupAssociation 섹션을 참조하세요.

delete-local-gateway-route-table-vpc-association

다음 코드 예시는 `delete-local-gateway-route-table-vpc-association`의 사용 방법을 보여 줍니다.

AWS CLI

VPC에서 로컬 게이트웨이 라우팅 테이블 연결 해제

다음 `delete-local-gateway-route-table-vpc-association` 예시에서는 지정된 로컬 게이트웨이 라우팅 테이블과 VPC 간의 연결을 삭제합니다.

```
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 AWS Outposts 사용 설명서의 VPC 연결을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteLocalGatewayRouteTableVpcAssociation 섹션을 참조하세요.

delete-local-gateway-route-table

다음 코드 예시는 delete-local-gateway-route-table의 사용 방법을 보여 줍니다.

AWS CLI

로컬 게이트웨이 라우팅 테이블 삭제

다음 delete-local-gateway-route-table 예시에서는 직접 VPC 라우팅 모드로 로컬 게이트웨이 라우팅 테이블을 만듭니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteLocalGatewayRouteTable 섹션을 참조하세요.

delete-local-gateway-route

다음 코드 예시는 delete-local-gateway-route의 사용 방법을 보여 줍니다.

AWS CLI

로컬 게이트웨이 라우팅 테이블에서 라우팅 삭제

다음 delete-local-gateway-route 예시에서는 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteLocalGatewayRoute 섹션을 참조하세요.

delete-managed-prefix-list

다음 코드 예시는 delete-managed-prefix-list의 사용 방법을 보여 줍니다.

AWS CLI

접두사 목록 삭제

다음 delete-managed-prefix-list 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteManagedPrefixList 섹션을 참조하세요.

delete-nat-gateway

다음 코드 예시는 delete-nat-gateway의 사용 방법을 보여 줍니다.

AWS CLI

NAT 게이트웨이 삭제

이 예시에서는 NAT 게이트웨이 nat-04ae55e711cec5680을 삭제합니다.

명령:

```
aws ec2 delete-nat-gateway --nat-gateway-id nat-04ae55e711cec5680
```

출력:

```
{
   "NatGatewayId": "nat-04ae55e711cec5680"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteNatGateway 섹션을 참조하세요.

delete-network-acl-entry

다음 코드 예시는 delete-network-acl-entry의 사용 방법을 보여 줍니다.

AWS CLI

네트워크 ACL 항목 삭제

이 예시에서는 지정된 네트워크 ACL에서 수신 규칙 번호 100을 삭제합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 delete-network-acl-entry --network-acl-id acl-5fb85d36 --ingress --rule-number 100
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteNetworkAclEntry 섹션을 참조하세요.

delete-network-acl

다음 코드 예시는 delete-network-acl의 사용 방법을 보여 줍니다.

AWS CLI

네트워크 ACL 삭제

이 예시에서는 지정된 네트워크 ACL을 삭제합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 delete-network-acl --network-acl-id acl-5fb85d36
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteNetworkAcl 섹션을 참조하세요.

delete-network-insights-access-scope-analysis

다음 코드 예시는 delete-network-insights-access-scope-analysis의 사용 방법을 보여 줍니다.

AWS CLI

네트워크 액세스 범위 분석 삭제

다음 delete-network-insights-access-scope-analysis 예시에서는 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 설명서의 AWS CLI를 사용하여 Network Access Analyzer 시작하기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteNetworkInsightsAccessScopeAnalysis 섹션을 참조하세요.

delete-network-insights-access-scope

다음 코드 예시는 delete-network-insights-access-scope의 사용 방법을 보여 줍니다.

AWS CLI

네트워크 액세스 범위 삭제

다음 delete-network-insights-access-scope 예시에서는 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 설명서의 AWS CLI를 사용하여 Network Access Analyzer 시작하기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteNetworkInsightsAccessScope 섹션을 참조하세요.

delete-network-insights-analysis

다음 코드 예시는 delete-network-insights-analysis의 사용 방법을 보여 줍니다.

AWS CLI

경로 분석 삭제

다음 delete-network-insights-analysis 예시에서는 지정된 분석을 삭제합니다.

```
aws ec2 delete-network-insights-analysis \
    --network-insights-analysis-id nia-02207aa13eb480c7a
```

출력:

```
{
    "NetworkInsightsAnalysisId": "nia-02207aa13eb480c7a"
}
```

자세한 내용은 Reachability Analyzer 설명서의 AWS CLI를 사용하여 시작하기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteNetworkInsightsAnalysis 섹션을 참조하세요.

delete-network-insights-path

다음 코드 예시는 delete-network-insights-path의 사용 방법을 보여 줍니다.

AWS CLI

경로 삭제

다음 delete-network-insights-path 예시에서는 지정된 경로를 삭제합니다. 경로를 삭제하려면 먼저 delete-network-insights-analysis 명령을 사용하여 해당 경로의 모든 분석을 삭제해야 합니다.

```
aws ec2 delete-network-insights-path \
    --network-insights-path-id nip-0b26f224f1d131fa8
```

출력:

```
{
    "NetworkInsightsPathId": "nip-0b26f224f1d131fa8"
}
```

자세한 내용은 Reachability Analyzer 설명서의 AWS CLI를 사용하여 시작하기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteNetworkInsightsPath 섹션을 참조하세요.

delete-network-interface-permission

다음 코드 예시는 delete-network-interface-permission의 사용 방법을 보여 줍니다.

AWS CLI

네트워크 인터페이스 권한 생성

이 예시에서는 지정된 네트워크 인터페이스 권한을 삭제합니다.

명령:

```
aws ec2 delete-network-interface-permission --network-interface-permission-id eni-perm-06fd19020ede149ea
```

출력:

```
{
  "Return": true
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteNetworkInterfacePermission 섹션을 참조하세요.

delete-network-interface

다음 코드 예시는 delete-network-interface의 사용 방법을 보여 줍니다.

AWS CLI

네트워크 인터페이스 삭제

이 예시에서는 지정된 네트워크 인터페이스를 삭제합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 delete-network-interface --network-interface-id eni-e5aa89a3
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteNetworkInterface 섹션을 참조하세요.

`delete-placement-group`

다음 코드 예시는 `delete-placement-group`의 사용 방법을 보여 줍니다.

AWS CLI

배치 그룹 삭제

이 예시 명령은 지정된 배치 그룹을 삭제합니다.

명령:

```
aws ec2 delete-placement-group --group-name my-cluster
```

- API 세부 정보는 AWS CLI 명령 참조의 DeletePlacementGroup 섹션을 참조하세요.

`delete-queued-reserved-instances`

다음 코드 예시는 `delete-queued-reserved-instances`의 사용 방법을 보여 줍니다.

AWS CLI

대기 중인 구매 삭제

다음 `delete-queued-reserved-instances` 예시에서는 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": []
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteQueuedReservedInstances 섹션을 참조하세요.

`delete-route-table`

다음 코드 예시는 `delete-route-table`의 사용 방법을 보여 줍니다.

AWS CLI

라우팅 테이블 삭제

이 예시에서는 지정된 라우팅 테이블을 삭제합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 delete-route-table --route-table-id rtb-22574640
```

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteRouteTable](#) 섹션을 참조하세요.

`delete-route`

다음 코드 예시는 `delete-route`의 사용 방법을 보여 줍니다.

AWS CLI

경로 삭제

이 예시에서는 지정된 라우팅 테이블에서 지정된 경로를 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
```

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteRoute](#) 섹션을 참조하세요.

`delete-security-group`

다음 코드 예시는 `delete-security-group`의 사용 방법을 보여 줍니다.

AWS CLI

[EC2-Classic] 보안 그룹을 삭제하는 방법

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 삭제하는 방법

이 예제에서는 ID가 sg-903004f8인 보안 그룹을 삭제합니다. 이름으로 EC2-VPC에 대한 보안 그룹을 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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteSecurityGroup을 참조하세요.

delete-snapshot

다음 코드 예시는 delete-snapshot의 사용 방법을 보여 줍니다.

AWS CLI

스냅샷 삭제

이 예제 명령은 스냅샷 ID가 snap-1234567890abcdef0인 스냅샷을 삭제합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 delete-snapshot --snapshot-id snap-1234567890abcdef0
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteSnapshot을 참조하세요.

delete-spot-datafeed-subscription

다음 코드 예시는 delete-spot-datafeed-subscription의 사용 방법을 보여 줍니다.

AWS CLI

스팟 인스턴스 데이터 피드 구독 취소

이 예시에서는 계정에 대한 스팟 데이터 피드 구독을 삭제합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 delete-spot-datafeed-subscription
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteSpotDatafeedSubscription을 참조하세요.

delete-subnet-cidr-reservation

다음 코드 예시는 delete-subnet-cidr-reservation의 사용 방법을 보여 줍니다.

AWS CLI

서브넷 CIDR 예약 삭제

다음 delete-subnet-cidr-reservation 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteSubnetCidrReservation 섹션을 참조하세요.

delete-subnet

다음 코드 예시는 delete-subnet의 사용 방법을 보여 줍니다.

AWS CLI

서브넷 삭제

이 예시에서는 지정된 서브넷을 삭제합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 delete-subnet --subnet-id subnet-9d4a7b6c
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteSubnet 섹션을 참조하세요.

delete-tags

다음 코드 예시는 delete-tags의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 리소스에서 태그 삭제

다음 delete-tags 예시에서는 지정된 이미지에서 태그 Stack=Test를 삭제합니다. 값과 키 이름을 모두 지정하는 경우 태그의 값이 지정된 값과 일치하는 경우에만 태그가 삭제됩니다.

```
aws ec2 delete-tags \
    --resources ami-1234567890abcdef0 \
    --tags Key=Stack,Value=Test
```

태그의 값을 지정하는 것은 선택 사항입니다. 다음 delete-tags 태그의 태그 값에 관계없이 지정된 인스턴스에서 키 이름이 purpose인 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 예시에서는 삭제할 태그의 태그 값으로 빈 문자열을 지정합니다.

```
aws ec2 delete-tags \
    --resources i-1234567890abcdef0 \
    --tags Key=Name,Value=
```

예시 2: 여러 리소스에서 태그 삭제

다음 delete-tags 예시에서는 인스턴스와 AMI 모두에서 태그``Purpose=Test``를 삭제합니다. 이전 예시에서와 같이 명령에서 태그 값을 생략할 수 있습니다.

```
aws ec2 delete-tags \
    --resources i-1234567890abcdef0 ami-1234567890abcdef0 \
    --tags Key=Purpose
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteTags 섹션을 참조하세요.

delete-traffic-mirror-filter-rule

다음 코드 예시는 delete-traffic-mirror-filter-rule의 사용 방법을 보여 줍니다.

AWS CLI

트래픽 미러 필터 규칙 삭제

다음 delete-traffic-mirror-filter-rule 예시에서는 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 AWS Traffic Mirroring 설명서의 트래픽 미러 필터 규칙 수정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteTrafficMirrorFilterRule 섹션을 참조하세요.

delete-traffic-mirror-filter

다음 코드 예시는 delete-traffic-mirror-filter의 사용 방법을 보여 줍니다.

AWS CLI

트래픽 미러 필터 삭제

다음 delete-traffic-mirror-filter 예시에서는 지정된 트래픽 미러 필터를 삭제합니다.

```
aws ec2 delete-traffic-mirror-filter \
    --traffic-mirror-filter-id tmf-0be0b25fcdEXAMPLE
```

출력:

```
{
    "TrafficMirrorFilterId": "tmf-0be0b25fcdEXAMPLE"
}
```

자세한 내용은 AWS Traffic Mirroring 설명서의 트래픽 미러 필터 삭제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteTrafficMirrorFilter 섹션을 참조하세요.

delete-traffic-mirror-session

다음 코드 예시는 delete-traffic-mirror-session의 사용 방법을 보여 줍니다.

AWS CLI

트래픽 미러 세션 생성

다음 delete-traffic-mirror-session 예시에서는 지정된 트래픽 미러 세션을 삭제합니다.

```
aws ec2 delete-traffic-mirror-session \
    --traffic-mirror-session-id tms-0af3141ce5EXAMPLE
```

출력:

```
{
    "TrafficMirrorSessionId": "tms-0af3141ce5EXAMPLE"
}
```

자세한 내용은 AWS Traffic Mirroring 설명서의 트래픽 미러 세션 삭제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteTrafficMirrorSession 섹션을 참조하세요.

delete-traffic-mirror-target

다음 코드 예시는 `delete-traffic-mirror-target`의 사용 방법을 보여 줍니다.

AWS CLI

트래픽 미러 대상 삭제

다음 `delete-traffic-mirror-target` 예시에서는 지정된 트래픽 미러 대상을 삭제합니다.

```
aws ec2 delete-traffic-mirror-target \
    --traffic-mirror-target-id tmt-060f48ce9EXAMPLE
```

출력:

```
{
    "TrafficMirrorTargetId": "tmt-060f48ce9EXAMPLE"
}
```

자세한 내용은 AWS Traffic Mirroring 설명서의 트래픽 미러 대상 삭제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteTrafficMirrorTarget 섹션을 참조하세요.

delete-transit-gateway-connect-peer

다음 코드 예시는 `delete-transit-gateway-connect-peer`의 사용 방법을 보여 줍니다.

AWS CLI

Transit Gateway Connect 피어 삭제

다음 `delete-transit-gateway-connect-peer` 예시에서는 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 설명서의 전송 게이트웨이 Connect 연결 및 Transit Gateway Connect 피어를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteTransitGatewayConnectPeer 섹션을 참조하세요.

delete-transit-gateway-connect

다음 코드 예시는 `delete-transit-gateway-connect`의 사용 방법을 보여 줍니다.

AWS CLI

Transit Gateway Connect 연결 삭제

다음 `delete-transit-gateway-connect` 예시에서는 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 설명서의 전송 게이트웨이 Connect 연결 및 Transit Gateway Connect 피어를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteTransitGatewayConnect 섹션을 참조하세요.

delete-transit-gateway-multicast-domain

다음 코드 예시는 `delete-transit-gateway-multicast-domain`의 사용 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 멀티캐스트 도메인 삭제

다음 `delete-transit-gateway-multicast-domain` 예시에서는 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 설명서의 Managing multicast domains 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteTransitGatewayMulticastDomain 섹션을 참조하세요.

delete-transit-gateway-peering-attachment

다음 코드 예시는 delete-transit-gateway-peering-attachment의 사용 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 피어링 연결 삭제

다음 delete-transit-gateway-peering-attachment 예시에서는 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 설명서의 [Transit Gateway Peering Attachments](https://docs.aws.amazon.com/vpc/latest/tgw/tgw-peering.html) 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteTransitGatewayPeeringAttachment](https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ec2/delete-transit-gateway-peering-attachment.html) 섹션을 참조하세요.

`delete-transit-gateway-policy-table`

다음 코드 예시는 `delete-transit-gateway-policy-table`의 사용 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 정책 테이블 삭제

다음 `delete-transit-gateway-policy-table` 예시에서는 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 전송 게이트웨이 정책 테이블을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteTransitGatewayPolicyTable 섹션을 참조하세요.

`delete-transit-gateway-prefix-list-reference`

다음 코드 예시는 `delete-transit-gateway-prefix-list-reference`의 사용 방법을 보여 줍니다.

AWS CLI

접두사 목록 참조 삭제

다음 `delete-transit-gateway-prefix-list-reference` 예시에서는 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 설명서의 접두사 목록 참조를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteTransitGatewayPrefixListReference](#) 섹션을 참조하세요.

delete-transit-gateway-route-table

다음 코드 예시는 `delete-transit-gateway-route-table`의 사용 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 라우팅 테이블 삭제

다음 `delete-transit-gateway-route-table` 예시에서는 지정된 전송 게이트웨이 라우팅 테이블 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 설명서의 [전송 게이트웨이 라우팅 테이블 삭제](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteTransitGatewayRouteTable](#) 섹션을 참조하세요.

delete-transit-gateway-route

다음 코드 예시는 `delete-transit-gateway-route`의 사용 방법을 보여 줍니다.

AWS CLI

라우팅 테이블에서 CIDR 블록 삭제

다음 `delete-transit-gateway-route` 예시에서는 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 설명서의 정적 라우팅 삭제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteTransitGatewayRoute 섹션을 참조하세요.

`delete-transit-gateway-vpc-attachment`

다음 코드 예시는 `delete-transit-gateway-vpc-attachment`의 사용 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 VPC 연결 삭제

다음 `delete-transit-gateway-vpc-attachment` 예시에서는 지정된 VPC 연결을 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 Gateway 설명서의 VPC 연결 삭제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteTransitGatewayVpcAttachment 섹션을 참조하세요.

delete-transit-gateway

다음 코드 예시는 `delete-transit-gateway`의 사용 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 삭제

다음 `delete-transit-gateway` 예시에서는 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 설명서의 전송 게이트웨이 삭제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteTransitGateway 섹션을 참조하세요.

delete-verified-access-endpoint

다음 코드 예시는 delete-verified-access-endpoint의 사용 방법을 보여 줍니다.

AWS CLI

Verified·Access 엔드포인트 삭제

다음 delete-verified-access-endpoint 예시에서는 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 AWS Verified Access 사용 설명서의 Verified Access endpoints 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteVerifiedAccessEndpoint 섹션을 참조하세요.

delete-verified-access-group

다음 코드 예시는 delete-verified-access-group의 사용 방법을 보여 줍니다.

AWS CLI

Verified·Access 그룹 삭제

다음 delete-verified-access-group 예시에서는 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 AWS Verified Access 사용 설명서의 Verified Access groups 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteVerifiedAccessGroup 섹션을 참조하세요.

delete-verified-access-instance

다음 코드 예시는 delete-verified-access-instance의 사용 방법을 보여 줍니다.

AWS CLI

Verified·Access 인스턴스 삭제

다음 delete-verified-access-instance 예시에서는 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 AWS Verified Access 사용 설명서의 Verified Access instances 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteVerifiedAccessInstance](https://docs.aws.amazon.com/goto/aws-cli/ec2-2016-11-15/DeleteVerifiedAccessInstance) 섹션을 참조하세요.

delete-verified-access-trust-provider

다음 코드 예시는 delete-verified-access-trust-provider의 사용 방법을 보여 줍니다.

AWS CLI

Verified·Access 신뢰 공급자 삭제

다음 delete-verified-access-trust-provider 예시에서는 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 AWS Verified Access 사용 설명서의 Trust providers for Verified Access 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteVerifiedAccessTrustProvider 섹션을 참조하세요.

delete-volume

다음 코드 예시는 delete-volume의 사용 방법을 보여 줍니다.

AWS CLI

볼륨 삭제

이 예시 명령은 볼륨 ID가 vol-049df61146c4d7901인 사용 가능한 볼륨을 삭제합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 delete-volume --volume-id vol-049df61146c4d7901
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteVolume 섹션을 참조하세요.

delete-vpc-endpoint-connection-notifications

다음 코드 예시는 delete-vpc-endpoint-connection-notifications의 사용 방법을 보여 줍니다.

AWS CLI

엔드포인트 연결 알림 삭제

이 예시에서는 지정된 엔드포인트 연결 알림을 삭제합니다.

명령:

```
aws ec2 delete-vpc-endpoint-connection-notifications --connection-notification-ids vpce-nfn-008776de7e03f5abc
```

출력:

```
{
  "Unsuccessful": []
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteVpcEndpointConnectionNotifications 섹션을 참조하세요.

delete-vpc-endpoint-service-configurations

다음 코드 예시는 delete-vpc-endpoint-service-configurations의 사용 방법을 보여 줍니다.

AWS CLI

엔드포인트 서비스 구성 삭제

이 예시에서는 지정된 엔드포인트 서비스 구성을 삭제합니다.

명령:

```
aws ec2 delete-vpc-endpoint-service-configurations --service-ids vpce-svc-03d5ebb7d9579a2b3
```

출력:

```
{
   "Unsuccessful": []
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteVpcEndpointServiceConfigurations 섹션을 참조하세요.

delete-vpc-endpoints

다음 코드 예시는 delete-vpc-endpoints의 사용 방법을 보여 줍니다.

AWS CLI

엔드포인트 삭제

이 예시에서는 엔드포인트 vpce-aa22bb33 및 vpce-1a2b3c4d를 삭제합니다. 명령이 부분적으로 성공하거나 실패한 경우 실패한 항목의 목록이 반환됩니다. 명령이 성공하면 반환된 목록은 비어 있습니다.

명령:

```
aws ec2 delete-vpc-endpoints --vpc-endpoint-ids vpce-aa22bb33 vpce-1a2b3c4d
```

출력:

```
{
  "Unsuccessful": []
```

```
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteVpcEndpoints 섹션을 참조하세요.

delete-vpc-peering-connection

다음 코드 예시는 delete-vpc-peering-connection의 사용 방법을 보여 줍니다.

AWS CLI

VPC 피어링 연결 삭제

이 예시에서는 지정된 VPC 피어링 연결을 삭제합니다.

명령:

```
aws ec2 delete-vpc-peering-connection --vpc-peering-connection-id pcx-1a2b3c4d
```

출력:

```
{
    "Return": true
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteVpcPeeringConnection 섹션을 참조하세요.

delete-vpc

다음 코드 예시는 delete-vpc의 사용 방법을 보여 줍니다.

AWS CLI

VPC를 삭제하는 방법

이 예시에서는 지정된 VPC를 삭제합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 delete-vpc --vpc-id vpc-a01106c2
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteVpc 섹션을 참조하세요.

delete-vpn-connection-route

다음 코드 예시는 delete-vpn-connection-route의 사용 방법을 보여 줍니다.

AWS CLI

VPN 연결에서 정적 경로 삭제

이 예시에서는 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
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteVpnConnectionRoute 섹션을 참조하세요.

delete-vpn-connection

다음 코드 예시는 delete-vpn-connection의 사용 방법을 보여 줍니다.

AWS CLI

VPN 연결 삭제

이 예시에서는 지정된 VPN 연결을 삭제합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 delete-vpn-connection --vpn-connection-id vpn-40f41529
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteVpnConnection 섹션을 참조하세요.

delete-vpn-gateway

다음 코드 예시는 delete-vpn-gateway의 사용 방법을 보여 줍니다.

AWS CLI

가상 프라이빗 게이트웨이 삭제

이 예시에서는 지정된 가상 프라이빗 게이트웨이를 삭제합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 delete-vpn-gateway --vpn-gateway-id vgw-9a4cacf3
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteVpnGateway 섹션을 참조하세요.

deprovision-byoip-cidr

다음 코드 예시는 deprovision-byoip-cidr의 사용 방법을 보여 줍니다.

AWS CLI

IP 주소 범위를 사용에서 제거하는 방법

다음 예시에서는 지정된 주소 범위를 AWS에서 사용하지 않도록 제거합니다.

```
aws ec2 deprovision-byoip-cidr \
    --cidr 203.0.113.25/24
```

출력:

```
{
    "ByoipCidr": {
        "Cidr": "203.0.113.25/24",
        "State": "pending-deprovision"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeprovisionByoipCidr 섹션을 참조하세요.

deprovision-ipam-pool-cidr

다음 코드 예시는 deprovision-ipam-pool-cidr의 사용 방법을 보여 줍니다.

AWS CLI

IPAM 풀 CIDR 프로비저닝 해제

다음 deprovision-ipam-pool-cidr 예시에서는 IPAM 풀에 프로비저닝된 CIDR을 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 풀 CIDR 프로비저닝 해제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeprovisionIpamPoolCidr 섹션을 참조하세요.

deregister-image

다음 코드 예시는 deregister-image의 사용 방법을 보여 줍니다.

AWS CLI

AMI 등록 해제

이 예시에서는 지정된 AMI의 등록을 취소합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 deregister-image --image-id ami-4fa54026
```

- API 세부 정보는 AWS CLI 명령 참조의 [DeregisterImage](#) 섹션을 참조하세요.

`deregister-instance-event-notification-attributes`

다음 코드 예시는 `deregister-instance-event-notification-attributes`의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 이벤트 알림에서 모든 태그 제거

다음 `deregister-instance-event-notification-attributes` 예시에서는 `IncludeAllTagsOfInstance`를 `false`로 설정하는 효과가 있는 `IncludeAllTagsOfInstance=true`를 제거합니다.

```
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 대한 예약된 이벤트](#)를 참조하세요.

예시 2: 이벤트 알림에서 특정 태그 제거

다음 `deregister-instance-event-notification-attributes` 예시에서는 이벤트 알림에 포함된 태그에서 지정된 태그를 제거합니다. 이벤트 알림에 포함된 나머지 태그를 설명하려면 `describe-instance-event-notification-attributes`를 사용합니다.

```
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 대한 예약된 이벤트를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeregisterInstanceEventNotificationAttributes 섹션을 참조하세요.

deregister-transit-gateway-multicast-group-members

다음 코드 예시는 `deregister-transit-gateway-multicast-group-members`의 사용 방법을 보여 줍니다.

AWS CLI

멀티캐스트 그룹에서 멤버 등록 취소

이 예시에서는 전송 게이트웨이 멀티캐스트 그룹에서 지정된 네트워크 인터페이스 그룹 멤버의 등록을 해제합니다.

```
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 AWS Transit Gateways 사용 설명서의 멀티캐스트 그룹에서 멤버 등록 취소를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeregisterTransitGatewayMulticastGroupMembers 섹션을 참조하세요.

deregister-transit-gateway-multicast-group-source

다음 코드 예시는 deregister-transit-gateway-multicast-group-source의 사용 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 멀티캐스트 그룹에서 소스 등록 취소

이 예시에서는 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 AWS Transit Gateways 사용 설명서의 멀티캐스트 그룹에서 소스 등록 취소를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeregisterTransitGatewayMulticastGroupSource 섹션을 참조하세요.

describe-account-attributes

다음 코드 예시는 describe-account-attributes의 사용 방법을 보여 줍니다.

AWS CLI

AWS 계정의 모든 속성 설명

이 예시에서는 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 속성 설명

이 예시에서는 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"
                    }
                ]
            }
        ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeAccountAttributes를 참조하세요.

describe-address-transfers

다음 코드 예시는 describe-address-transfers의 사용 방법을 보여 줍니다.

AWS CLI

탄력적 IP 주소 전송 설명

다음 describe-address-transfers 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeAddressTransfers 섹션을 참조하세요.

describe-addresses-attribute

다음 코드 예시는 describe-addresses-attribute의 사용 방법을 보여 줍니다.

AWS CLI

탄력적 IP 주소와 연결된 도메인 이름의 속성 보기

다음 describe-addresses-attribute 예시에서는 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 설명서의 이메일 애플리케이션에 역방향 DNS 사용을 참조하거나 AWS CLI 명령 참조의 modify-address-attribute를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeAddressesAttribute 섹션을 참조하세요.

describe-addresses

다음 코드 예시는 describe-addresses의 사용 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: EC2-VPC에 대한 탄력적 IP 주소의 세부 정보를 검색하는 방법

다음 describe-addresses 예제에서는 VPC의 인스턴스에서 사용할 탄력적 IP 주소의 세부 정보를 표시합니다.

```
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-VPC의 인스턴스와 연결된, 지정된 할당 ID를 보유한 탄력적 IP 주소의 세부 정보를 표시합니다.

```
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 주소의 세부 정보를 검색하는 방법

다음 describe-addresses 예제에서는 EC2-VPC 내 특정 프라이빗 IP 주소와 연결된 탄력적 IP 주소의 세부 정보를 표시합니다.

```
aws ec2 describe-addresses \
    --filters "Name=private-ip-address,Values=10.251.50.12"
```

예제 5: EC2-Classic에서 탄력적 IP 주소의 세부 정보를 검색하는 방법

다음 describe-addresses 예제에서는 EC2-Classic에서 사용할 탄력적 IP 주소의 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 연결된, 값이 203.0.110.25인 탄력적 IP 주소의 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조에서 DescribeAddresses를 참조하세요.

describe-aggregate-id-format

다음 코드 예시는 describe-aggregate-id-format의 사용 방법을 보여 줍니다.

AWS CLI

특정 리전의 모든 리소스 유형에 대한 더 긴 ID 형식 설정 설명

다음 describe-aggregate-id-format 예시에서는 현재 리전의 전체 긴 ID 형식 지정 상태를 설명합니다. Deadline 값은 이러한 리소스가 짧은 ID 형식에서 긴 ID 형식으로 영구적으로 전환할 수 있는 기한이 만료되었음을 나타냅니다. UseLongIdsAggregated 값은 모든 IAM 사용자 및 IAM 역할이 모든 리소스 유형에 대해 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
        },
        ...
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeAggregateIdFormat 섹션을 참조하세요.

`describe-availability-zones`

다음 코드 예시는 `describe-availability-zones`의 사용 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조에서 DescribeAvailabilityZones를 참조하세요.

describe-aws-network-performance-metric-subscription

다음 코드 예시는 describe-aws-network-performance-metric-subscription의 사용 방법을 보여 줍니다.

AWS CLI

지표 구독 설명

다음 describe-aws-network-performance-metric-subscriptions 예시에서는 지표 구독을 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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeAwsNetworkPerformanceMetricSubscription 섹션을 참조하세요.

describe-aws-network-performance-metric-subscriptions

다음 코드 예시는 describe-aws-network-performance-metric-subscriptions의 사용 방법을 보여 줍니다.

AWS CLI

지표 구독 설명

다음 describe-aws-network-performance-metric-subscriptions 예시에서는 지표 구독을 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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeAwsNetworkPerformanceMetricSubscriptions 섹션을 참조하세요.

describe-bundle-tasks

다음 코드 예시는 describe-bundle-tasks의 사용 방법을 보여 줍니다.

AWS CLI

번들 태스크 설명

이 예시에서는 모든 번들 태스크를 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"
    }
  ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeBundleTasks 섹션을 참조하세요.

describe-byoip-cidrs

다음 코드 예시는 describe-byoip-cidrs의 사용 방법을 보여 줍니다.

AWS CLI

프로비저닝된 주소 범위 설명

다음 describe-byoip-cidrs 예시에서는 AWS에서 사용하도록 프로비저닝한 공용 IPv4 주소 범위에 대한 세부 정보를 표시합니다.

```
aws ec2 describe-byoip-cidrs
```

출력:

```
{
    "ByoipCidrs": [
        {
            "Cidr": "203.0.113.25/24",
            "StatusMessage": "ipv4pool-ec2-1234567890abcdef0",
            "State": "provisioned"
        }
```

```
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeByoipCidrs 섹션을 참조하세요.

describe-capacity-reservation-fleets

다음 코드 예시는 describe-capacity-reservation-fleets의 사용 방법을 보여 줍니다.

AWS CLI

용량 예약 플릿 보기

다음 describe-capacity-reservation-fleets 예시에서는 지정된 용량 예약 플릿에 대한 구성 및 용량 정보를 나열합니다. 또한 플릿 내부에 있는 개별 용량 예약에 대한 세부 정보도 나열됩니다.

```
aws ec2 describe-capacity-reservation-fleets \
    --capacity-reservation-fleet-ids crf-abcdef01234567890
```

출력:

```
{
    "CapacityReservationFleets": [
        {
            "State": "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 예약에 대한 자세한 내용은 Amazon EC2 사용 설명서의 용량 예약 플릿을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeCapacityReservationFleets 섹션을 참조하세요.

describe-capacity-reservations

다음 코드 예시는 `describe-capacity-reservations`의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 하나 이상의 용량 예약 설명

다음 `describe-capacity-reservations` 예시에서는 현재 AWS 리전의 모든 용량 예약에 대한 세부 정보를 표시합니다.

```
aws ec2 describe-capacity-reservations
```

출력:

```
{
    "CapacityReservations": [
        {
            "CapacityReservationId": "cr-1234abcd56EXAMPLE ",
            "OwnerId": "123456789111",
            "CapacityReservationArn": "arn:aws:ec2:us-east-1:123456789111:capacity-
reservation/cr-1234abcd56EXAMPLE",
            "AvailabilityZoneId": "use1-az2",
            "InstanceType": "c5.large",
            "InstancePlatform": "Linux/UNIX",
            "AvailabilityZone": "us-east-1a",
            "Tenancy": "default",
```

```
            "TotalInstanceCount": 1,
            "AvailableInstanceCount": 1,
            "EbsOptimized": true,
            "EphemeralStorage": false,
            "State": "active",
            "StartDate": "2024-10-23T15:00:24+00:00",
            "EndDateType": "unlimited",
            "InstanceMatchCriteria": "open",
            "CreateDate": "2024-10-23T15:00:24+00:00",
            "Tags": [],
            "CapacityAllocations": []
        },
        {
            "CapacityReservationId": "cr-abcdEXAMPLE9876ef ",
            "OwnerId": "123456789111",
            "CapacityReservationArn": "arn:aws:ec2:us-east-1:123456789111:capacity-
reservation/cr-abcdEXAMPLE9876ef",
            "AvailabilityZoneId": "use1-az2",
            "InstanceType": "c4.large",
            "InstancePlatform": "Linux/UNIX",
            "AvailabilityZone": "us-east-1a",
            "Tenancy": "default",
            "TotalInstanceCount": 1,
            "AvailableInstanceCount": 1,
            "EbsOptimized": true,
            "EphemeralStorage": false,
            "State": "cancelled",
            "StartDate": "2024-10-23T15:01:03+00:00",
            "EndDateType": "unlimited",
            "InstanceMatchCriteria": "open",
            "CreateDate": "2024-10-23T15:01:02+00:00",
            "Tags": [],
            "CapacityAllocations": []
        }
    ]
}
```

예시 2: 하나 이상의 용량 예약 설명

다음 `describe-capacity-reservations` 예시에서는 지정된 용량 예약에 대한 세부 정보를 표시합니다.

```
aws ec2 describe-capacity-reservations \
```

```
    --capacity-reservation-ids cr-1234abcd56EXAMPLE
```

출력:

```
{
    "CapacityReservations": [
        {
            "CapacityReservationId": "cr-abcdEXAMPLE9876ef ",
            "OwnerId": "123456789111",
            "CapacityReservationArn": "arn:aws:ec2:us-east-1:123456789111:capacity-
reservation/cr-abcdEXAMPLE9876ef",
            "AvailabilityZoneId": "use1-az2",
            "InstanceType": "c4.large",
            "InstancePlatform": "Linux/UNIX",
            "AvailabilityZone": "us-east-1a",
            "Tenancy": "default",
            "TotalInstanceCount": 1,
            "AvailableInstanceCount": 1,
            "EbsOptimized": true,
            "EphemeralStorage": false,
            "State": "active",
            "StartDate": "2024-10-23T15:01:03+00:00",
            "EndDateType": "unlimited",
            "InstanceMatchCriteria": "open",
            "CreateDate": "2024-10-23T15:01:02+00:00",
            "Tags": [],
            "CapacityAllocations": []
        }
    ]
}
```

자세한 내용은 Linux 인스턴스용 Amazon Elastic Compute Cloud 사용 설명서의 용량 예약 보기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeCapacityReservations 섹션을 참조하세요.

`describe-carrier-gateways`

다음 코드 예시는 `describe-carrier-gateways`의 사용 방법을 보여 줍니다.

AWS CLI

모든 통신 사업자 게이트웨이 설명

다음 `describe-carrier-gateways` 예시에서는 모든 통신 사업자 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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeCarrierGateways 섹션을 참조하세요.

describe-classic-link-instances

다음 코드 예시는 `describe-classic-link-instances`의 사용 방법을 보여 줍니다.

AWS CLI

연결된 EC2-Classic 인스턴스 설명

이 예시에서는 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 예시에서는 연결된 모든 EC2-Classic 인스턴스를 나열하고, VPC vpc-88888888 에 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"
                                }
                        ]
                }
        ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeClassicLinkInstances 섹션을 참조하세요.

describe-client-vpn-authorization-rules

다음 코드 예시는 describe-client-vpn-authorization-rules의 사용 방법을 보여 줍니다.

AWS CLI

Client VPN 엔드포인트에 대한 권한 부여 규칙 설명

다음 describe-client-vpn-authorization-rules 예시에서는 지정된 클라이언트 VPN 엔드포인트에 대한 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 Client VPN 관리자 안내서의 권한 부여 규칙을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeClientVpnAuthorizationRules 섹션을 참조하세요.

`describe-client-vpn-connections`

다음 코드 예시는 `describe-client-vpn-connections`의 사용 방법을 보여 줍니다.

AWS CLI

Client VPN 엔드포인트에 대한 연결 설명

다음 `describe-client-vpn-connections` 예시에서는 지정된 클라이언트 VPN 엔드포인트에 대한 클라이언트 연결 관련 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 AWS Client VPN 관리자 안내서의 클라이언트 구성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeClientVpnConnections 섹션을 참조하세요.

describe-client-vpn-endpoints

다음 코드 예시는 describe-client-vpn-endpoints의 사용 방법을 보여 줍니다.

AWS CLI

Client VPN 엔드포인트 설명

다음 describe-client-vpn-endpoints 예시에서는 모든 클라이언트 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 AWS Client VPN 관리자 안내서의 Client VPN 엔드포인트를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeClientVpnEndpoints 섹션을 참조하세요.

`describe-client-vpn-routes`

다음 코드 예시는 `describe-client-vpn-routes`의 사용 방법을 보여 줍니다.

AWS CLI

Client VPN 엔드포인트에 대한 라우팅 설명

다음 `describe-client-vpn-routes` 예시에서는 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 Client VPN 관리자 안내서의 라우팅을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeClientVpnRoutes 섹션을 참조하세요.

describe-client-vpn-target-networks

다음 코드 예시는 describe-client-vpn-target-networks의 사용 방법을 보여 줍니다.

AWS CLI

Client VPN 엔드포인트의 대상 네트워크 설명

다음 describe-client-vpn-target-networks 예시에서는 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 Client VPN 관리자 안내서의 Target Networks 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeClientVpnTargetNetworks 섹션을 참조하세요.

describe-coip-pools

다음 코드 예시는 describe-coip-pools의 사용 방법을 보여 줍니다.

AWS CLI

고객 소유 IP 주소 풀 설명

다음 describe-coip-pools 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeCoipPools 섹션을 참조하세요.

describe-conversion-tasks

다음 코드 예시는 describe-conversion-tasks의 사용 방법을 보여 줍니다.

AWS CLI

변환 작업의 상태 보기

이 예시에서는 ID가 import-i-ffvko9js인 변환 태스크의 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeConversionTasks 섹션을 참조하세요.

describe-customer-gateways

다음 코드 예시는 `describe-customer-gateways`의 사용 방법을 보여 줍니다.

AWS CLI

고객 게이트웨이 설명

이 예시에서는 고객 게이트웨이를 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 게이트웨이 설명

이 예시에서는 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeCustomerGateways 섹션을 참조하세요.

describe-dhcp-options

다음 코드 예시는 describe-dhcp-options의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: DHCP 옵션 설명

다음 describe-dhcp-options 예시에서는 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 Amazon VPC 사용 설명서의 DHCP 옵션 세트 작업을 참조하세요.

예시 2: DHCP 옵션 설명 및 출력 필터링

다음 `describe-dhcp-options` 예시에서는 도메인 네임 서버에 `example.com`가 있는 DHCP 옵션만 반환하는 필터를 사용하여 DHCP 옵션을 설명합니다. 이 예시에서는 `--query` 파라미터를 사용하여 출력에 구성 정보와 ID만 표시할 수 있습니다.

```
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 Amazon VPC 사용 설명서의 DHCP 옵션 세트 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDhcpOptions 섹션을 참조하세요.

describe-egress-only-internet-gateways

다음 코드 예시는 describe-egress-only-internet-gateways의 사용 방법을 보여 줍니다.

AWS CLI

외부 전용 인터넷 게이트웨이 설명

이 예시에서는 송신 전용 인터넷 게이트웨이를 설명합니다.

명령:

```
aws ec2 describe-egress-only-internet-gateways
```

출력:

```
{
  "EgressOnlyInternetGateways": [
      {
          "EgressOnlyInternetGatewayId": "eigw-015e0e244e24dfe8a",
          "Attachments": [
              {
                  "State": "attached",
                  "VpcId": "vpc-0c62a468"
              }
          ]
      }
  ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeEgressOnlyInternetGateways 섹션을 참조하세요.

describe-elastic-gpus

다음 코드 예시는 describe-elastic-gpus의 사용 방법을 보여 줍니다.

AWS CLI

Elastic GPU 설명

명령:

```
aws ec2 describe-elastic-gpus --elastic-gpu-
ids egpu-12345678901234567890abcdefghijkl
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeElasticGpus 섹션을 참조하세요.

describe-export-image-tasks

다음 코드 예시는 describe-export-image-tasks의 사용 방법을 보여 줍니다.

AWS CLI

이미지 내보내기 태스크 모니터링

다음 describe-export-image-tasks 예시에서는 지정된 내보내기 이미지 태스크의 상태를 확인합니다. Amazon S3의 결과 이미지 파일은 my-export-bucket/exports/export-ami-1234567890abcdef0.vmdk입니다.

```
aws ec2 describe-export-image-tasks \
    --export-image-task-ids export-ami-1234567890abcdef0
```

진행 중인 이미지 내보내기 태스크의 출력입니다.

```
{
    "ExportImageTasks": [
        {
            "ExportImageTaskId": "export-ami-1234567890abcdef0"
            "Progress": "21",
            "S3ExportLocation": {
                "S3Bucket": "my-export-bucket",
                "S3Prefix": "exports/"
            },
            "Status": "active",
            "StatusMessage": "updating"
        }
    ]
}
```

완료된 이미지 내보내기 태스크의 출력입니다.

```
{
    "ExportImageTasks": [
        {
            "ExportImageTaskId": "export-ami-1234567890abcdef0"
            "S3ExportLocation": {
                "S3Bucket": "my-export-bucket",
                "S3Prefix": "exports/"
            },
            "Status": "completed"
        }
    ]
}
```

자세한 내용은 VM Import/Export 사용 설명서의 AMI에서 VM 내보내기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeExportImageTasks 섹션을 참조하세요.

describe-export-tasks

다음 코드 예시는 describe-export-tasks의 사용 방법을 보여 줍니다.

AWS CLI

인스턴스 내보내기 작업에 대한 세부 정보 나열

이 예시에서는 ID export-i-fh8sjjsq의 내보내기 태스크를 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeExportTasks 섹션을 참조하세요.

describe-fast-launch-images

다음 코드 예시는 describe-fast-launch-images의 사용 방법을 보여 줍니다.

AWS CLI

더 빠른 시작을 위해 구성된 Windows AMI에 대한 세부 정보 설명

다음 describe-fast-launch-images 예시에서는 빠른 실행을 위해 구성된 계정의 각 AMI에 대한 세부 정보를 설명합니다. 세부 정보로는 리소스 유형, 스냅샷 구성, 실행 템플릿 세부 정보, 최대 병렬 실행 횟수, AMI 소유자 ID, 빠른 실행 구성의 상태, 상태가 변경된 이유, 상태 변경이 발생한 시간 등이 포함됩니다.

```
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 위해 Windows AMI를 구성하는 방법에 대한 자세한 내용은 Amazon EC2 사용 설명서의 더 빠른 시작을 위해 AMI 구성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeFastLaunchImages 섹션을 참조하세요.

describe-fast-snapshot-restores

다음 코드 예시는 describe-fast-snapshot-restores의 사용 방법을 보여 줍니다.

AWS CLI

빠른 스냅샷 복원 설명

다음 describe-fast-snapshot-restores 예시에서는 상태가 disabled로 설정된 모든 빠른 스냅샷 복원에 대한 세부 정보를 표시합니다.

```
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 예시에서는 모든 빠른 스냅샷 복원을 설명합니다.

```
aws ec2 describe-fast-snapshot-restores
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeFastSnapshotRestores 섹션을 참조하세요.

describe-fleet-history

다음 코드 예시는 describe-fleet-history의 사용 방법을 보여 줍니다.

AWS CLI

EC2 플릿 기록 설명

다음 describe-fleet-history 예시에서는 지정된 시간부터 시작하여 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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeFleetHistory 섹션을 참조하세요.

describe-fleet-instances

다음 코드 예시는 describe-fleet-instances의 사용 방법을 보여 줍니다.

AWS CLI

EC2 플릿에 대해 실행 중인 인스턴스를 설명

다음 `describe-fleet-instances` 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeFleetInstances 섹션을 참조하세요.

describe-fleets

다음 코드 예시는 `describe-fleets`의 사용 방법을 보여 줍니다.

AWS CLI

EC2 플릿 설명

다음 `describe-fleets` 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeFleets 섹션을 참조하세요.

describe-flow-logs

다음 코드 예시는 describe-flow-logs의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 모든 흐름 로그 설명

다음 describe-flow-logs 예시에서는 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}"
            }
        ]
}
```

예시 2: 흐름 로그의 하위 세트 설명

다음 `describe-flow-logs` 예시에서는 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"
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeFlowLogs 섹션을 참조하세요.

describe-fpga-image-attribute

다음 코드 예시는 `describe-fpga-image-attribute`의 사용 방법을 보여 줍니다.

AWS CLI

Amazon FPGA 이미지의 속성 설명

이 예시에서는 지정된 AFI에 대한 로드 권한을 설명합니다.

명령:

```
aws ec2 describe-fpga-image-attribute --fpga-image-id afi-0d123e123bfc85abc --
attribute loadPermission
```

출력:

```
{
  "FpgaImageAttribute": {
      "FpgaImageId": "afi-0d123e123bfc85abc",
      "LoadPermissions": [
          {
              "UserId": "123456789012"
          }
      ]
  }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeFpgaImageAttribute 섹션을 참조하세요.

describe-fpga-images

다음 코드 예시는 describe-fpga-images의 사용 방법을 보여 줍니다.

AWS CLI

Amazon FPGA 이미지 설명

이 예시에서는 123456789012 계정에서 소유한 AFI를 설명합니다.

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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeFpgaImages 섹션을 참조하세요.

describe-host-reservation-offerings

다음 코드 예시는 describe-host-reservation-offerings의 사용 방법을 보여 줍니다.

AWS CLI

전용 호스트 예약 제안 설명

이 예시에서는 구매할 수 있는 M4 인스턴스 제품군에 대한 전용 호스트 예약을 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
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeHostReservationOfferings 섹션을 참조하세요.

describe-host-reservations

다음 코드 예시는 describe-host-reservations의 사용 방법을 보여 줍니다.

AWS CLI

계정의 전용 호스트 예약 설명

이 예시에서는 계정의 전용 호스트 예약을 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
      }
  ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeHostReservations 섹션을 참조하세요.

describe-hosts

다음 코드 예시는 describe-hosts의 사용 방법을 보여 줍니다.

AWS CLI

전용 호스트에 관한 세부 정보 보기

다음 describe-hosts 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeHosts 섹션을 참조하세요.

describe-iam-instance-profile-associations

다음 코드 예시는 describe-iam-instance-profile-associations의 사용 방법을 보여 줍니다.

AWS CLI

IAM 인스턴스 프로파일 연결을 설명하는 방법

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"
            }
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조에서 DescribeIamInstanceProfileAssociations를 참조하세요.

describe-id-format

다음 코드 예시는 `describe-id-format`의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 리소스의 ID 형식 설명

다음 `describe-id-format` 예시에서는 보안 그룹의 ID 형식 지정을 설명합니다.

```
aws ec2 describe-id-format \
    --resource security-group
```

다음 예시 출력에서는 `Deadline` 값이 이 리소스 유형이 짧은 ID 형식에서 긴 ID 형식으로 영구적으로 전환해야 하는 기한이 2018년 8월 15일 00:00 UTC에 만료되었음을 나타냅니다.

```
{
    "Statuses": [
        {
            "Deadline": "2018-08-15T00:00:00.000Z",
            "Resource": "security-group",
            "UseLongIds": true
        }
    ]
}
```

예시 2: 모든 리소스의 ID 형식 설명

다음 `describe-id-format` 예시에서는 모든 리소스 유형에 대한 ID 형식을 설명합니다. 짧은 ID 형식을 지원하던 모든 리소스 유형이 긴 ID 형식을 사용하도록 전환되었습니다.

```
aws ec2 describe-id-format
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeIdFormat 섹션을 참조하세요.

describe-identity-id-format

다음 코드 예시는 describe-identity-id-format의 사용 방법을 보여 줍니다.

AWS CLI

IAM 역할의 ID 형식 설명

다음 describe-identity-id-format 예시에서는 AWS 계정의 EC2Role IAM 역할에 의해 만들어진 인스턴스가 수신하는 ID 형식을 설명합니다.

```
aws ec2 describe-identity-id-format \
    --principal-arn arn:aws:iam::123456789012:role/my-iam-role \
    --resource instance
```

다음 출력에서는 이 역할에 의해 생성된 인스턴스가 긴 ID 형식의 ID를 수신한다는 것을 나타냅니다.

```
{
    "Statuses": [
        {
            "Deadline": "2016-12-15T00:00:00Z",
            "Resource": "instance",
            "UseLongIds": true
        }
    ]
}
```

IAM 사용자의 ID 형식 설명

다음 describe-identity-id-format 예시에서는 AWS 계정의 AdminUser IAM 사용자에 의해 만들어진 스냅샷 수신하는 ID 형식을 설명합니다.

```
aws ec2 describe-identity-id-format \
    --principal-arn arn:aws:iam::123456789012:user/AdminUser \
    --resource snapshot
```

출력은 이 사용자가 만든 스냅샷이 긴 ID 형식의 ID를 받는다는 것을 나타냅니다.

```
{
    "Statuses": [
        {
            "Deadline": "2016-12-15T00:00:00Z",
            "Resource": "snapshot",
            "UseLongIds": true
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeIdentityIdFormat 섹션을 참조하세요.

describe-image-attribute

다음 코드 예시는 `describe-image-attribute`의 사용 방법을 보여 줍니다.

AWS CLI

AMI의 시작 권한 설명

이 예시에서는 지정된 AMI에 대한 실행 권한을 설명합니다.

명령:

```
aws ec2 describe-image-attribute --image-id ami-5731123e --
attribute launchPermission
```

출력:

```
{
    "LaunchPermissions": [
        {
            "UserId": "123456789012"
        }
    ],
    "ImageId": "ami-5731123e",
}
```

AMI의 제품 코드 설명

이 예시에서는 지정된 AMI의 제품 코드를 설명합니다. 이 AMI에는 제품 코드가 없습니다.

명령:

```
aws ec2 describe-image-attribute --image-id ami-5731123e --attribute productCodes
```

출력:

```
{
    "ProductCodes": [],
    "ImageId": "ami-5731123e",
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeImageAttribute 섹션을 참조하세요.

describe-images

다음 코드 예시는 describe-images의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: AMI를 설명하는 방법

다음 describe-images 예제에서는 지정된 리전에서 지정된 AMI를 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 EC2 사용 설명서에서 Amazon Machine Image(AMI)를 참조하세요.

예제 2: 필터를 기반으로 AMI를 설명하는 방법

다음 `describe-images` 예제에서는 Amazon에서 제공하는 Amazon EBS 지원 Windows AMI를 설명합니다.

```
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 설명서에서 리소스 나열 및 필터링을 참조하세요.

예제 3: 태그를 기반으로 AMI를 설명하는 방법

다음 describe-images 예제에서는 Type=Custom 태그가 있는 모든 AMI를 설명합니다. 이 예제에서는 --query 파라미터를 사용하여 AMI ID만 표시합니다.

```
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 설명서에서 [태그 작업]을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeImages]를 참조하세요.

describe-import-image-tasks

다음 코드 예시는 describe-import-image-tasks의 사용 방법을 보여 줍니다.

AWS CLI

이미지 가져오기 태스크 모니터링

다음 describe-import-image-tasks 예시에서는 지정된 가져오기 이미지 태스크의 상태를 확인합니다.

```
aws ec2 describe-import-image-tasks \
    --import-task-ids import-ami-1234567890abcdef0
```

진행 중인 이미지 내보내기 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 내보내기 작업의 출력입니다. 결과 AMI의 ID는 `ImageId`에서 제공합니다.

```
{
    "ImportImageTasks": [
        {
            "ImportTaskId": "import-ami-1234567890abcdef0",
            "ImageId": "ami-1234567890abcdef0",
            "SnapshotDetails": [
                {
                    "DiskImageSize": 705638400.0,
                    "Format": "ova",
                    "SnapshotId": "snap-1234567890abcdef0"
                    "Status": "completed",
                    "UserBucket": {
                        "S3Bucket": "my-import-bucket",
                        "S3Key": "vms/my-server-vm.ova"
                    }
                }
            ],
            "Status": "completed"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeImportImageTasks 섹션을 참조하세요.

`describe-import-snapshot-tasks`

다음 코드 예시는 `describe-import-snapshot-tasks`의 사용 방법을 보여 줍니다.

AWS CLI

스냅샷 가져오기 태스크 모니터링

다음 `describe-import-snapshot-tasks` 예시에서는 지정된 가져오기 스냅샷 태스크의 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 출력: 결과 스냅샷의 ID는 `SnapshotId`에서 제공합니다.

```
{
    "ImportSnapshotTasks": [
        {
            "Description": "My server VMDK",
            "ImportTaskId": "import-snap-1234567890abcdef0",
            "SnapshotTaskDetail": {
                "Description": "My server VMDK",
```

```
                    "DiskImageSize": "705638400.0",
                    "Format": "VMDK",
                    "SnapshotId": "snap-1234567890abcdef0"
                    "Status": "completed",
                    "UserBucket": {
                        "S3Bucket": "my-import-bucket",
                        "S3Key": "vms/my-server-vm.vmdk"
                    }
                }
            }
        ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeImportSnapshotTasks 섹션을 참조하세요.

describe-instance-attribute

다음 코드 예시는 describe-instance-attribute의 사용 방법을 보여 줍니다.

AWS CLI

인스턴스 유형 설명

이 예시에서는 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 속성 설명

이 예시에서는 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 매핑 설명

이 예시에서는 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"
            }
        }
    ],
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeInstanceAttribute 섹션을 참조하세요.

describe-instance-connect-endpoints

다음 코드 예시는 describe-instance-connect-endpoints의 사용 방법을 보여 줍니다.

AWS CLI

EC2 인스턴스 연결 엔드포인트 설명

다음 describe-instance-connect-endpoints 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeInstanceConnectEndpoints 섹션을 참조하세요.

`describe-instance-credit-specifications`

다음 코드 예시는 `describe-instance-credit-specifications`의 사용 방법을 보여 줍니다.

AWS CLI

인스턴스 하나 이상의 CPU 사용량에 대한 크레딧 옵션 설명

다음 `describe-instance-credit-specifications` 예시에서는 지정된 인스턴스에 대한 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 성능 버스트 가능 인스턴스 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeInstanceCreditSpecifications 섹션을 참조하세요.

describe-instance-event-notification-attributes

다음 코드 예시는 describe-instance-event-notification-attributes의 사용 방법을 보여 줍니다.

AWS CLI

예약된 이벤트 알림의 태그 설명

다음 describe-instance-event-notification-attributes 예시에서는 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 대한 예약된 이벤트를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeInstanceEventNotificationAttributes 섹션을 참조하세요.

describe-instance-event-windows

다음 코드 예시는 describe-instance-event-windows의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 모든 이벤트 기간 설명

다음 describe-instance-event-windows 예시에서는 지정된 리전 내의 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"
}
```

예시 2: 특정 이벤트 기간 설명

다음 describe-instance-event-windows 예시에서는 instance-event-window 파라미터를 사용하여 특정 이벤트 윈도우를 설명하는 방식으로 특정 이벤트를 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": []
        }
}
```

예시 3: 하나 이상의 필터와 일치하는 이벤트 기간 설명

다음 `describe-instance-event-windows` 예시에서는 `filter` 파라미터를 사용하여 하나 이상의 필터와 일치하는 이벤트 기간을 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 일치하는 항목이 없으면 인스턴스 태그 또는 전용 호스트 ID(인스턴스가 전용 호스트에 있는 경우)와 같은 이벤트 기간과의 간접 연결로 폴백됩니다.

```
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": []
        }
    ]
}
```

예시에서는 인스턴스가 이벤트 기간과 연결된 전용 호스트에 있습니다.

이벤트 기간 제약 조건은 Amazon EC2 사용 설명서의 고려 사항을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeInstanceEventWindows 섹션을 참조하세요.

describe-instance-image-metadata

다음 코드 예시는 `describe-instance-image-metadata`의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 모든 인스턴스의 AMI 메타데이터 설명

다음 `describe-instance-image-metadata` 예제에서는 지정된 리전에 있는 AWS 계정의 모든 인스턴스의 AMI 메타데이터를 설명합니다.

```
aws ec2 describe-instance-image-metadata \
    --region us-east-1
```

출력:

```
{
```

```
    "InstanceImageMetadata": [
        {
            "InstanceId": "i-1234567890EXAMPLE",
            "InstanceType": "t2.micro",
            "LaunchTime": "2024-08-28T11:25:45+00:00",
            "AvailabilityZone": "us-east-1a",
            "State": {
                "Code": 16,
                "Name": "running"
            },
            "OwnerId": "123412341234",
            "Tags": [
                {
                    "Key": "MyTagName",
                    "Value": "my-tag-value"
                }
            ],
            "ImageMetadata": {
                "ImageId": "ami-0b752bf1df193a6c4",
                "Name": "al2023-ami-2023.5.20240819.0-kernel-6.1-x86_64",
                "OwnerId": "137112412989",
                "State": "available",
                "ImageOwnerAlias": "amazon",
                "CreationDate": "2023-01-25T17:20:40Z",
                "DeprecationTime": "2025-01-25T17:20:40Z",
                "IsPublic": true
            }
        }
    ],
    "NextToken": "...EXAMPLEwIAABAA2JHaFxLnEXAMPLE..."
}
```

자세한 내용은 Amazon EC2 사용 설명서에서 Amazon EC2의 Amazon Machine Image(AMI)를 참조하세요.

예제 2: 지정된 인스턴스의 AMI 메타데이터 설명

다음 `describe-instance-image-metadata` 예제에서는 지정된 인스턴스에 대한 AMI 메타데이터를 설명합니다.

```
aws ec2 describe-instance-image-metadata \
    --region us-east-1 \
    --instance-ids i-1234567890EXAMPLE i-0987654321EXAMPLE
```

출력:

```
{
    "InstanceImageMetadata": [
        {
            "InstanceId": "i-1234567890EXAMPLE",
            "InstanceType": "t2.micro",
            "LaunchTime": "2024-08-28T11:25:45+00:00",
            "AvailabilityZone": "us-east-1a",
            "State": {
                "Code": 16,
                "Name": "running"
            },
            "OwnerId": "123412341234",
            "Tags": [
                {
                    "Key": "MyTagName",
                    "Value": "my-tag-value"
                }
            ],
            "ImageMetadata": {
                "ImageId": "ami-0b752bf1df193a6c4",
                "Name": "al2023-ami-2023.5.20240819.0-kernel-6.1-x86_64",
                "OwnerId": "137112412989",
                "State": "available",
                "ImageOwnerAlias": "amazon",
                "CreationDate": "2023-01-25T17:20:40Z",
                "DeprecationTime": "2025-01-25T17:20:40Z",
                "IsPublic": true
            }
        },
        {
            "InstanceId": "i-0987654321EXAMPLE",
            "InstanceType": "t2.micro",
            "LaunchTime": "2024-08-28T11:25:45+00:00",
            "AvailabilityZone": "us-east-1a",
            "State": {
                "Code": 16,
                "Name": "running"
            },
            "OwnerId": "123412341234",
            "Tags": [
                {
                    "Key": "MyTagName",
```

```
                    "Value": "my-tag-value"
                }
            ],
            "ImageMetadata": {
                "ImageId": "ami-0b752bf1df193a6c4",
                "Name": "al2023-ami-2023.5.20240819.0-kernel-6.1-x86_64",
                "OwnerId": "137112412989",
                "State": "available",
                "ImageOwnerAlias": "amazon",
                "CreationDate": "2023-01-25T17:20:40Z",
                "DeprecationTime": "2025-01-25T17:20:40Z",
                "IsPublic": true
            }
        }
    ]
}
```

자세한 내용은 Amazon EC2 사용 설명서에서 [Amazon EC2의 Amazon Machine Image(AMI)](#)를 참조하세요.

예제 3: 필터를 기반으로 인스턴스의 AMI 메타데이터 설명

다음 `describe-instance-image-metadata` 예제에서는 `us-east-1a` 가용 영역의 `t2.nano` 및 `t2.micro` 인스턴스에 대한 AMI 메타데이터를 설명합니다.

```
aws ec2 describe-instance-image-metadata \
    --region us-east-1 \
    --filters Name=availability-zone,Values=us-east-1a Name=instance-type,Values=t2.nano,t2.micro
```

출력:

```
{
    "InstanceImageMetadata": [
        {
            "InstanceId": "i-1234567890EXAMPLE",
            "InstanceType": "t2.micro",
            "LaunchTime": "2024-08-28T11:25:45+00:00",
            "AvailabilityZone": "us-east-1a",
            "State": {
                "Code": 16,
                "Name": "running"
            },
```

```
                "OwnerId": "123412341234",
                "Tags": [
                    {
                        "Key": "MyTagName",
                        "Value": "my-tag-value"
                    }
                ],
                "ImageMetadata": {
                    "ImageId": "ami-0b752bf1df193a6c4",
                    "Name": "al2023-ami-2023.5.20240819.0-kernel-6.1-x86_64",
                    "OwnerId": "137112412989",
                    "State": "available",
                    "ImageOwnerAlias": "amazon",
                    "CreationDate": "2023-01-25T17:20:40Z",
                    "DeprecationTime": "2025-01-25T17:20:40Z",
                    "IsPublic": true
                }
            },
            {
                "InstanceId": "i-0987654321EXAMPLE",
                "InstanceType": "t2.micro",
                "LaunchTime": "2024-08-28T11:25:45+00:00",
                "AvailabilityZone": "us-east-1a",
                "State": {
                    "Code": 16,
                    "Name": "running"
                },
                "OwnerId": "123412341234",
                "Tags": [
                    {
                        "Key": "MyTagName",
                        "Value": "my-tag-value"
                    }
                ],
                "ImageMetadata": {
                    "ImageId": "ami-0b752bf1df193a6c4",
                    "Name": "al2023-ami-2023.5.20240819.0-kernel-6.1-x86_64",
                    "OwnerId": "137112412989",
                    "State": "available",
                    "ImageOwnerAlias": "amazon",
                    "CreationDate": "2023-01-25T17:20:40Z",
                    "DeprecationTime": "2025-01-25T17:20:40Z",
                    "IsPublic": true
                }
```

```
        }
    ],
    "NextToken": "...EXAMPLEV7ixRYHwIAABAA2JHaFxLnDAzpatfEXAMPLE..."
}
```

자세한 내용은 Amazon EC2 사용 설명서에서 [Amazon EC2의 Amazon Machine Image(AMI)](https://docs.aws.amazon.com/)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeInstanceImageMetadata](https://docs.aws.amazon.com/) 섹션을 참조하세요.

`describe-instance-status`

다음 코드 예시는 `describe-instance-status`의 사용 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 설명서에서 인스턴스 상태 모니터링을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조에서 DescribeInstanceStatus를 참조하세요.

describe-instance-topology

다음 코드 예시는 `describe-instance-topology`의 사용 방법을 보여 줍니다.

AWS CLI

모든 인스턴스의 인스턴스 토폴로지 설명

다음 `describe-instance-topology` 예시에서는 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"
}
```

더 많은 예시를 포함한 자세한 내용은 Amazon EC2 사용 설명서의 Amazon EC2 인스턴스 토폴로지를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeInstanceTopology 섹션을 참조하세요.

describe-instance-type-offerings

다음 코드 예시는 describe-instance-type-offerings의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 리전에서 제공되는 인스턴스 유형 나열

다음 describe-instance-type-offerings 제에서는 AWS CLI의 기본 리전으로 구성된 리전에서 제공되는 인스턴스 유형을 나열합니다.

```
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"
      },
      ...
  ]
}
```

예시 2: 가용 영역에서 제공되는 인스턴스 유형 나열

다음 describe-instance-type-offerings 예시에서는 지정된 가용 영역에서 제공되는 인스턴스 유형을 나열합니다. 가용 영역은 지정된 리전 내에 있어야 합니다.

```
aws ec2 describe-instance-type-offerings \
    --location-type availability-zone \
    --filters Name=location,Values=us-east-2a \
    --region us-east-2
```

예시 3: 인스턴스 유형이 지원되는지 확인

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 예시에서는 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"
]
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeInstanceTypeOfferings 섹션을 참조하세요.

describe-instance-types

다음 코드 예시는 describe-instance-types의 사용 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 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 Amazon Elastic Compute Cloud 사용 설명서의 인스턴스 유형을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeInstanceTypes를 참조하세요.

describe-instances

다음 코드 예시는 describe-instances의 사용 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": "111111111111111111111"
                    },
                    "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 설명서에서 CLI를 사용하여 나열 및 필터링을 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 유형과 가용 영역의 인스턴스를 필터링하는 방법

다음 `describe-instances` 예제에서는 JSON 입력 파일을 사용하여 이전 예제와 동일한 필터링을 수행합니다. 필터가 복잡해지면 JSON 파일에서 필터를 더 쉽게 지정할 수 있습니다.

```
aws ec2 describe-instances \
    --filters file://filters.json
```

`filters.json`의 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 인스턴스의 인스턴스 및 서브넷 ID만 표시하는 방법

다음 describe-instances 예제에서는 --query 파라미터를 사용하여 모든 인스턴스의 인스턴스 및 서브넷 ID만 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 인스턴스 ID만 표시하는 방법

다음 `describe-instances` 예제에서는 필터를 사용하여 결과 범위를 지정된 유형의 인스턴스로 지정하고 `--query` 파라미터를 사용하여 인스턴스 ID만 표시합니다.

```
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 인스턴스 ID, 가용 영역, 지정된 태그 값만 표시하는 방법

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 Linux 인스턴스용 Amazon EC2 사용 설명서에서 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 Linux 인스턴스용 Amazon EC2 사용 설명서에서 [배치 그룹의 인스턴스 설명](https://)을 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 설명서에서 인스턴스 메타데이터의 인스턴스 태그 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeInstances를 참조하세요.

describe-internet-gateways

다음 코드 예시는 describe-internet-gateways의 사용 방법을 보여 줍니다.

AWS CLI

인터넷 게이트웨이 설명

다음 describe-internet-gateways 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeInternetGateways 섹션을 참조하세요.

describe-ipam-pools

다음 코드 예시는 describe-ipam-pools의 사용 방법을 보여 줍니다.

AWS CLI

IPAM 풀에 대한 세부 정보 보기

다음 `describe-ipam-pools` 예시에서는 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"
                }
            ]
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeIpamPools 섹션을 참조하세요.

`describe-ipam-resource-discoveries`

다음 코드 예시는 `describe-ipam-resource-discoveries`의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 리소스 검색의 전체 세부 정보 보기

이 예시에서는 관리자가 내 조직에 있는 리소스의 IP 주소를 관리하고 모니터링할 수 있도록 리소스 검색을 만들어 다른 AWS Organization의 IPAM 관리자와 공유하려는 위임받은 IPAM 관리자의 경우를 가정합니다.

이 예시는 다음과 같은 경우에 유용할 수 있습니다.

리소스 검색을 만들려고 했지만 한도인 1개에 도달했다는 오류가 발생한 경우 리소스 검색을 이미 만들었고 계정에서 이를 확인하려는 경우. 리전에 IPAM에서 검색되지 않는 리소스가 있는 경우. 리소스에 대해 정의된 `--operating-regions`를 확인하고 리소스를 검색할 수 있도록 올바른 리전을 운영 리전으로 추가했는지 확인하려고 합니다.

다음 `describe-ipam-resource-discoveries` 예시에서는 AWS 계정의 리소스 검색에 대한 세부 정보를 나열합니다. AWS 리전당 리소스 검색을 한 번만 할 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 IPAM을 조직 외부 계정과 통합 섹션을 참조하세요.

예시 2: 리소스 검색 ID만 보기

다음 `describe-ipam-resource-discoveries` 예시에서는 AWS 계정의 리소스 검색에 대한 ID를 나열합니다. AWS 리전당 리소스 검색을 한 번만 할 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 IPAM을 조직 외부 계정과 통합 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeIpamResourceDiscoveries 섹션을 참조하세요.

describe-ipam-resource-discovery-associations

다음 코드 예시는 describe-ipam-resource-discovery-associations의 사용 방법을 보여 줍니다.

AWS CLI

IPAM과의 모든 리소스 검색 연결 보기

이 예시에서는 리소스 검색을 IPAM과 연결하여 다른 계정을 IPAM과 통합하는 IPAM 위임된 관리자입니다. IPAM이 리소스 검색의 운영 리전에서 리소스를 예상대로 검색하지 못하는 것을 발견했습니다. 리소스 검색의 상태 및 상태를 확인하여 리소스 검색을 생성한 계정이 여전히 활성 상태이고 리소스 검색이 계속 공유되고 있는지 확인하려고 합니다.

--region은 IPAM의 홈 리전이어야 합니다.

다음 describe-ipam-resource-discovery-associations 예시에서는 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 예시에서는 이 명령을 실행한 후 기본 리소스 검색이 아닌 리소스 검색(`"IsDefault": false ``) that is ``"ResourceDiscoveryStatus": "not-found"` 및 `"State": "create-complete"`)이 하나씩 있음을 알 수 있습니다. 리소스 검색 소유자의 계정이 종료되었습니다. 다른 경우에 `"ResourceDiscoveryStatus": "not-found"`와 `"State": "associate-complete"`가 표시되면 다음 중 하나가 발생했음을 나타냅니다.

리소스 검색 소유자가 리소스 검색을 삭제했습니다. 리소스 검색 소유자가 리소스 검색을 공유 해제했습니다.

자세한 내용은 Amazon VPC IPAM 사용 설명서의 IPAM을 조직 외부 계정과 통합 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeIpamResourceDiscoveryAssociations 섹션을 참조하세요.

describe-ipam-scopes

다음 코드 예시는 `describe-ipam-scopes`의 사용 방법을 보여 줍니다.

AWS CLI

IPAM 범위에 대한 세부 정보 보기

다음 `describe-ipam-scopes` 예시에서는 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"
                    }
                ]
            }
        ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeIpamScopes 섹션을 참조하세요.

describe-ipams

다음 코드 예시는 describe-ipams의 사용 방법을 보여 줍니다.

AWS CLI

IPAM에 대한 세부 정보 보기

다음 describe-ipams 예시에서는 IPAM의 세부 정보를 보여줍니다.

```
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"
                }
            ]
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeIpams 섹션을 참조하세요.

describe-ipv6-pools

다음 코드 예시는 describe-ipv6-pools의 사용 방법을 보여 줍니다.

AWS CLI

IPv6 주소 풀 설명

다음 describe-ipv6-pools 예시에서는 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 세부 정보는 AWS CLI 명령 참조의 DescribeIpv6Pools 섹션을 참조하세요.

describe-key-pairs

다음 코드 예시는 describe-key-pairs의 사용 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 설명서에서 퍼블릭 키 설명을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeKeyPairs를 참조하세요.

describe-launch-template-versions

다음 코드 예시는 describe-launch-template-versions의 사용 방법을 보여 줍니다.

AWS CLI

시작 템플릿 버전 설명

이 예시에서는 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeLaunchTemplateVersions 섹션을 참조하세요.

describe-launch-templates

다음 코드 예시는 describe-launch-templates의 사용 방법을 보여 줍니다.

AWS CLI

시작 템플릿 설명

이 예시에서는 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"
        },
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeLaunchTemplates 섹션을 참조하세요.

describe-local-gateway-route-table-virtual-interface-group-associations

다음 코드 예시는 describe-local-gateway-route-table-virtual-interface-group-associations의 사용 방법을 보여 줍니다.

AWS CLI

가상 인터페이스 그룹과 로컬 게이트웨이 라우팅 테이블 간의 연결 설명

다음 describe-local-gateway-route-table-virtual-interface-group-associations 예시에서는 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 AWS Outposts 사용 설명서의 로컬 게이트웨이로 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeLocalGatewayRouteTableVirtualInterfaceGroupAssociations 섹션을 참조하세요.

`describe-local-gateway-route-table-vpc-associations`

다음 코드 예시는 `describe-local-gateway-route-table-vpc-associations`의 사용 방법을 보여 줍니다.

AWS CLI

VPC와 로컬 게이트웨이 라우팅 테이블 간의 연결 설명

다음 `describe-local-gateway-route-table-vpc-associations` 예시에서는 VPC와 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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeLocalGatewayRouteTableVpcAssociations 섹션을 참조하세요.

`describe-local-gateway-route-tables`

다음 코드 예시는 `describe-local-gateway-route-tables`의 사용 방법을 보여 줍니다.

AWS CLI

로컬 게이트웨이 라우팅 테이블 설명

다음 `describe-local-gateway-route-tables` 예시에서는 로컬 게이트웨이 라우팅 테이블에 대한 세부 정보를 표시합니다.

```
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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeLocalGatewayRouteTables 섹션을 참조하세요.

`describe-local-gateway-virtual-interface-groups`

다음 코드 예시는 `describe-local-gateway-virtual-interface-groups`의 사용 방법을 보여 줍니다.

AWS CLI

로컬 게이트웨이 가상 인터페이스 그룹 설명

다음 describe-local-gateway-virtual-interface-groups 예시에서는 AWS 계정의 로컬 게이트웨이 가상 인터페이스 그룹을 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 AWS Outposts 사용 설명서의 로컬 게이트웨이로 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeLocalGatewayVirtualInterfaceGroups 섹션을 참조하세요.

describe-local-gateway-virtual-interfaces

다음 코드 예시는 describe-local-gateway-virtual-interfaces의 사용 방법을 보여 줍니다.

AWS CLI

로컬 게이트웨이 가상 인터페이스 설명

다음 describe-local-gateway-virtual-interfaces 예시에서는 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 AWS Outposts 사용 설명서의 로컬 게이트웨이로 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeLocalGatewayVirtualInterfaces 섹션을 참조하세요.

describe-local-gateways

다음 코드 예시는 describe-local-gateways의 사용 방법을 보여 줍니다.

AWS CLI

로컬 게이트웨이 설명

다음 describe-local-gateways 예시에서는 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeLocalGateways 섹션을 참조하세요.

describe-locked-snapshots

다음 코드 예시는 describe-locked-snapshots의 사용 방법을 보여 줍니다.

AWS CLI

스냅샷의 잠금 상태 설명

다음 describe-locked-snapshots 예시에서는 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 Snapshot Lock을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeLockedSnapshots 섹션을 참조하세요.

`describe-managed-prefix-lists`

다음 코드 예시는 `describe-managed-prefix-lists`의 사용 방법을 보여 줍니다.

AWS CLI

관리형 접두사 목록 설명

다음 `describe-managed-prefix-lists` 예시에서는 AWS 계정 `123456789012`가 소유한 접두사 목록을 설명합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeManagedPrefixLists 섹션을 참조하세요.

describe-moving-addresses

다음 코드 예시는 describe-moving-addresses의 사용 방법을 보여 줍니다.

AWS CLI

이전 주소 설명

이 예시에서는 이동하는 모든 탄력적 IP 주소를 설명합니다.

명령:

```
aws ec2 describe-moving-addresses
```

출력:

```
{
  "MovingAddressStatuses": [
    {
```

```
        "PublicIp": "198.51.100.0",
        "MoveStatus": "MovingToVpc"
      }
    ]
}
```

이 예시에서는 EC2-VPC 플랫폼으로 이동하는 모든 주소를 설명합니다.

명령:

```
aws ec2 describe-moving-addresses --filters Name=moving-status,Values=MovingToVpc
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeMovingAddresses 섹션을 참조하세요.

describe-nat-gateways

다음 코드 예시는 describe-nat-gateways의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 퍼블릭 NAT 게이트웨이 설명

다음 describe-nat-gateways 예시에서는 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"
        }
    ]
}
```

예시 2: 프라이빗 NAT 게이트웨이 설명

다음 `describe-nat-gateways` 예시에서는 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 정보는 Amazon VPC 사용 설명서의 NAT 게이트웨이 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeNatGateways을 참조하세요.

describe-network-acls

다음 코드 예시는 describe-network-acls의 사용 방법을 보여 줍니다.

AWS CLI

네트워크 ACL 설명

다음 describe-network-acls 예시에서는 네트워크 ACL에 대한 세부 정보를 검색합니다.

```
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 네트워크 ACL을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeNetworkAcls을 참조하세요.

describe-network-insights-access-scope-analyses

다음 코드 예시는 describe-network-insights-access-scope-analyses의 사용 방법을 보여줍니다.

AWS CLI

Network Insights 액세스 범위 분석 설명

다음 describe-network-insights-access-scope-analyses 예시에서는 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 설명서의 [AWS CLI를 사용하여 Network Access Analyzer 시작하기](https://docs.aws.amazon.com/vpc/latest/network-access-analyzer/getting-started-cli-naa.html)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeNetworkInsightsAccessScopeAnalyses](https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ec2/describe-network-insights-access-scope-analyses.html) 섹션을 참조하세요.

describe-network-insights-access-scopes

다음 코드 예시는 describe-network-insights-access-scopes의 사용 방법을 보여 줍니다.

AWS CLI

Network Insights 액세스 범위 설명

다음 `describe-network-insights-access-scopes` 예시에서는 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 설명서의 AWS CLI를 사용하여 Network Access Analyzer 시작하기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeNetworkInsightsAccessScopes 섹션을 참조하세요.

describe-network-insights-analyses

다음 코드 예시는 `describe-network-insights-analyses`의 사용 방법을 보여 줍니다.

AWS CLI

경로 분석 결과를 보는 방법

다음 `describe-network-insights-analyses` 예시에서는 지정된 분석을 설명합니다. 이 예시에서 소스는 인터넷 게이트웨이, 대상은 EC2 인스턴스, 프로토콜은 TCP입니다. 분석에 성공했고(`Status`는 `succeeded`) 경로에 연결할 수 없습니다(`NetworkPathFound`는 `false`). 설명 코드 `ENI_SG_RULES_MISMATCH`는 인스턴스에 대한 보안 그룹에 대상 포트의 트래픽을 허용하는 규칙이 포함되어 있지 않음을 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 설명서의 AWS CLI를 사용하여 시작하기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeNetworkInsightsAnalyses 섹션을 참조하세요.

describe-network-insights-paths

다음 코드 예시는 describe-network-insights-paths의 사용 방법을 보여 줍니다.

AWS CLI

경로 설명

다음 describe-network-insights-paths 예시에서는 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 설명서의 AWS CLI를 사용하여 시작하기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeNetworkInsightsPaths 섹션을 참조하세요.

describe-network-interface-attribute

다음 코드 예시는 describe-network-interface-attribute의 사용 방법을 보여 줍니다.

AWS CLI

네트워크 인터페이스의 연결 속성 설명

이 예시 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 속성 설명

이 예시 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 속성 설명

이 예시 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 속성 설명

이 예시 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
```

```
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeNetworkInterfaceAttribute](#) 섹션을 참조하세요.

describe-network-interface-permissions

다음 코드 예시는 describe-network-interface-permissions의 사용 방법을 보여 줍니다.

AWS CLI

네트워크 인터페이스 권한 설명

이 예시에서는 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"
      }
  ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeNetworkInterfacePermissions](#) 섹션을 참조하세요.

describe-network-interfaces

다음 코드 예시는 describe-network-interfaces의 사용 방법을 보여 줍니다.

AWS CLI

네트워크 인터페이스 설명

이 예시에서는 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 예시에서는 `Purpose` 키와 `Prod` 값을 가진 태그가 있는 네트워크 인터페이스에 대해 설명합니다.

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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeNetworkInterfaces 섹션을 참조하세요.

describe-placement-groups

다음 코드 예시는 describe-placement-groups의 사용 방법을 보여 줍니다.

AWS CLI

모든 배치 그룹 설명

이 예시 명령은 모든 배치 그룹을 설명합니다.

명령:

```
aws ec2 describe-placement-groups
```

출력:

```
{
    "PlacementGroups": [
        {
            "GroupName": "my-cluster",
            "State": "available",
            "Strategy": "cluster"
        },
        ...
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribePlacementGroups 섹션을 참조하세요.

describe-prefix-lists

다음 코드 예시는 describe-prefix-lists의 사용 방법을 보여 줍니다.

AWS CLI

접두사 목록 설명

이 예시에서는 리전에서 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"
      }
    ]
  }
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribePrefixLists 섹션을 참조하세요.

describe-principal-id-format

다음 코드 예시는 `describe-principal-id-format`의 사용 방법을 보여 줍니다.

AWS CLI

긴 ID 형식이 활성화된 IAM 사용자 및 역할의 ID 형식 설명

다음 `describe-principal-id-format` 예시에서는 루트 사용자, 모든 IAM 역할 및 긴 ID 형식을 사용하도록 설정한 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
                },
            ]
        },
        ...
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribePrincipalIdFormat 섹션을 참조하세요.

describe-public-ipv4-pools

다음 코드 예시는 `describe-public-ipv4-pools`의 사용 방법을 보여 줍니다.

AWS CLI

퍼블릭 IPv4 주소 풀 설명

다음 `describe-public-ipv4-pools` 예시에서는 기존 보유 IP 주소 사용(BYOIP)을 사용하여 퍼블릭 IPv4 주소 범위를 프로비저닝할 때 생성한 주소 풀에 대한 세부 정보를 표시합니다.

```
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 세부 정보는 AWS CLI 명령 참조의 DescribePublicIpv4Pools 섹션을 참조하세요.

describe-regions

다음 코드 예시는 describe-regions의 사용 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 설명서에서 리전 및 가용 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 설명서에서 [리전 및 가용 영역]을 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 설명서에서 리전 및 가용 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 설명서에서 리전 및 가용 영역을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조에서 DescribeRegions를 참조하세요.

describe-replace-root-volume-tasks

다음 코드 예시는 describe-replace-root-volume-tasks의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 특정 루트 볼륨 교체 작업에 대한 정보를 보는 방법

다음 describe-replace-root-volume-tasks 예시에서는 루트 볼륨 교체 태스크 eplacevol-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 교체]를 참조하세요.

예시 2: 특정 인스턴스의 모든 루트 볼륨 교체 작업에 대한 정보를 보는 방법

다음 `describe-replace-root-volume-tasks` 예시에서는 i-0123456789abcdefa에 대한 모든 루트 볼륨 교체 태스크를 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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeReplaceRootVolumeTasks 섹션을 참조하세요.

describe-reserved-instances-listings

다음 코드 예시는 `describe-reserved-instances-listings`의 사용 방법을 보여 줍니다.

AWS CLI

예약 인스턴스 목록 설명

다음 `describe-reserved-instances-listings` 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeReservedInstancesListings 섹션을 참조하세요.

describe-reserved-instances-modifications

다음 코드 예시는 `describe-reserved-instances-modifications`의 사용 방법을 보여 줍니다.

AWS CLI

예약 인스턴스 수정 설명

이 예시 명령은 계정에 대해 제출된 모든 예약된 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"
                }
            ]
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeReservedInstancesModifications 섹션을 참조하세요.

describe-reserved-instances-offerings

다음 코드 예시는 describe-reserved-instances-offerings의 사용 방법을 보여 줍니다.

AWS CLI

예약 인스턴스 오퍼링 설명

이 예시 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 제공 설명

이 예시에서는 t1.micro 인스턴스 유형, Windows(Amazon VPC) 제품, 고활용도 오퍼링 사양으로 AWS에서 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeReservedInstancesOfferings 섹션을 참조하세요.

`describe-reserved-instances`

다음 코드 예시는 `describe-reserved-instances`의 사용 방법을 보여 줍니다.

AWS CLI

예약 인스턴스 설명

이 예시 명령은 소유하고 있는 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 인스턴스 설명

이 예시에서는 응답을 필터링하여 us-west-1c에 3년짜리 t2.micro Linux/UNIX 예약 인스턴스만 포함하도록 합니다.

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 Command Line Interface 사용 설명서에서 Amazon EC2 인스턴스 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeReservedInstances 섹션을 참조하세요.

describe-route-tables

다음 코드 예시는 describe-route-tables의 사용 방법을 보여 줍니다.

AWS CLI

라우팅 테이블 설명

다음 describe-route-tables 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeRouteTables 섹션을 참조하세요.

describe-scheduled-instance-availability

다음 코드 예시는 describe-scheduled-instance-availability의 사용 방법을 보여 줍니다.

AWS CLI

사용 가능한 일정 설명

이 예시에서는 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

- API 세부 정보는 AWS CLI 명령 참조의 DescribeScheduledInstanceAvailability 섹션을 참조하세요.

describe-scheduled-instances

다음 코드 예시는 describe-scheduled-instances의 사용 방법을 보여 줍니다.

AWS CLI

정기 인스턴스 설명

이 예시에서는 지정된 정기 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 예시에서는 모든 정기 인스턴스를 설명합니다.

명령:

```
aws ec2 describe-scheduled-instances
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeScheduledInstances 섹션을 참조하세요.

describe-security-group-references

다음 코드 예시는 describe-security-group-references의 사용 방법을 보여 줍니다.

AWS CLI

보안 그룹 참조 설명

이 예시에서는 sg-bbbb2222의 보안 그룹 참조를 설명합니다. 응답은 보안 그룹 sg-bbbb2222가 VPC 의 보안 그룹 vpc-aaaaaaaa에서 참조되고 있음을 나타냅니다.

명령:

```
aws ec2 describe-security-group-references --group-id sg-bbbbb22222
```

출력:

```
{
  "SecurityGroupsReferenceSet": [
    {
      "ReferencingVpcId": "vpc-aaaaaaaa ",
      "GroupId": "sg-bbbbb22222",
      "VpcPeeringConnectionId": "pcx-b04deed9"
    }
  ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeSecurityGroupReferences 섹션을 참조하세요.

describe-security-group-rules

다음 코드 예시는 describe-security-group-rules의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 보안 그룹에 대한 보안 그룹 규칙 설명

다음 describe-security-group-rules 예시에서는 지정된 보안 그룹의 보안 그룹 규칙을 설명합니다. filters 옵션을 사용하여 특정 보안 그룹으로 결과 범위를 지정할 수 있습니다.

```
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": []
        }
```

```
    ]
}
```

예시 2: 보안 그룹 규칙 설명

다음 describe-security-group-rules 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeSecurityGroupRules 섹션을 참조하세요.

describe-security-group-vpc-associations

다음 코드 예시는 describe-security-group-vpc-associations의 사용 방법을 보여 줍니다.

AWS CLI

VPC 연결 설명

다음 describe-security-group-vpc-associations 예제에서는 지정된 보안 그룹의 VPC 연결을 설명합니다.

```
aws ec2 describe-security-group-vpc-associations \
    --filters Name=group-id,Values=sg-04dbb43907d3f8a78
```

출력:

```
{
    "SecurityGroupVpcAssociations": [
        {
            "GroupId": "sg-04dbb43907d3f8a78",
            "VpcId": "vpc-0bf4c2739bc05a694",
            "VpcOwnerId": "123456789012",
            "State": "associated"
        }
    ]
}
```

자세한 내용은 Amazon VPC 사용 설명서의 보안 그룹을 여러 VPC와 연결을 참조하시기 바랍니다.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeSecurityGroupVpcAssociations 섹션을 참조하시기 바랍니다.

`describe-security-groups`

다음 코드 예시는 `describe-security-groups`의 사용 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` 예시에서는 필터를 사용하여 SSH 트래픽을 허용하는 규칙(포트 22)과 모든 주소(`0.0.0.0/0`)의 트래픽을 허용하는 규칙이 있는 보안 그룹으로 결과 범위를 지정합니다. 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 포함된 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 보안 그룹의 이름 및 ID만 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 설명서에서 [태그 작업]을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeSecurityGroups]를 참조하세요.

describe-snapshot-attribute

다음 코드 예시는 `describe-snapshot-attribute`의 사용 방법을 보여 줍니다.

AWS CLI

스냅샷의 스냅샷 속성 설명

다음 `describe-snapshot-attribute` 예시에서는 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 공유]를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeSnapshotAttribute] 섹션을 참조하세요.

describe-snapshot-tier-status

다음 코드 예시는 describe-snapshot-tier-status의 사용 방법을 보여 줍니다.

AWS CLI

아카이빙된 스냅샷에 대한 아카이브 정보 보기

다음 describe-snapshot-tier-status 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeSnapshotTierStatus 섹션을 참조하세요.

describe-snapshots

다음 코드 예시는 describe-snapshots의 사용 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 EC2 사용 설명서에서 Amazon EBS 스냅샷을 참조하세요.

예제 2: 필터를 기반으로 스냅샷을 설명하는 방법

다음 `describe-snapshots` 예제에서는 필터를 사용하여 결과 범위를 AWS 계정에서 소유한 `pending` 상태의 스냅샷으로 지정합니다. 이 예제에서는 `--query` 파라미터를 사용하여 스냅샷 ID 및 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 ID만 표시합니다.

```
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 설명서에서 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 설명서에서 태그 작업을 참조하세요.

예제 4: 수명에 기반하여 스냅샷을 설명하는 방법

다음 describe-snapshots 예제에서는 JMesPath 표현식을 사용하여 지정된 날짜 이전에 AWS 계정에서 생성한 모든 스냅샷을 설명합니다. 스냅샷 ID만 표시합니다.

```
aws ec2 describe-snapshots \
    --owner-ids 012345678910 \
    --query "Snapshots[?(StartTime<='2020-03-31')].[SnapshotId]"
```

필터를 사용하는 추가 예제는 Amazon EC2 사용 설명서에서 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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeSnapshots를 참조하세요.

describe-spot-datafeed-subscription

다음 코드 예시는 describe-spot-datafeed-subscription의 사용 방법을 보여 줍니다.

AWS CLI

계정에 대한 스팟 인스턴스 데이터 피드 구독 설명

이 예시 명령은 계정에 대한 데이터 피드를 설명합니다.

명령:

```
aws ec2 describe-spot-datafeed-subscription
```

출력:

```
{
    "SpotDatafeedSubscription": {
        "OwnerId": "123456789012",
        "Prefix": "spotdata",
        "Bucket": "amzn-s3-demo-bucket",
        "State": "Active"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeSpotDatafeedSubscription 섹션을 참조하세요.

describe-spot-fleet-instances

다음 코드 예시는 describe-spot-fleet-instances의 사용 방법을 보여 줍니다.

AWS CLI

스팟 플릿과 연결된 스팟 인스턴스 설명

이 예시 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeSpotFleetInstances 섹션을 참조하세요.

describe-spot-fleet-request-history

다음 코드 예시는 describe-spot-fleet-request-history의 사용 방법을 보여 줍니다.

AWS CLI

스팟 플릿 기록 설명

이 예시 명령은 지정된 시간부터 시작하여 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 예시에서는 스팟 플릿에 대한 스팟 인스턴스 두 개를 성공적으로 실행한 결과를 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeSpotFleetRequestHistory 섹션을 참조하세요.

`describe-spot-fleet-requests`

다음 코드 예시는 `describe-spot-fleet-requests`의 사용 방법을 보여 줍니다.

AWS CLI

스팟 플릿 요청 설명

이 예시에서는 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 요청 설명

이 예시에서는 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"
    }
  ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeSpotFleetRequests 섹션을 참조하세요.

describe-spot-instance-requests

다음 코드 예시는 describe-spot-instance-requests의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 스팟 인스턴스 요청 설명

다음 describe-spot-instance-requests 예시에서는 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"
        }
    ]
}
```

예시 2: 필터를 기반으로 스팟 인스턴스 요청 설명

다음 `describe-spot-instance-requests` 예시에서는 필터를 사용하여 지정된 가용 영역에서 지정된 인스턴스 유형을 가진 스팟 인스턴스 요청으로 결과 범위를 지정합니다. 이 예시에서는 `--query` 파라미터를 사용하여 인스턴스 ID만 표시합니다.

```
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 예시에서는 Amazon Elastic Compute Cloud 사용 설명서의 [리소스 나열 및 필터링]을 참조하세요.

예시 3: 태그를 기반으로 스팟 인스턴스 요청 설명

다음 `describe-spot-instance-requests` 예시에서는 태그 필터를 사용하여 `cost-center=cc123` 태그가 있는 스팟 인스턴스 요청으로 결과 범위를 지정합니다.

```
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 설명서에서 [태그 작업]을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeSpotInstanceRequests] 섹션을 참조하세요.

`describe-spot-price-history`

다음 코드 예시는 `describe-spot-price-history`의 사용 방법을 보여 줍니다.

AWS CLI

스팟 가격 내역 설명

이 예시 명령은 1월의 특정 날짜에 대한 m1.xlarge 인스턴스의 스팟 가격 내역을 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/UNIX Amazon VPC의 스팟 가격 기록 설명

이 예시 명령은 1월의 특정 날짜에 대한 m1.xlarge, Linux/UNIX Amazon VPC 인스턴스의 스팟 가격 내역을 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"
      }
  ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeSpotPriceHistory 섹션을 참조하세요.

describe-stale-security-groups

다음 코드 예시는 describe-stale-security-groups의 사용 방법을 보여 줍니다.

AWS CLI

상태 보안 그룹 설명

이 예시에서는 vpc-11223344의 오래된 보안 그룹 규칙을 설명합니다. 이 응답은 계정의 sg-5fa68d3a에 피어 VPC의 sg-279ab042를 참조하는 오래된 수신 SSH 규칙이 있고 계정의 sg-fe6fba9a에 피어 VPC의 sg-ef6fba8b를 참조하는 오래된 송신 SSH 규칙이 있음을 보여줍니다.

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"
    }
  ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeStaleSecurityGroups](#) 섹션을 참조하세요.

describe-store-image-tasks

다음 코드 예시는 `describe-store-image-tasks`의 사용 방법을 보여 줍니다.

AWS CLI

AMI 저장 태스크의 진행률 설명

다음 `describe-store-image-tasks` 예시에서는 AMI 저장 태스크의 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 사용하여 AMI를 저장하고 복원하는 방법에 대한 자세한 내용은 Amazon EC2 사용 설명서의 S3를 사용하여 AMI 저장 및 복원<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide ami-store-restore.html>을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeStoreImageTasks 섹션을 참조하세요.

describe-subnets

다음 코드 예시는 describe-subnets의 사용 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 설명서에서 VPC 및 서브넷 작업을 참조하세요.

예제 2: 특정 VPC의 서브넷을 설명하는 방법

다음 `describe-subnets` 예제에서는 필터를 사용하여 지정된 VPC의 서브넷에 대한 세부 정보를 검색합니다.

```
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 설명서에서 VPC 및 서브넷 작업을 참조하세요.

예제 3: 특정 태그의 서브넷을 설명하는 방법

다음 describe-subnets 예제에서는 필터를 사용하여 CostCenter=123 태그가 있는 해당 서브넷 세부 정보를 검색하고 --query 파라미터를 사용하여 이 태그가 있는 서브넷의 서브넷 ID를 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 설명서에서 VPC 및 서브넷 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조에서 DescribeSubnets를 참조하세요.

describe-tags

다음 코드 예시는 describe-tags의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 단일 리소스에 대한 모든 태그 설명

다음 describe-tags 예시에서는 지정된 인스턴스에 대한 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"
        }
    ]
}
```

예시 2: 리소스 유형에 대한 모든 태그 설명

다음 `describe-tags` 예시에서는 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"
        }
    ]
}
```

예시 3: 모든 태그 설명

다음 `describe-tags` 예시에서는 모든 리소스에 대한 태그를 설명합니다.

```
aws ec2 describe-tags
```

예시 4: 태그 키를 기반으로 리소스의 태그 설명

다음 describe-tags 예시에서는 키가 Stack로 시작하는 태그가 있는 리소스에 대한 태그를 설명합니다.

```
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"
        }
    ]
}
```

예시 5: 태그 키 및 태그 값을 기반으로 리소스의 태그 설명

다음 describe-tags 예시에서는 Stack=Test 태그가 있는 리소스의 태그를 설명합니다.

```
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 예시에서는 대체 구문을 사용하여 Stack=Test 태그가 있는 리소스를 설명합니다.

```
aws ec2 describe-tags \
    --filters "Name=tag:Stack,Values=Test"
```

다음 describe-tags 예시에서는 Purpose 키가 있고 값이 없는 태그가 있는 모든 인스턴스에 대한 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeTags 섹션을 참조하세요.

describe-traffic-mirror-filters

다음 코드 예시는 describe-traffic-mirror-filters의 사용 방법을 보여 줍니다.

AWS CLI

트래픽 미러 필터를 보는 방법

다음 describe-traffic-mirror-filters 예시에서는 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 Traffic Mirroring 설명서의 트래픽 미러 필터 보기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeTrafficMirrorFilters 섹션을 참조하세요.

describe-traffic-mirror-sessions

다음 코드 예시는 describe-traffic-mirror-sessions의 사용 방법을 보여 줍니다.

AWS CLI

트래픽 미러 세션 설명

다음 `describe-traffic-mirror-sessions` 예시에서는 트래픽 미러 세션의 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 AWS Traffic Mirroring 설명서의 트래픽 미러 세션 세부 정보 보기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeTrafficMirrorSessions 섹션을 참조하세요.

describe-traffic-mirror-targets

다음 코드 예시는 `describe-traffic-mirror-targets`의 사용 방법을 보여 줍니다.

AWS CLI

트래픽 미러 대상 설명

다음 `describe-traffic-mirror-targets` 예시에서는 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 VPC Traffic Mirroring 설명서의 트래픽 미러 대상을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeTrafficMirrorTargets 섹션을 참조하세요.

describe-transit-gateway-attachments

다음 코드 예시는 `describe-transit-gateway-attachments`의 사용 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 피어링 연결 보기

다음 `describe-transit-gateway-attachments` 예시에서는 Transit Gateway Attachment에 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 설명서의 전송 게이트웨이 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeTransitGatewayAttachments 섹션을 참조하세요.

describe-transit-gateway-connect-peers

다음 코드 예시는 describe-transit-gateway-connect-peers의 사용 방법을 보여 줍니다.

AWS CLI

Transit Gateway Connect 피어 설명

다음 describe-transit-gateway-connect-peers 예시에서는 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 설명서의 [전송 게이트웨이 Connect 연결 및 Transit Gateway Connect 피어](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeTransitGatewayConnectPeers](#) 섹션을 참조하세요.

describe-transit-gateway-connects

다음 코드 예시는 `describe-transit-gateway-connects`의 사용 방법을 보여 줍니다.

AWS CLI

Transit Gateway Connect 연결 설명

다음 `describe-transit-gateway-connects` 예시에서는 지정된 Connect 연결을 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 설명서의 [전송 게이트웨이 Connect 연결 및 Transit Gateway Connect 피어](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeTransitGatewayConnects](#) 섹션을 참조하세요.

describe-transit-gateway-multicast-domains

다음 코드 예시는 `describe-transit-gateway-multicast-domains`의 사용 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 멀티캐스트 도메인 설명

다음 `describe-transit-gateway-multicast-domains` 예시에서는 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 설명서의 Managing multicast domains 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeTransitGatewayMulticastDomains 섹션을 참조하세요.

describe-transit-gateway-peering-attachments

다음 코드 예시는 describe-transit-gateway-peering-attachments의 사용 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 피어링 연결 설명

다음 describe-transit-gateway-peering-attachments 예시에서는 모든 전송 게이트웨이 피어링 어태치먼트에 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 설명서의 [Transit Gateway Peering Attachments](#) 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeTransitGatewayPeeringAttachments](#) 섹션을 참조하세요.

describe-transit-gateway-policy-tables

다음 코드 예시는 describe-transit-gateway-policy-tables의 사용 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 정책 테이블 설명

다음 describe-transit-gateway-policy-tables 예시에서는 지정된 전송 게이트웨이 정책 테이블을 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 전송 게이트웨이 정책 테이블을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeTransitGatewayPolicyTables 섹션을 참조하세요.

describe-transit-gateway-route-tables

다음 코드 예시는 describe-transit-gateway-route-tables의 사용 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 라우팅 테이블 설명

다음 describe-transit-gateway-route-tables 예시에서는 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 설명서의 [전송 게이트웨이 라우팅 테이블 보기](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeTransitGatewayRouteTables](#) 섹션을 참조하세요.

describe-transit-gateway-vpc-attachments

다음 코드 예시는 `describe-transit-gateway-vpc-attachments`의 사용 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 VPC 연결 설명

다음 `describe-transit-gateway-vpc-attachments` 예시에서는 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 Gateway 설명서의 VPC 연결 보기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeTransitGatewayVpcAttachments 섹션을 참조하세요.

`describe-transit-gateways`

다음 코드 예시는 `describe-transit-gateways`의 사용 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 설명

다음 `describe-transit-gateways` 예시에서는 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"
                }
            ]
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeTransitGateways 섹션을 참조하세요.

describe-verified-access-endpoints

다음 코드 예시는 `describe-verified-access-endpoints`의 사용 방법을 보여 줍니다.

AWS CLI

Verified·Access 엔드포인트 삭제

다음 `describe-verified-access-endpoints` 예시에서는 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 AWS Verified Access 사용 설명서의 Verified Access endpoints 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeVerifiedAccessEndpoints 섹션을 참조하세요.

describe-verified-access-groups

다음 코드 예시는 describe-verified-access-groups의 사용 방법을 보여 줍니다.

AWS CLI

Verified·Access 그룹 생성

다음 describe-verified-access-groups 예시에서는 지정된 Verified Access 그룹을 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 AWS Verified Access 사용 설명서의 Verified Access groups 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeVerifiedAccessGroups 섹션을 참조하세요.

describe-verified-access-instance-logging-configurations

다음 코드 예시는 describe-verified-access-instance-logging-configurations의 사용 방법을 보여 줍니다.

AWS CLI

Verified Access 인스턴스의 로깅 구성 설명

다음 describe-verified-access-instance-logging-configurations 예시에서는 지정된 Verified Access 인스턴스에 대한 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 AWS Verified Access 사용 설명서의 Verified Access logs 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeVerifiedAccessInstanceLoggingConfigurations 섹션을 참조하세요.

describe-verified-access-instances

다음 코드 예시는 describe-verified-access-instances의 사용 방법을 보여 줍니다.

AWS CLI

Verified·Access 인스턴스 설명

다음 describe-verified-access-instances 예시에서는 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 AWS Verified Access 사용 설명서의 Verified Access instances 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeVerifiedAccessInstances 섹션을 참조하세요.

describe-verified-access-trust-providers

다음 코드 예시는 describe-verified-access-trust-providers의 사용 방법을 보여 줍니다.

AWS CLI

Verified Access 신뢰 공급자 설명

다음 describe-verified-access-trust-providers 예시에서는 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 AWS Verified Access 사용 설명서의 Trust providers for Verified Access 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeVerifiedAccessTrustProviders 섹션을 참조하세요.

describe-volume-attribute

다음 코드 예시는 describe-volume-attribute의 사용 방법을 보여 줍니다.

AWS CLI

볼륨 속성 설명

이 예시 명령은 ID가 vol-049df61146c4d7901인 볼륨의 autoEnableIo 속성을 설명합니다.

명령:

```
aws ec2 describe-volume-attribute --volume-id vol-049df61146c4d7901 --
attribute autoEnableIO
```

출력:

```
{
    "AutoEnableIO": {
        "Value": false
    },
    "VolumeId": "vol-049df61146c4d7901"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeVolumeAttribute 섹션을 참조하세요.

describe-volume-status

다음 코드 예시는 describe-volume-status의 사용 방법을 보여 줍니다.

AWS CLI

단일 볼륨의 상태 설명

이 예시 명령은 볼륨 vol-1234567890abcdef0의 상태를 설명합니다.

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 상태 설명

이 예시 명령은 손상된 모든 볼륨의 상태를 설명합니다. 이 예시 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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeVolumeStatus 섹션을 참조하세요.

describe-volumes-modifications

다음 코드 예시는 describe-volumes-modifications의 사용 방법을 보여 줍니다.

AWS CLI

볼륨의 수정 상태 설명

다음 describe-volumes-modifications 예시에서는 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
    }
```

```
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeVolumesModifications 섹션을 참조하세요.

describe-volumes

다음 코드 예시는 describe-volumes의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 볼륨 설명

다음 describe-volumes 예시에서는 현재 리전 내 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
        }
    ]
}
```

예시 2: 특정 인스턴스에 연결된 볼륨 설명

다음 describe-volumes 예시에서는 지정된 인스턴스에 연결되어 있고 인스턴스가 종료될 때 삭제되도록 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.

예시 3: 특정 가용 영역에서 사용 가능한 볼륨 설명

다음 describe-volumes 예시에서는 지정된 가용 영역에 있고 상태가 available인 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.

예시 4: 태그를 기반으로 볼륨 설명

다음 describe-volumes 예시에서는 태그 키가 Name이고 값이 Test로 시작하는 모든 볼륨을 설명합니다. 그런 다음 출력은 볼륨의 태그와 ID만 표시하는 쿼리로 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 설명서에서 태그 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeVolumes 섹션을 참조하세요.

`describe-vpc-attribute`

다음 코드 예시는 `describe-vpc-attribute`의 사용 방법을 보여 줍니다.

AWS CLI

enableDnsSupport 속성 설명

이 예시에서는 `enableDnsSupport` 속성을 설명합니다. 이 속성은 VPC에 DNS 확인이 활성화되어 있는지 여부를 나타냅니다. 이 속성이 `true`인 경우 Amazon DNS 서버는 인스턴스의 DNS 호스트 이름을 해당 IP 주소로 확인하지만, 그렇지 않으면 확인하지 않습니다.

명령:

```
aws ec2 describe-vpc-attribute --vpc-id vpc-a01106c2 --attribute enableDnsSupport
```

출력:

```
{
    "VpcId": "vpc-a01106c2",
    "EnableDnsSupport": {
        "Value": true
    }
}
```

enableDnsHostnames 속성 설명

이 예시에서는 `enableDnsHostnames` 속성을 설명합니다. 이 속성은 VPC에서 시작된 인스턴스가 DNS 호스트 이름을 가져오는지 나타냅니다. 이 속성이 `true`인 경우 VPC의 인스턴스가 DNS 호스트 이름을 가져오고, 그렇지 않으면 가져오지 않습니다.

명령:

```
aws ec2 describe-vpc-attribute --vpc-id vpc-a01106c2 --attribute enableDnsHostnames
```

출력:

```
{
    "VpcId": "vpc-a01106c2",
    "EnableDnsHostnames": {
        "Value": true
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeVpcAttribute 섹션을 참조하세요.

`describe-vpc-classic-link-dns-support`

다음 코드 예시는 `describe-vpc-classic-link-dns-support`의 사용 방법을 보여 줍니다.

AWS CLI

VPCs에 대한 ClassicLink DNS 지원 설명

이 예시에서는 모든 VPC의 ClassicLink DNS 지원 상태를 설명합니다.

명령:

```
aws ec2 describe-vpc-classic-link-dns-support
```

출력:

```
{
  "Vpcs": [
    {
      "VpcId": "vpc-88888888",
      "ClassicLinkDnsSupported": true
    },
    {
      "VpcId": "vpc-1a2b3c4d",
      "ClassicLinkDnsSupported": false
    }
  ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeVpcClassicLinkDnsSupport 섹션을 참조하세요.

describe-vpc-classic-link

다음 코드 예시는 describe-vpc-classic-link의 사용 방법을 보여 줍니다.

AWS CLI

VPCs의 ClassicLink 상태 설명

이 예시에서는 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 예시에서는 Classiclink에 대해 사용하도록 설정된 VPC만 나열합니다(`is-classic-link-enabled` 필터 값이 `true`로 설정되어 있음).

명령:

```
aws ec2 describe-vpc-classic-link --filter "Name=is-classic-link-enabled,Values=true"
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeVpcClassicLink 섹션을 참조하세요.

describe-vpc-endpoint-associations

다음 코드 예시는 `describe-vpc-endpoint-associations`의 사용 방법을 보여 줍니다.

AWS CLI

VPC 엔드포인트 연결 설명

다음 `describe-vpc-endpoint-associations` 예제에서는 VPC 엔드포인트 연결을 설명합니다.

```
aws ec2 describe-vpc-endpoint-associations
```

출력:

```
{
    "VpcEndpointAssociations": [
```

```
        {
            "Id": "vpce-rsc-asc-0a810ca6ac8866bf9",
            "VpcEndpointId": "vpce-019b90d6f16d4f958",
            "AssociatedResourceAccessibility": "Accessible",
            "DnsEntry": {
                "DnsName":
 "vpce-019b90d6f16d4f958.rcfg-07129f3acded87625.4232ccc.vpc-lattice-rsc.us-
east-2.on.aws",
                "HostedZoneId": "Z03265862FOUNWMZOKUF4"
            },
            "AssociatedResourceArn": "arn:aws:vpc-lattice:us-
east-1:123456789012:resourceconfiguration/rcfg-07129f3acded87625"
        }
    ]
}
```

자세한 내용은 AWS PrivateLink 사용 설명서의 [Manage VPC endpoint associations](https://docs.aws.amazon.com/vpc/latest/privatelink/resource-configuration-associations.html#resource-config-manage-ep-association) 섹션을 참조하시기 바랍니다.

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeVpcEndpointAssociations](https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ec2/describe-vpc-endpoint-associations.html) 섹션을 참조하시기 바랍니다.

describe-vpc-endpoint-connection-notifications

다음 코드 예시는 `describe-vpc-endpoint-connection-notifications`의 사용 방법을 보여 줍니다.

AWS CLI

엔드포인트 연결 알림 설명

다음 `describe-vpc-endpoint-connection-notifications` 예시에서는 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeVpcEndpointConnectionNotifications](#) 섹션을 참조하세요.

`describe-vpc-endpoint-connections`

다음 코드 예시는 `describe-vpc-endpoint-connections`의 사용 방법을 보여 줍니다.

AWS CLI

VPC 엔드포인트 연결 설명

이 예시에서는 엔드포인트 서비스에 대한 인터페이스 엔드포인트 연결에 대해 설명하고 결과를 필터링하여 `PendingAcceptance`인 엔드포인트를 표시합니다.

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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeVpcEndpointConnections](#) 섹션을 참조하세요.

describe-vpc-endpoint-service-configurations

다음 코드 예시는 `describe-vpc-endpoint-service-configurations`의 사용 방법을 보여 줍니다.

AWS CLI

엔드포인트 서비스 구성 설명

다음 `describe-vpc-endpoint-service-configurations` 예시에서는 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 AWS PrivateLink 사용 설명서의 개념을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeVpcEndpointServiceConfigurations 섹션을 참조하세요.

describe-vpc-endpoint-service-permissions

다음 코드 예시는 describe-vpc-endpoint-service-permissions의 사용 방법을 보여 줍니다.

AWS CLI

엔드포인트 서비스 권한 설명

이 예시에서는 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"
       }
   ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeVpcEndpointServicePermissions 섹션을 참조하세요.

describe-vpc-endpoint-services

다음 코드 예시는 describe-vpc-endpoint-services의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 모든 VPC 엔드포인트 서비스 설명

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"
    ]
}
```

예시 2: 엔드포인트 서비스에 대한 세부 정보 설명

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 AWS PrivateLink 사용 설명서의 사용 가능한 AWS 서비스 이름 보기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeVpcEndpointServices 섹션을 참조하세요.

describe-vpc-endpoints

다음 코드 예시는 describe-vpc-endpoints의 사용 방법을 보여 줍니다.

AWS CLI

VPC 엔드포인트 설명

다음 describe-vpc-endpoints 예시에서는 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 AWS PrivateLink 사용 설명서의 개념을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeVpcEndpoints 섹션을 참조하세요.

`describe-vpc-peering-connections`

다음 코드 예시는 `describe-vpc-peering-connections`의 사용 방법을 보여 줍니다.

AWS CLI

VPC 피어링 연결 설명

이 예시에서는 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 연결 설명

이 예시에서는 보류 중 수락 허용 상태인 모든 VPC 피어링 연결을 설명합니다.

명령:

```
aws ec2 describe-vpc-peering-connections --filters Name=status-code,Values=pending-acceptance
```

이 예시에서는 Owner=Finance 태그가 있는 모든 VPC 피어링 연결을 설명합니다.

명령:

```
aws ec2 describe-vpc-peering-connections --filters Name=tag:Owner,Values=Finance
```

이 예시에서는 지정된 VPC(vpc-1a2b3c4d)에 대해 요청한 모든 VPC 피어링 연결을 설명합니다.

명령:

```
aws ec2 describe-vpc-peering-connections --filters Name=requester-vpc-info.vpc-id,Values=vpc-1a2b3c4d
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeVpcPeeringConnections 섹션을 참조하세요.

`describe-vpcs`

다음 코드 예시는 `describe-vpcs`의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 모든 VPC를 설명하는 방법

다음 `describe-vpcs` 예제에서는 VPC에 대한 세부 정보를 검색합니다.

```
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 VPC를 설명하는 방법

다음 `describe-vpcs` 예제에서는 지정된 VPC에 대한 세부 정보를 검색합니다.

```
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"
                }
            ]
        }
    ]
```

```
}
```

- API 세부 정보는 AWS CLI 명령 참조에서 [DescribeVpcs](https://docs.aws.amazon.com/goto/aws-cli/ec2-2016-11-15/DescribeVpcs)를 참조하세요.

describe-vpn-connections

다음 코드 예시는 `describe-vpn-connections`의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: VPN 연결 설명

다음 `describe-vpn-connections` 예시에서는 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 AWS Site-to-Site VPN 작동 방식을 참조하세요.

예시 2: 사용 가능한 VPN 연결 설명

다음 `describe-vpn-connections` 예시에서는 상태가 `available`인 Site-to-Site VPN 연결을 설명합니다.

```
aws ec2 describe-vpn-connections \
    --filters "Name=state,Values=available"
```

자세한 내용은 AWS Site-to-Site VPN 사용 설명서의 AWS Site-to-Site VPN 작동 방식을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeVpnConnections 섹션을 참조하세요.

describe-vpn-gateways

다음 코드 예시는 `describe-vpn-gateways`의 사용 방법을 보여 줍니다.

AWS CLI

가상 프라이빗 게이트웨이 설명

이 예시에서는 가상 프라이빗 게이트웨이를 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"
                }
            ]
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeVpnGateways 섹션을 참조하세요.

detach-classic-link-vpc

다음 코드 예시는 detach-classic-link-vpc의 사용 방법을 보여 줍니다.

AWS CLI

VPC에서 EC2-Classic 인스턴스의 링크 해제(분리)

이 예시에서는 VPC vpc-88888888 에서 인스턴스 i-0598c7d356eba48d7의 연결을 해제합니다.

명령:

```
aws ec2 detach-classic-link-vpc --instance-id i-0598c7d356eba48d7 --vpc-
id vpc-88888888
```

출력:

```
{
  "Return": true
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DetachClassicLinkVpc 섹션을 참조하세요.

detach-internet-gateway

다음 코드 예시는 detach-internet-gateway의 사용 방법을 보여 줍니다.

AWS CLI

VPC에서 인터넷 게이트웨이 분리

다음 detach-internet-gateway 예시에서는 지정된 인터넷 게이트웨이를 특정 VPC에서 분리합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 DetachInternetGateway 섹션을 참조하세요.

detach-network-interface

다음 코드 예시는 detach-network-interface의 사용 방법을 보여 줍니다.

AWS CLI

인스턴스에서 네트워크 인터페이스 분리

이 예시에서는 지정된 인스턴스에서 지정된 네트워크 인터페이스를 분리합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 detach-network-interface --attachment-id eni-attach-66c4350a
```

- API 세부 정보는 AWS CLI 명령 참조의 DetachNetworkInterface 섹션을 참조하세요.

detach-verified-access-trust-provider

다음 코드 예시는 detach-verified-access-trust-provider의 사용 방법을 보여 줍니다.

AWS CLI

인스턴스에서 신뢰 공급자 분리

다음 detach-verified-access-trust-provider 예시에서는 지정된 Verified Access 인스턴스에서 지정된 Verified Access 신뢰 공급자를 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 AWS Verified Access 사용 설명서의 Verified Access instances 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DetachVerifiedAccessTrustProvider 섹션을 참조하세요.

detach-volume

다음 코드 예시는 detach-volume의 사용 방법을 보여 줍니다.

AWS CLI

인스턴스에서 볼륨 분리

이 예시 명령은 볼륨(vol-049df61146c4d7901)이 연결된 인스턴스에서 볼륨을 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DetachVolume 섹션을 참조하세요.

detach-vpn-gateway

다음 코드 예시는 detach-vpn-gateway의 사용 방법을 보여 줍니다.

AWS CLI

VPC에서 가상 프라이빗 게이트웨이 분리

이 예시에서는 지정한 가상 프라이빗 게이트웨이를 지정한 VPC에서 분리합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 detach-vpn-gateway --vpn-gateway-id vgw-9a4cacf3 --vpc-id vpc-a01106c2
```

- API 세부 정보는 AWS CLI 명령 참조의 DetachVpnGateway 섹션을 참조하세요.

disable-address-transfer

다음 코드 예시는 disable-address-transfer의 사용 방법을 보여 줍니다.

AWS CLI

탄력적 IP 주소 전송 비활성화

다음 disable-address-transfer 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 DisableAddressTransfer 섹션을 참조하세요.

disable-aws-network-performance-metric-subscription

다음 코드 예시는 `disable-aws-network-performance-metric-subscription`의 사용 방법을 보여 줍니다.

AWS CLI

지표 구독을 비활성화하는 방법

다음 `disable-aws-network-performance-metric-subscription` 예시에서는 지정된 소스와 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 CLI를 사용한 CloudWatch 구독 관리 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisableAwsNetworkPerformanceMetricSubscription 섹션을 참조하세요.

disable-ebs-encryption-by-default

다음 코드 예시는 `disable-ebs-encryption-by-default`의 사용 방법을 보여 줍니다.

AWS CLI

기본적으로 EBS 암호화 비활성화

다음 disable-ebs-encryption-by-default 예시에서는 현재 리전 내 AWS 계정에 대해 기본적으로 EBS 암호화를 비활성화합니다.

```
aws ec2 disable-ebs-encryption-by-default
```

출력:

```
{
    "EbsEncryptionByDefault": false
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DisableEbsEncryptionByDefault 섹션을 참조하세요.

disable-fast-launch

다음 코드 예시는 disable-fast-launch의 사용 방법을 보여 줍니다.

AWS CLI

이미지의 빠른 시작을 중단하는 방법

다음 disable-fast-launch 예제에서는 지정된 AMI에서 빠른 실행을 중단하고 기존의 사전 프로비저닝된 스냅샷을 정리합니다.

```
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"
}
```

자세한 내용은 Amazon EC2 사용 설명서의 Windows AMI에 대한 EC2 빠른 시작 설정 구성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisableFastLaunch 섹션을 참조하세요.

disable-fast-snapshot-restores

다음 코드 예시는 disable-fast-snapshot-restores의 사용 방법을 보여 줍니다.

AWS CLI

빠른 스냅샷 복원 비활성화

다음 disable-fast-snapshot-restores 예시에서는 지정된 가용 영역에서 지정된 스냅샷에 대해 빠른 스냅샷 복원을 사용하지 않도록 설정합니다.

```
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": []
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DisableFastSnapshotRestores 섹션을 참조하세요.

disable-image-block-public-access

다음 코드 예시는 `disable-image-block-public-access`의 사용 방법을 보여 줍니다.

AWS CLI

지정된 리전의 AMI에 대한 퍼블릭 액세스 차단 비활성화

다음 `disable-image-block-public-access` 예시에서는 지정된 리전의 계정 수준에서 AMI에 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 AMI에 대한 퍼블릭 액세스 차단을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisableImageBlockPublicAccess 섹션을 참조하세요.

disable-image-deprecation

다음 코드 예시는 `disable-image-deprecation`의 사용 방법을 보여 줍니다.

AWS CLI

AMI의 사용 중단 취소

다음 `disable-image-deprecation` 예시에서는 AMI의 사용 중단을 취소하여 `describe-images` 출력에서 `DeprecationTime` 필드를 제거합니다. 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 사용 중지를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisableImageDeprecation 섹션을 참조하세요.

disable-image-deregistration-protection

다음 코드 예시는 disable-image-deregistration-protection의 사용 방법을 보여 줍니다.

AWS CLI

등록 취소 방지 사용 해제

다음 disable-image-deregistration-protection 예제에서는 지정된 이미지에 대한 등록 취소 보호를 사용 해제합니다.

```
aws ec2 disable-image-deregistration-protection \
    --image-id ami-0b1a928a144a74ec9
```

출력:

```
{
    "Return": "disabled"
}
```

자세한 내용은 Amazon EC2 사용 설명서의 Protect an AMI from deregistration 섹션을 참조하시기 바랍니다.

- API 세부 정보는 AWS CLI 명령 참조의 DisableImageDeregistrationProtection 섹션을 참조하시기 바랍니다.

disable-image

다음 코드 예시는 disable-image의 사용 방법을 보여 줍니다.

AWS CLI

AMI 비활성화

다음 `disable-image` 예시에서는 지정된 AMI를 비활성화합니다.

```
aws ec2 disable-image \
    --image-id ami-1234567890abcdef0
```

출력:

```
{
    "Return": "true"
}
```

자세한 내용은 Amazon EC2 사용 설명서의 AMI 비활성화를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisableImage 섹션을 참조하세요.

`disable-ipam-organization-admin-account`

다음 코드 예시는 `disable-ipam-organization-admin-account`의 사용 방법을 보여 줍니다.

AWS CLI

위임된 IPAM 관리자를 비활성화하는 방법

특정 시나리오에서는 IPAM을 AWS Organizations와 통합합니다. 이렇게 하면 AWS Organizations 관리 계정이 AWS Organizations 멤버 계정을 IPAM 관리자로 위임합니다.

이 예시에서는 IPAM 관리자 계정을 위임한 AWS Organizations 관리 계정이며 해당 계정이 IPAM 관리자가 되지 못하도록 하려는 경우입니다.

이 요청을 할 때는 모든 AWS 리전을 `--region`에 사용할 수 있습니다. 원래 관리자를 위임한 리전, IPAM이 생성된 리전 또는 IPAM 운영 리전을 사용할 필요는 없습니다. 위임된 관리자 계정을 비활성화한 경우 언제든지 다시 사용하도록 설정하거나 새 계정을 IPAM 관리자로 위임할 수 있습니다.

다음 `disable-ipam-organization-admin-account` 예시에서는 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 설명서의 AWS Organizations의 계정과 IPAM 통합을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisableIpamOrganizationAdminAccount 섹션을 참조하세요.

disable-serial-console-access

다음 코드 예시는 `disable-serial-console-access`의 사용 방법을 보여 줍니다.

AWS CLI

계정의 EC2 직렬 콘솔에 대한 액세스 비활성화

다음 `disable-serial-console-access` 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 DisableSerialConsoleAccess 섹션을 참조하세요.

disable-snapshot-block-public-access

다음 코드 예시는 `disable-snapshot-block-public-access`의 사용 방법을 보여 줍니다.

AWS CLI

스냅샷에 대한 퍼블릭 액세스 차단 비활성화

다음 disable-snapshot-block-public-access 예시에서는 스냅샷에 대한 퍼블릭 액세스 차단을 비활성화하여 스냅샷의 공개 공유를 허용합니다.

```
aws ec2 disable-snapshot-block-public-access
```

출력:

```
{
    "State": "unblocked"
}
```

자세한 내용은 Amazon EBS 사용 설명서의 [스냅샷에 대한 퍼블릭 액세스 차단]을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DisableSnapshotBlockPublicAccess] 섹션을 참조하세요.

disable-transit-gateway-route-table-propagation

다음 코드 예시는 disable-transit-gateway-route-table-propagation의 사용 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 연결을 비활성화하여 지정된 전파 라우팅 테이블에 라우팅 전파

다음 disable-transit-gateway-route-table-propagation 예시에서는 지정된 전파 라우팅 테이블로 경로를 전파하지 않도록 지정된 연결을 비활성화합니다.

```
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 설명서의 전송 게이트웨이 라우팅 테이블을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisableTransitGatewayRouteTablePropagation 섹션을 참조하세요.

disable-vgw-route-propagation

다음 코드 예시는 disable-vgw-route-propagation의 사용 방법을 보여 줍니다.

AWS CLI

라우팅 전파 비활성화

이 예시에서는 지정된 가상 프라이빗 게이트웨이가 지정된 라우팅 테이블로 정적 경로를 전파하지 못하도록 합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 disable-vgw-route-propagation --route-table-id rtb-22574640 --gateway-id vgw-9a4cacf3
```

- API 세부 정보는 AWS CLI 명령 참조의 DisableVgwRoutePropagation 섹션을 참조하세요.

disable-vpc-classic-link-dns-support

다음 코드 예시는 disable-vpc-classic-link-dns-support의 사용 방법을 보여 줍니다.

AWS CLI

VPC에 대한 ClassicLink DNS 지원 비활성화

이 예시에서는 vpc-88888888에 대한 ClassicLink DNS 지원을 비활성화합니다.

명령:

```
aws ec2 disable-vpc-classic-link-dns-support --vpc-id vpc-88888888
```

출력:

```
{
```

```
  "Return": true
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DisableVpcClassicLinkDnsSupport 섹션을 참조하세요.

disable-vpc-classic-link

다음 코드 예시는 `disable-vpc-classic-link`의 사용 방법을 보여 줍니다.

AWS CLI

VPC의 ClassicLink 비활성화

이 예시에서는 vpc-8888888용 ClassicLink를 비활성화합니다.

명령:

```
aws ec2 disable-vpc-classic-link --vpc-id vpc-88888888
```

출력:

```
{
  "Return": true
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DisableVpcClassicLink 섹션을 참조하세요.

disassociate-address

다음 코드 예시는 `disassociate-address`의 사용 방법을 보여 줍니다.

AWS CLI

EC2-Classic에서 탄력적 IP 주소를 연결 해제하는 방법

이 예제에서는 EC2-Classic의 인스턴스에서 탄력적 IP 주소를 연결 해제합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 disassociate-address --public-ip 198.51.100.0
```

EC2-VPC에서 탄력적 IP 주소를 연결 해제하는 방법

이 예제에서는 VPC의 인스턴스에서 탄력적 IP 주소를 연결 해제합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 disassociate-address --association-id eipassoc-2bebb745
```

- API 세부 정보는 AWS CLI 명령 참조의 DisassociateAddress를 참조하세요.

disassociate-client-vpn-target-network

다음 코드 예시는 disassociate-client-vpn-target-network의 사용 방법을 보여 줍니다.

AWS CLI

Client VPN 엔드포인트에서 네트워크 연결 해제

다음 disassociate-client-vpn-target-network 예시에서는 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 Client VPN 관리자 안내서의 Target Networks 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisassociateClientVpnTargetNetwork 섹션을 참조하세요.

disassociate-iam-instance-profile

다음 코드 예시는 disassociate-iam-instance-profile의 사용 방법을 보여 줍니다.

AWS CLI

IAM 인스턴스 프로파일 연결 해제

이 예시에서는 IAM 인스턴스 프로파일을 연결 ID iip-assoc-05020b59952902f5f와 연결 해제합니다.

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"
      }
  }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DisassociateIamInstanceProfile 섹션을 참조하세요.

disassociate-instance-event-window

다음 코드 예시는 disassociate-instance-event-window의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 이벤트 기간에서 하나 이상의 인스턴스 연결 해제

다음 disassociate-instance-event-window 예시에서는 이벤트 기간에서 하나 이상의 인스턴스를 연결 해제합니다. instance-event-window-id 파라미터를 지정하여 이벤트 기간을 지

정합니다. 인스턴스를 연결 해제하려면 association-target 파라미터를 지정하고 파라미터 값으로 하나 이상의 인스턴스 ID를 지정합니다.

```
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 사항 섹션을 참조하세요.

예시 2: 이벤트 기간에서 인스턴스 태그의 연결 해제

다음 disassociate-instance-event-window 예시에서는 이벤트 기간에서 인스턴스 태그를 연결 해제합니다. instance-event-window-id 파라미터를 지정하여 이벤트 기간을 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 사항 섹션을 참조하세요.

예시 3: 이벤트 기간에서 전용 호스트의 연결 해제

다음 disassociate-instance-event-window 예시에서는 이벤트 기간에서 전용 호스트를 연결 해제합니다. instance-event-window-id 파라미터를 지정하여 이벤트 기간을 지정합니다. 전용 호스트를 연결 해제하려면 association-target 파라미터를 지정하고 파라미터 값으로 하나 이상의 전용 호스트 ID를 지정합니다.

```
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 사항 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisassociateInstanceEventWindow 섹션을 참조하세요.

disassociate-ipam-resource-discovery

다음 코드 예시는 disassociate-ipam-resource-discovery의 사용 방법을 보여 줍니다.

AWS CLI

리소스 검색의 연결 해제

이 예시에서는 IPAM 위임된 관리자 계정으로 IPAM 리소스 검색을 IPAM에서 연결 해제하려는 경우입니다. describe 명령을 실행한 결과 "ResourceDiscoveryStatus": "not-found"를 발견하고 다른 연결을 위한 공간을 확보하기 위해 IPAM에서 연결을 해제하려고 합니다.

다음 disassociate-ipam-resource-discovery 예시에서는 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 IPAM을 조직 외부 계정과 통합 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisassociateIpamResourceDiscovery 섹션을 참조하세요.

disassociate-nat-gateway-address

다음 코드 예시는 disassociate-nat-gateway-address의 사용 방법을 보여 줍니다.

AWS CLI

퍼블릭 NAT 게이트웨이와 탄력적 IP 주소 연결의 해제

다음 disassociate-nat-gateway-address 지정된 퍼블릭 NAT 게이트웨이에서 지정된 탄력적 IP 주소를 연결 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 정보는 Amazon VPC 사용 설명서의 NAT 게이트웨이 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisassociateNatGatewayAddress 섹션을 참조하세요.

disassociate-route-table

다음 코드 예시는 disassociate-route-table의 사용 방법을 보여 줍니다.

AWS CLI

라우팅 테이블의 연결 해제

이 예시에서는 지정한 라우팅 테이블을 지정한 서브넷에서 연결 해제합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 disassociate-route-table --association-id rtbassoc-781d0d1a
```

- API 세부 정보는 AWS CLI 명령 참조의 DisassociateRouteTable 섹션을 참조하세요.

disassociate-security-group-vpc

다음 코드 예시는 disassociate-security-group-vpc의 사용 방법을 보여 줍니다.

AWS CLI

VPC에서 보안 그룹 연결 해제

다음 disassociate-security-group-vpc 예제에서는 지정된 VPC를 지정된 보안 그룹에서 연결 해제합니다.

```
aws ec2 disassociate-security-group-vpc \
    --group-id sg-04dbb43907d3f8a78 \
    --vpc-id vpc-0bf4c2739bc05a694
```

출력:

```
{
```

```
    "State": "disassociating"
}
```

자세한 내용은 Amazon VPC 사용 설명서의 보안 그룹을 여러 VPC와 연결을 참조하시기 바랍니다.

- API 세부 정보는 AWS CLI 명령 참조의 DisassociateSecurityGroupVpc 섹션을 참조하시기 바랍니다.

disassociate-subnet-cidr-block

다음 코드 예시는 disassociate-subnet-cidr-block의 사용 방법을 보여 줍니다.

AWS CLI

서브넷에서 IPv6 CIDR 블록의 연결 해제

이 예시에서는 CIDR 블록의 연결 ID를 사용하여 서브넷에서 IPv6 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"
      }
  }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DisassociateSubnetCidrBlock 섹션을 참조하세요.

disassociate-transit-gateway-multicast-domain

다음 코드 예시는 disassociate-transit-gateway-multicast-domain의 사용 방법을 보여 줍니다.

AWS CLI

멀티캐스트 도메인에서 서브넷의 연결 해제

다음 `disassociate-transit-gateway-multicast-domain` 예시에서는 지정된 멀티캐스트 도메인에서 서브넷을 연결 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 설명서의 멀티캐스트 도메인을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisassociateTransitGatewayMulticastDomain 섹션을 참조하세요.

`disassociate-transit-gateway-route-table`

다음 코드 예시는 `disassociate-transit-gateway-route-table`의 사용 방법을 보여 줍니다.

AWS CLI

라우팅 연결에서 전송 게이트웨이 라우팅 테이블의 연결 해제

다음 disassociate-transit-gateway-route-table 예시에서는 지정된 연결을 전송 게이트웨이 라우팅 테이블에서 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 설명서의 전송 게이트웨이 라우팅 테이블을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisassociateTransitGatewayRouteTable 섹션을 참조하세요.

disassociate-vpc-cidr-block

다음 코드 예시는 disassociate-vpc-cidr-block의 사용 방법을 보여 줍니다.

AWS CLI

VPC에서 IPv6 CIDR 블록의 연결 해제

이 예시에서는 CIDR 블록의 연결 ID를 사용하여 VPC에서 IPv6 CIDR 블록의 연결을 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"
}
```

VPC에서 IPv4 CIDR 블록의 연결 해제

이 예시에서는 VPC에서 IPv4 CIDR 블록을 연결 해제합니다.

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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DisassociateVpcCidrBlock 섹션을 참조하세요.

`enable-address-transfer`

다음 코드 예시는 `enable-address-transfer`의 사용 방법을 보여 줍니다.

AWS CLI

탄력적 IP 주소 전송 활성화

다음 `enable-address-transfer` 예시에서는 지정된 탄력적 IP 주소의 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.

- API 세부 정보는 AWS CLI 명령 참조의 EnableAddressTransfer 섹션을 참조하세요.

`enable-aws-network-performance-metric-subscription`

다음 코드 예시는 `enable-aws-network-performance-metric-subscription`의 사용 방법을 보여 줍니다.

AWS CLI

지표 구독을 활성화하는 방법

다음 `enable-aws-network-performance-metric-subscription` 예시에서는 지정된 소스와 대상 리전 간의 집계 네트워크 지연 시간 모니터링을 활성화합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 EnableAwsNetworkPerformanceMetricSubscription 섹션을 참조하세요.

enable-ebs-encryption-by-default

다음 코드 예시는 enable-ebs-encryption-by-default의 사용 방법을 보여 줍니다.

AWS CLI

기본적으로 EBS 암호화 활성화

다음 enable-ebs-encryption-by-default 예시에서는 현재 리전 내 AWS 계정에 대해 기본적으로 EBS 암호화를 활성화합니다.

```
aws ec2 enable-ebs-encryption-by-default
```

출력:

```
{
    "EbsEncryptionByDefault": true
}
```

- API 세부 정보는 AWS CLI 명령 참조의 EnableEbsEncryptionByDefault 섹션을 참조하세요.

enable-fast-launch

다음 코드 예시는 enable-fast-launch의 사용 방법을 보여 줍니다.

AWS CLI

이미지에 대한 빠른 시작

다음 enable-fast-launch 예제에서는 빠른 실행을 위해 지정된 AMI를 구성하고 시작할 최대 병렬 인스턴스 수를 6으로 설정합니다. AMI를 사전 프로비저닝하는 데 사용할 리소스 유형은 기본값이기도 한 snapshot으로 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"
}
```

자세한 내용은 Amazon EC2 사용 설명서의 Windows AMI에 대한 EC2 빠른 시작 설정 구성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 EnableFastLaunch 섹션을 참조하세요.

enable-fast-snapshot-restores

다음 코드 예시는 enable-fast-snapshot-restores의 사용 방법을 보여 줍니다.

AWS CLI

빠른 스냅샷 복원 활성화

다음 enable-fast-snapshot-restores 예시에서는 지정된 가용 영역에서 지정된 스냅샷에 대해 빠른 스냅샷 복원을 비활성화합니다.

```
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": []
}
```

- API 세부 정보는 AWS CLI 명령 참조의 EnableFastSnapshotRestores 섹션을 참조하세요.

`enable-image-block-public-access`

다음 코드 예시는 `enable-image-block-public-access`의 사용 방법을 보여 줍니다.

AWS CLI

지정된 리전의 AMI에 대한 퍼블릭 액세스 차단 활성화

다음 `enable-image-block-public-access` 예시에서는 지정된 리전의 계정 수준에서 AMI에 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 AMI에 대한 퍼블릭 액세스 차단을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 EnableImageBlockPublicAccess 섹션을 참조하세요.

enable-image-deprecation

다음 코드 예시는 enable-image-deprecation의 사용 방법을 보여 줍니다.

AWS CLI

AMI를 사용 중지하려면

다음 enable-image-deprecation 예시에서는 특정 날짜와 시간에 AMI를 사용 중지합니다. 초 단위로 값을 지정하면 Amazon EC2가 초를 가장 가까운 분으로 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 사용 중지를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 EnableImageDeprecation 섹션을 참조하세요.

enable-image-deregistration-protection

다음 코드 예시는 enable-image-deregistration-protection의 사용 방법을 보여 줍니다.

AWS CLI

등록 취소 보호 사용 해제

다음 `enable-image-deregistration-protection` 예제에서는 지정된 이미지에 대한 등록 취소 보호를 사용 설정합니다.

```
aws ec2 enable-image-deregistration-protection \
    --image-id ami-0b1a928a144a74ec9
```

출력:

```
{
    "Return": "enabled-without-cooldown"
}
```

자세한 내용은 Amazon EC2 사용 설명서의 등록 취소로부터 EC2 AMI 보호를 참조하시기 바랍니다.

- API 세부 정보는 AWS CLI 명령 참조의 EnableImageDeregistrationProtection을 참조하시기 바랍니다.

`enable-image`

다음 코드 예시는 `enable-image`의 사용 방법을 보여 줍니다.

AWS CLI

AMI 활성화

다음 `enable-image` 예시에서는 지정된 AMI를 활성화합니다.

```
aws ec2 enable-image \
    --image-id ami-1234567890abcdef0
```

출력:

```
{
    "Return": "true"
}
```

자세한 내용은 Amazon EC2 사용 설명서의 AMI 비활성화를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 EnableImage 섹션을 참조하세요.

enable-ipam-organization-admin-account

다음 코드 예시는 enable-ipam-organization-admin-account의 사용 방법을 보여 줍니다.

AWS CLI

AWS Organizations와 통합하고 멤버 계정을 IPAM 계정으로 위임

다음 enable-ipam-organization-admin-account 예시에서는 IPAM을 AWS Organizations와 통합하고 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 Organizations와 IPAM 통합](https://docs.aws.amazon.com/)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [EnableIpamOrganizationAdminAccount](https://docs.aws.amazon.com/) 섹션을 참조하세요.

enable-reachability-analyzer-organization-sharing

다음 코드 예시는 enable-reachability-analyzer-organization-sharing의 사용 방법을 보여 줍니다.

AWS CLI

Reachability Analyzer의 신뢰할 수 있는 액세스 활성화

다음 enable-reachability-analyzer-organization-sharing 예시에서는 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 계정 간 분석을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 EnableReachabilityAnalyzerOrganizationSharing 섹션을 참조하세요.

enable-serial-console-access

다음 코드 예시는 enable-serial-console-access의 사용 방법을 보여 줍니다.

AWS CLI

계정의 직렬 콘솔에 대한 액세스 비활성화

다음 enable-serial-console-access 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 EnableSerialConsoleAccess 섹션을 참조하세요.

enable-snapshot-block-public-access

다음 코드 예시는 enable-snapshot-block-public-access의 사용 방법을 보여 줍니다.

AWS CLI

스냅샷에 대한 퍼블릭 액세스 차단 사용

다음 enable-snapshot-block-public-access 예시에서는 스냅샷의 모든 공개 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.

- API 세부 정보는 AWS CLI 명령 참조의 EnableSnapshotBlockPublicAccess 섹션을 참조하세요.

enable-transit-gateway-route-table-propagation

다음 코드 예시는 enable-transit-gateway-route-table-propagation의 사용 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 연결을 활성화하여 경로를 지정된 전파 라우팅 테이블에 전파하는 방법

다음 enable-transit-gateway-route-table-propagation 예시에서는 지정된 전파 라우팅 테이블로 경로를 전파하지 않도록 지정된 연결을 활성화합니다.

```
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 설명서의 전송 게이트웨이 라우팅 테이블을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [EnableTransitGatewayRouteTablePropagation](#) 섹션을 참조하세요.

enable-vgw-route-propagation

다음 코드 예시는 enable-vgw-route-propagation의 사용 방법을 보여 줍니다.

AWS CLI

라우팅 전파 활성화

이 예시에서는 지정된 가상 프라이빗 게이트웨이가 지정된 라우팅 테이블로 정적 경로를 전파하도록 설정합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 enable-vgw-route-propagation --route-table-id rtb-22574640 --gateway-id vgw-9a4cacf3
```

- API 세부 정보는 AWS CLI 명령 참조의 [EnableVgwRoutePropagation](#) 섹션을 참조하세요.

enable-volume-io

다음 코드 예시는 enable-volume-io의 사용 방법을 보여 줍니다.

AWS CLI

볼륨에 대한 I/O 활성화

이 예시에서는 vol-1234567890abcdef0 볼륨에서 I/O를 활성화합니다.

명령:

```
aws ec2 enable-volume-io --volume-id vol-1234567890abcdef0
```

출력:

```
{
  "Return": true
}
```

- API 세부 정보는 AWS CLI 명령 참조의 EnableVolumeIo 섹션을 참조하세요.

enable-vpc-classic-link-dns-support

다음 코드 예시는 enable-vpc-classic-link-dns-support의 사용 방법을 보여 줍니다.

AWS CLI

ClassicLink DNS 지원 활성화

이 예시에서는 vpc-88888888에 대한 ClassicLink DNS 지원을 활성화합니다.

명령:

```
aws ec2 enable-vpc-classic-link-dns-support --vpc-id vpc-88888888
```

출력:

```
{
  "Return": true
}
```

- API 세부 정보는 AWS CLI 명령 참조의 EnableVpcClassicLinkDnsSupport 섹션을 참조하세요.

enable-vpc-classic-link

다음 코드 예시는 enable-vpc-classic-link의 사용 방법을 보여 줍니다.

AWS CLI

VPC에서 ClassicLink 활성화

이 예시에서는 ClassicLink에 vpc-8888888을 활성화합니다.

명령:

```
aws ec2 enable-vpc-classic-link --vpc-id vpc-88888888
```

출력:

```
{
  "Return": true
```

```
}
```

- API 세부 정보는 AWS CLI 명령 참조의 EnableVpcClassicLink 섹션을 참조하세요.

`export-client-vpn-client-certificate-revocation-list`

다음 코드 예시는 `export-client-vpn-client-certificate-revocation-list`의 사용 방법을 보여 줍니다.

AWS CLI

클라이언트 인증서 해지 목록 내보내기

다음 `export-client-vpn-client-certificate-revocation-list` 예시에서는 지정된 Client VPN 엔드포인트에 대한 클라이언트 인증서 취소 목록을 내보냅니다. 이 예시에서는 읽기 쉽도록 텍스트 형식으로 출력이 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 AWS Client VPN 관리자 안내서의 클라이언트 인증서 해지 목록을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ExportClientVpnClientCertificateRevocationList](https://docs.aws.amazon.com/goto/aws-cli/ec2-2016-11-15/ExportClientVpnClientCertificateRevocationList) 섹션을 참조하세요.

export-client-vpn-client-configuration

다음 코드 예시는 `export-client-vpn-client-configuration`의 사용 방법을 보여 줍니다.

AWS CLI

클라이언트 구성 내보내기

다음 `export-client-vpn-client-configuration` 예시에서는 스냅샷의 모든 공개 공유를 차단합니다. 이 예시에서는 읽기 쉽도록 텍스트 형식으로 출력이 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 AWS Client VPN 관리자 안내서의 Client VPN 엔드포인트 구성 파일 내보내기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ExportClientVpnClientConfiguration 섹션을 참조하세요.

export-image

다음 코드 예시는 export-image의 사용 방법을 보여 줍니다.

AWS CLI

AMI에서 VM 내보내기

다음 export-image 예시에서는 지정된 AMI를 지정된 형식으로 지정된 버킷으로 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ExportImage 섹션을 참조하세요.

get-associated-ipv6-pool-cidrs

다음 코드 예시는 get-associated-ipv6-pool-cidrs의 사용 방법을 보여 줍니다.

AWS CLI

IPv6 주소 풀에 대한 연결을 가져오는 방법

다음 get-associated-ipv6-pool-cidrs 예시에서는 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 세부 정보는 AWS CLI 명령 참조의 GetAssociatedIpv6PoolCidrs 섹션을 참조하세요.

get-aws-network-performance-data

다음 코드 예시는 get-aws-network-performance-data의 사용 방법을 보여 줍니다.

AWS CLI

네트워크 성능 데이터를 가져오는 방법

다음 get-aws-network-performance-data 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 GetAwsNetworkPerformanceData 섹션을 참조하세요.

get-capacity-reservation-usage

다음 코드 예시는 `get-capacity-reservation-usage`의 사용 방법을 보여 줍니다.

AWS CLI

AWS 계정 간 용량 예약 사용량을 보는 방법

다음 `get-capacity-reservation-usage` 예시에서는 지정된 용량 예약에 대한 사용량 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 Amazon EC2 사용 설명서의 공유 용량 예약을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetCapacityReservationUsage 섹션을 참조하세요.

get-coip-pool-usage

다음 코드 예시는 get-coip-pool-usage의 사용 방법을 보여 줍니다.

AWS CLI

고객 소유 IP 주소 풀 사용을 가져오는 방법

다음 get-coip-pool-usage 예시에서는 지정된 고객 소유 IP 주소 풀의 사용 내역을 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 Outposts 랙에 대한 AWS Outposts 사용 설명서의 고객 소유 IP 주소를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetCoipPoolUsage 섹션을 참조하세요.

`get-console-output`

다음 코드 예시는 `get-console-output`의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 콘솔 출력을 가져오는 방법

다음 `get-console-output` 예시에서는 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.

예시 2: 최신 콘솔 출력을 가져오는 방법

다음 `get-console-output` 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 GetConsoleOutput 섹션을 참조하세요.

get-console-screenshot

다음 코드 예시는 `get-console-screenshot`의 사용 방법을 보여 줍니다.

AWS CLI

실행 중인 인스턴스의 스크린샷 검색

다음 `get-console-screenshot` 예시에서는 지정된 인스턴스의 스크린샷을 .jpg 형식으로 검색합니다. 스크린샷은 Base64로 인코딩된 문자열로 반환됩니다.

```
aws ec2 get-console-screenshot \
    --instance-id i-1234567890abcdef0
```

출력:

```
{
    "ImageData": "997987/8kgj49ikjhewkwwe0008084EXAMPLE",
```

```
    "InstanceId": "i-1234567890abcdef0"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetConsoleScreenshot 섹션을 참조하세요.

get-default-credit-specification

다음 코드 예시는 get-default-credit-specification의 사용 방법을 보여 줍니다.

AWS CLI

기본 크레딧 옵션 설명

다음 get-default-credit-specification 예시에서는 T2 인스턴스의 기본 크레딧 옵션을 설명합니다.

```
aws ec2 get-default-credit-specification \
    --instance-family t2
```

출력:

```
{
    "InstanceFamilyCreditSpecification": {
        "InstanceFamily": "t2",
        "CpuCredits": "standard"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetDefaultCreditSpecification 섹션을 참조하세요.

get-ebs-default-kms-key-id

다음 코드 예시는 get-ebs-default-kms-key-id의 사용 방법을 보여 줍니다.

AWS CLI

EBS 암호화를 위한 기본 CMK 설명

다음 get-ebs-default-kms-key-id 예시에서는 AWS 계정의 EBS 암호화를 위한 기본 CMK를 설명합니다.

```
aws ec2 get-ebs-default-kms-key-id
```

출력에는 EBS 암호화를 위한 기본 CMK가 표시되며, 별칭 `alias/aws/ebs`를 사용하는 AWS 관리형 CMK입니다.

```
{
    "KmsKeyId": "alias/aws/ebs"
}
```

다음 출력은 EBS 암호화를 위한 사용자 지정 CMK를 보여줍니다.

```
{
    "KmsKeyId": "arn:aws:kms:us-
west-2:123456789012:key/0ea3fef3-80a7-4778-9d8c-1c0c6EXAMPLE"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetEbsDefaultKmsKeyId 섹션을 참조하세요.

`get-ebs-encryption-by-default`

다음 코드 예시는 `get-ebs-encryption-by-default`의 사용 방법을 보여 줍니다.

AWS CLI

EBS 암호화를 기본적으로 활성화했는지 여부 설명

다음 `get-ebs-encryption-by-default` 예시에서는 현재 리전의 AWS 계정에 대해 기본적으로 EBS 암호화가 사용 설정되어 있는지 여부를 나타냅니다.

```
aws ec2 get-ebs-encryption-by-default
```

다음 출력은 기본적으로 EBS 암호화가 비활성화되어 있음을 나타냅니다.

```
{
    "EbsEncryptionByDefault": false
}
```

다음 출력은 기본적으로 EBS 암호화가 활성화되어 있음을 나타냅니다.

```
{
```

```
    "EbsEncryptionByDefault": true
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetEbsEncryptionByDefault 섹션을 참조하세요.

get-flow-logs-integration-template

다음 코드 예시는 `get-flow-logs-integration-template`의 사용 방법을 보여 줍니다.

AWS CLI

VPC 흐름 로그와 Amazon Athena의 통합을 자동화하는 CloudFormation 템플릿 생성

다음 `get-flow-logs-integration-template` 예시에서는 CloudFormation 템플릿을 생성하여 VPC 흐름 로그와 Amazon Athena의 통합을 자동화합니다.

Linux:

```
aws ec2 get-flow-logs-integration-template \
    --flow-log-id fl-1234567890abcdef0 \
    --config-delivery-s3-destination-arn arn:aws:s3:::amzn-s3-demo-bucket \
    --integrate-services
 AthenaIntegrations='[{IntegrationResultS3DestinationArn=arn:aws:s3:::amzn-s3-demo-
bucket,PartitionLoadFrequency=none,PartitionStartDate=2021-07-21T00:40:00,PartitionEndDate=2
{IntegrationResultS3DestinationArn=arn:aws:s3:::amzn-s3-demo-
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:::amzn-s3-demo-bucket ^
    --integrate-
services AthenaIntegrations=[{IntegrationResultS3DestinationArn=arn:aws:s3:::amzn-
s3-demo-
bucket,PartitionLoadFrequency=none,PartitionStartDate=2021-07-21T00:40:00,PartitionEndDate=2
{IntegrationResultS3DestinationArn=arn:aws:s3:::amzn-s3-demo-
bucket,PartitionLoadFrequency=none,PartitionStartDate=2021-07-21T00:40:00,PartitionEndDate=2
```

출력:

```
{
```

```
    "Result": "https://amzn-s3-demo-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 템플릿으로 작업을 참조하세요.

Amazon Athena 및 흐름 로그 사용에 대한 자세한 내용은 Amazon Virtual Private Cloud 사용 설명서의 Amazon Athena를 사용하여 흐름 로그 쿼리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetFlowLogsIntegrationTemplate 섹션을 참조하세요.

`get-groups-for-capacity-reservation`

다음 코드 예시는 `get-groups-for-capacity-reservation`의 사용 방법을 보여 줍니다.

AWS CLI

용량 예약이 있는 리소스 그룹 나열

다음 `get-groups-for-capacity-reservation` 예시에서는 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 Amazon EC2 사용 설명서의 용량 예약 그룹을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetGroupsForCapacityReservation 섹션을 참조하세요.

get-host-reservation-purchase-preview

다음 코드 예시는 get-host-reservation-purchase-preview의 사용 방법을 보여 줍니다.

AWS CLI

전용 호스트 예약에 대한 구매 미리 보기를 가져오는 방법

이 예시에서는 계정에서 지정한 전용 호스트에 대한 지정 전용 호스트 예약의 비용을 미리 볼 수 있습니다.

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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetHostReservationPurchasePreview 섹션을 참조하세요.

get-image-block-public-access-state

다음 코드 예시는 get-image-block-public-access-state의 사용 방법을 보여 줍니다.

AWS CLI

지정된 리전의 AMI에 대한 퍼블릭 액세스 차단 상태 가져오기

다음 get-image-block-public-access-state 예시에서는 지정된 리전 내 계정 수준에서 AMI에 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 AMI에 대한 퍼블릭 액세스 차단을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetImageBlockPublicAccessState 섹션을 참조하세요.

get-instance-types-from-instance-requirements

다음 코드 예시는 get-instance-types-from-instance-requirements의 사용 방법을 보여 줍니다.

AWS CLI

지정된 속성과 일치하는 인스턴스 유형을 미리 보는 방법

다음 get-instance-types-from-instance-requirements 예시에서는 먼저 --generate-cli-skeleton 파라미터를 사용하여 지정할 수 있는 모든 가능한 속성의 목록을 생성하고 이 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 인스턴스 요구 사항의 인스턴스 유형 가져오기 <https://docs.aws.amazon.com/cli/latest/reference/ec2/get-instance-types-from-instance-requirements.html>를 참조하세요.

`attributes.json`에 지정된 속성을 가진 인스턴스 유형을 미리 봅니다. `--cli-input-json` 파라미터를 사용하여 JSON 파일의 이름과 경로를 지정합니다. 다음 요청에서는 출력이 테이블로 형식 지정됩니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 GetInstanceTypesFromInstanceRequirements 섹션을 참조하세요.

get-instance-uefi-data

다음 코드 예시는 `get-instance-uefi-data`의 사용 방법을 보여 줍니다.

AWS CLI

인스턴스에서 UEFI 데이터 검색

다음 `get-instance-uefi-data` 예시에서는 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>
    AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAD4L/J/AODshho="
}
```


자세한 내용은 Amazon EC2 사용 설명서의 UEFI 보안 부팅을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetInstanceUefiData 섹션을 참조하세요.

get-ipam-address-history

다음 코드 예시는 `get-ipam-address-history`의 사용 방법을 보여 줍니다.

AWS CLI

CIDR 기록 보기

다음 `get-ipam-address-history` 예시에서는 CIDR의 기록을 가져옵니다.

(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.

- API 세부 정보는 AWS CLI 명령 참조의 GetIpamAddressHistory 섹션을 참조하세요.

get-ipam-discovered-accounts

다음 코드 예시는 `get-ipam-discovered-accounts`의 사용 방법을 보여 줍니다.

AWS CLI

IPAM에서 검색한 계정을 보는 방법

이 시나리오에서는 IPAM이 검색 중인 리소스를 소유한 AWS 계정을 확인하려는 IPAM 위임된 관리자입니다.

`--discovery-region`은 모니터링되는 계정 상태를 보려는 IPAM 운영 리전입니다. 예를 들어, IPAM 운영 리전이 3개인 경우 이 요청을 세 번 수행하여 각 특정 리전의 검색과 관련된 타임스탬프를 볼 수 있습니다.

다음 `get-ipam-discovered-accounts` 예시에서는 IPAM이 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 IPAM을 조직 외부 계정과 통합 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetIpamDiscoveredAccounts 섹션을 참조하세요.

get-ipam-discovered-public-addresses

다음 코드 예시는 get-ipam-discovered-public-addresses의 사용 방법을 보여 줍니다.

AWS CLI

검색된 퍼블릭 IP 주소를 보는 방법

이 예시에서는 IPAM 위임된 관리자로서 IPAM이 검색한 리소스의 IP 주소를 보려고 합니다. describe-ipam-resource-discoveries를 사용하여 리소스 검색 ID를 가져올 수 있습니다.

다음 get-ipam-discovered-public-addresses 예시에서는 리소스 검색을 위해 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 Public IP Insights 보기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetIpamDiscoveredPublicAddresses 섹션을 참조하세요.

get-ipam-discovered-resource-cidrs

다음 코드 예시는 `get-ipam-discovered-resource-cidrs`의 사용 방법을 보여 줍니다.

AWS CLI

IPAM에서 검색한 IP 주소 CIDRs을 보는 방법

이 예시에서는 IPAM이 검색하는 리소스의 IP 주소 CIDR과 관련된 세부 정보를 보려는 IPAM 위임된 관리자입니다.

이 요청을 완료하는 방법:

선택한 리소스 검색은 IPAM과 연결되어 있어야 합니다. `--resource-region`는 리소스가 생성된 AWS 리전입니다.

다음 `get-ipam-discovered-resource-cidrs` 예시에서는 IPAM이 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.

- API 세부 정보는 AWS CLI 명령 참조의 GetIpamDiscoveredResourceCidrs 섹션을 참조하세요.

`get-ipam-pool-allocations`

다음 코드 예시는 `get-ipam-pool-allocations`의 사용 방법을 보여 줍니다.

AWS CLI

IPAM 풀에서 할당된 CIDRs을 가져오는 방법

다음 `get-ipam-pool-allocations` 예시에서는 IPAM 풀에서 할당된 CIDR을 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetIpamPoolAllocations 섹션을 참조하세요.

get-ipam-pool-cidrs

다음 코드 예시는 `get-ipam-pool-cidrs`의 사용 방법을 보여 줍니다.

AWS CLI

CIDR을 IPAM 풀에 프로비저닝

다음 `get-ipam-pool-cidrs` 예시에서는 IPAM 풀에 프로비저닝된 CIDR을 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetIpamPoolCidrs 섹션을 참조하세요.

`get-ipam-resource-cidrs`

다음 코드 예시는 `get-ipam-resource-cidrs`의 사용 방법을 보여 줍니다.

AWS CLI

리소스에 할당된 CIDRs을 가져오는 방법

다음 `get-ipam-resource-cidrs` 예시에서는 리소스에 할당된 CIDR을 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.

- API 세부 정보는 AWS CLI 명령 참조의 GetIpamResourceCidrs 섹션을 참조하세요.

get-launch-template-data

다음 코드 예시는 `get-launch-template-data`의 사용 방법을 보여 줍니다.

AWS CLI

시작 템플릿에 필요한 인스턴스 데이터 가져오기

이 예시에서는 지정된 인스턴스에 대한 데이터를 가져와서 LaunchTemplateData로 내용을 반환하는 --query 옵션을 사용합니다. 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"
            }
        ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetLaunchTemplateData 섹션을 참조하세요.

get-managed-prefix-list-associations

다음 코드 예시는 get-managed-prefix-list-associations의 사용 방법을 보여 줍니다.

AWS CLI

접두사 목록 연결을 가져오는 방법

다음 get-managed-prefix-list-associations 예시에서는 지정된 접두사 목록과 연관된 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.

- API 세부 정보는 AWS CLI 명령 참조의 GetManagedPrefixListAssociations 섹션을 참조하세요.

get-managed-prefix-list-entries

다음 코드 예시는 `get-managed-prefix-list-entries`의 사용 방법을 보여 줍니다.

AWS CLI

접두사 목록 항목 가져오기

다음 `get-managed-prefix-list-entries` 명령은 지정된 접두사 목록에 대한 항목을 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.

- API 세부 정보는 AWS CLI 명령 참조의 GetManagedPrefixListEntries 섹션을 참조하세요.

get-network-insights-access-scope-analysis-findings

다음 코드 예시는 `get-network-insights-access-scope-analysis-findings`의 사용 방법을 보여 줍니다.

AWS CLI

Network Insights 액세스 범위 분석 결과를 얻으려면

다음 `get-network-insights-access-scope-analysis-findings` 예시에서는 AWS 계정의 특정 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 설명서의 AWS CLI를 사용하여 Network Access Analyzer 시작하기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetNetworkInsightsAccessScopeAnalysisFindings 섹션을 참조하세요.

get-network-insights-access-scope-content

다음 코드 예시는 `get-network-insights-access-scope-content`의 사용 방법을 보여 줍니다.

AWS CLI

Network Insights 액세스 범위 콘텐츠를 가져오는 방법

다음 `get-network-insights-access-scope-content` 예시에서는 AWS 계정의 특정 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 설명서의 AWS CLI를 사용하여 Network Access Analyzer 시작하기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetNetworkInsightsAccessScopeContent 섹션을 참조하세요.

get-password-data

다음 코드 예시는 get-password-data의 사용 방법을 보여 줍니다.

AWS CLI

암호화된 암호를 가져오는 방법

이 예시에서는 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 가져오는 방법

이 예시에서는 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetPasswordData 섹션을 참조하세요.

get-reserved-instances-exchange-quote

다음 코드 예시는 `get-reserved-instances-exchange-quote`의 사용 방법을 보여 줍니다.

AWS CLI

전환형 예약 인스턴스 교환에 대한 견적을 가져오기

이 예시에서는 지정된 전환형 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [GetReservedInstancesExchangeQuote](#) 섹션을 참조하세요.

get-security-groups-for-vpc

다음 코드 예시는 `get-security-groups-for-vpc`의 사용 방법을 보여 줍니다.

AWS CLI

지정된 VPC의 네트워크 인터페이스와 연결할 수 있는 보안 그룹을 보는 방법

다음 `get-security-groups-for-vpc` 예시에서는 VPC의 네트워크 인터페이스에 연결할 수 있는 보안 그룹을 보여줍니다.

```
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"
```

```
        }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetSecurityGroupsForVpc 섹션을 참조하세요.

get-serial-console-access-status

다음 코드 예시는 get-serial-console-access-status의 사용 방법을 보여 줍니다.

AWS CLI

직렬 콘솔에 대한 계정 액세스 상태 보기

다음 get-serial-console-access-status 예시에서는 계정에 대해 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.

- API 세부 정보는 AWS CLI 명령 참조의 GetSerialConsoleAccessStatus 섹션을 참조하세요.

get-snapshot-block-public-access-state

다음 코드 예시는 get-snapshot-block-public-access-state의 사용 방법을 보여 줍니다.

AWS CLI

스냅샷에 대한 퍼블릭 액세스 차단의 현재 상태를 가져오는 방법

다음 get-snapshot-block-public-access-state 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 GetSnapshotBlockPublicAccessState 섹션을 참조하세요.

get-spot-placement-scores

다음 코드 예시는 get-spot-placement-scores의 사용 방법을 보여 줍니다.

AWS CLI

지정된 요구 사항에 대한 스팟 배치 점수 계산

다음 get-spot-placement-scores 예시에서는 먼저 --generate-cli-skeleton 파라미터를 사용하여 스팟 배치 점수 구성에 지정할 수 있는 모든 가능한 파라미터 목록을 생성하고 이 목록을 JSON 파일에 저장합니다. 그런 다음 JSON 파일을 사용하여 스팟 배치 점수를 계산하는 데 사용할 요구 사항을 구성합니다.

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/AWSEC2/latest/UserGuide/spot-placement-score .html#calculate-sps-cli>을 참조하세요.

`attributes.json`에 지정된 요구 사항에 대한 스팟 배치 점수를 계산합니다. `--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 `false`로 설정되거나 생략된 경우 출력합니다(생략된 경우 기본값은 `false`). 점수가 매겨진 리전 목록이 반환됩니다.

```
"Recommendation": [
    {
        "Region": "us-east-1",
        "Score": 7
    },
    {
        "Region": "us-west-1",
        "Score": 5
    },
   ...
```

`SingleAvailabilityZone`이 `true`로 설정된 경우 출력입니다. 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 계산 및 구성 예시에 대한 자세한 내용은 Amazon EC2 사용 설명서의 스팟 배치 점수 계산을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetSpotPlacementScores 섹션을 참조하세요.

`get-subnet-cidr-reservations`

다음 코드 예시는 `get-subnet-cidr-reservations`의 사용 방법을 보여 줍니다.

AWS CLI

서브넷 CIDR 예약에 대한 정보를 가져오는 방법

다음 `get-subnet-cidr-reservations` 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 GetSubnetCidrReservations 섹션을 참조하세요.

get-transit-gateway-attachment-propagations

다음 코드 예시는 `get-transit-gateway-attachment-propagations`의 사용 방법을 보여 줍니다.

AWS CLI

지정된 리소스 연결이 라우팅을 전파하는 라우팅 테이블 나열

다음 `get-transit-gateway-attachment-propagations` 예시에서는 지정된 리소스 연결이 경로를 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 설명서의 전송 게이트웨이 라우팅 테이블을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetTransitGatewayAttachmentPropagations 섹션을 참조하세요.

get-transit-gateway-multicast-domain-associations

다음 코드 예시는 get-transit-gateway-multicast-domain-associations의 사용 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 멀티캐스트 도메인에 대한 연결 정보 보기

다음 get-transit-gateway-multicast-domain-associations 예시에서는 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 설명서의 멀티캐스트 도메인 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetTransitGatewayMulticastDomainAssociations 섹션을 참조하세요.

get-transit-gateway-prefix-list-references

다음 코드 예시는 get-transit-gateway-prefix-list-references의 사용 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 라우팅 테이블에서 접두사 목록 참조

다음 get-transit-gateway-prefix-list-references 예시에서는 지정된 전송 게이트웨이 라우팅 테이블에 대한 접두사 목록 참조를 가져오고 특정 접두사 목록의 ID를 기준으로 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 설명서의 [접두사 목록 참조](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [GetTransitGatewayPrefixListReferences](#) 섹션을 참조하세요.

get-transit-gateway-route-table-associations

다음 코드 예시는 `get-transit-gateway-route-table-associations`의 사용 방법을 보여 줍니다.

AWS CLI

지정된 전송 게이트웨이 라우팅 테이블에 대한 연결에 대한 정보 가져오기

다음 `get-transit-gateway-route-table-associations` 예시에서는 지정된 전송 게이트웨이 라우팅 테이블에 대한 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 설명서의 전송 게이트웨이 라우팅 테이블을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetTransitGatewayRouteTableAssociations 섹션을 참조하세요.

get-transit-gateway-route-table-propagations

다음 코드 예시는 `get-transit-gateway-route-table-propagations`의 사용 방법을 보여 줍니다.

AWS CLI

지정된 전송 게이트웨이 라우팅 테이블의 라우팅 테이블 전파에 대한 정보 표시

다음 get-transit-gateway-route-table-propagations 예시에서는 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 설명서의 전송 게이트웨이 라우팅 테이블을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetTransitGatewayRouteTablePropagations 섹션을 참조하세요.

get-verified-access-endpoint-policy

다음 코드 예시는 get-verified-access-endpoint-policy의 사용 방법을 보여 줍니다.

AWS CLI

엔드포인트의 Verified Access 정책을 가져오는 방법

다음 `get-verified-access-endpoint-policy` 예시에서는 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 AWS Verified Access 사용 설명서의 [Verified Access policies](https://docs.aws.amazon.com/verified-access/latest/ug/auth-policies.html) 정책을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [GetVerifiedAccessEndpointPolicy](https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ec2/get-verified-access-endpoint-policy.html) 섹션을 참조하세요.

`get-verified-access-group-policy`

다음 코드 예시는 `get-verified-access-group-policy`의 사용 방법을 보여 줍니다.

AWS CLI

그룹의 Verified Access 정책을 가져오는 방법

다음 `get-verified-access-group-policy` 예시에서는 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 AWS Verified Access 사용 설명서의 Verified Access groups 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetVerifiedAccessGroupPolicy 섹션을 참조하세요.

get-vpn-connection-device-sample-configuration

다음 코드 예시는 get-vpn-connection-device-sample-configuration의 사용 방법을 보여 줍니다.

AWS CLI

단순 구성 파일 다운로드

다음 get-vpn-connection-device-sample-configuration 예시에서는 지정된 샘플 구성 파일을 다운로드합니다. 게이트웨이 디바이스를 샘플 구성 파일로 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.

- API 세부 정보는 AWS CLI 명령 참조의 GetVpnConnectionDeviceSampleConfiguration 섹션을 참조하세요.

get-vpn-connection-device-types

다음 코드 예시는 get-vpn-connection-device-types의 사용 방법을 보여 줍니다.

AWS CLI

샘플 구성 파일이 있는 게이트웨이 디바이스 나열

다음 `get-vpn-connection-device-types` 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 GetVpnConnectionDeviceTypes 섹션을 참조하세요.

import-client-vpn-client-certificate-revocation-list

다음 코드 예시는 `import-client-vpn-client-certificate-revocation-list`의 사용 방법을 보여 줍니다.

AWS CLI

클라이언트 인증서 해지 목록 가져오기

다음 `import-client-vpn-client-certificate-revocation-list` 예시에서는 로컬 컴퓨터의 파일 위치를 지정하여 클라이언트 인증서 해지 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 AWS Client VPN 관리자 안내서의 [클라이언트 인증서 해지 목록](#)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ImportClientVpnClientCertificateRevocationList](#) 섹션을 참조하세요.

`import-image`

다음 코드 예시는 `import-image`의 사용 방법을 보여 줍니다.

AWS CLI

VM 이미지 파일을 AMI로 가져오는 방법

다음 `import-image` 예시에서는 지정된 OVA를 가져옵니다.

```
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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ImportImage 섹션을 참조하세요.

`import-key-pair`

다음 코드 예시는 `import-key-pair`의 사용 방법을 보여 줍니다.

AWS CLI

퍼블릭 키 가져오기

먼저 원하는 도구로 키 쌍을 생성합니다. 예를 들어 다음 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 예시 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ImportKeyPair 섹션을 참조하세요.

`import-snapshot`

다음 코드 예시는 `import-snapshot`의 사용 방법을 보여 줍니다.

AWS CLI

스냅샷을 가져오는 방법

다음 `import-snapshot` 예시에서는 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"
        }
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ImportSnapshot 섹션을 참조하세요.

list-images-in-recycle-bin

다음 코드 예시는 list-images-in-recycle-bin의 사용 방법을 보여 줍니다.

AWS CLI

휴지통에 이미지 나열

다음 list-images-in-recycle-bin 예시에서는 현재 휴지통에 보관되어 있는 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 EBS 사용 설명서의 휴지통에서 삭제된 AMI 복구를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListImagesInRecycleBin 섹션을 참조하세요.

list-snapshots-in-recycle-bin

다음 코드 예시는 list-snapshots-in-recycle-bin의 사용 방법을 보여 줍니다.

AWS CLI

휴지통의 스냅샷 보기

다음 `list-snapshots-in-recycle-bin` 예시에서는 휴지통에 있는 스냅샷에 대한 정보를 나열합니다. 이러한 정보로는 스냅샷 ID, 스냅샷에 대한 설명을 포함하여 스냅샷에 대한 정보를 나열합니다. 스냅샷이 생성된 볼륨의 ID, 스냅샷이 삭제되어 휴지통에 들어간 날짜와 시간, 보존 기간이 만료되는 날짜와 시간이 있습니다.

```
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"
        }
    ]
}
```

휴지통에 대한 자세한 내용은 Amazon EBS 사용 설명서의 휴지통에서 삭제된 스냅샷 복구를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListSnapshotsInRecycleBin 섹션을 참조하세요.

`lock-snapshot`

다음 코드 예시는 `lock-snapshot`의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 거버넌스 모드에서 스냅샷 잠금

다음 `lock-snapshot` 예시에서는 지정된 스냅샷을 거버넌스 모드에서 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 Snapshot Lock을 참조하세요.

예시 2: 규정 준수 모드에서 스냅샷 잠금

다음 `lock-snapshot` 예시에서는 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 Snapshot Lock을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 LockSnapshot 섹션을 참조하세요.

`modify-address-attribute`

다음 코드 예시는 `modify-address-attribute`의 사용 방법을 보여 줍니다.

AWS CLI

탄력적 IP 주소와 연결된 도메인 이름 속성 수정

다음 `modify-address-attribute` 예시에서는 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 AWS CLI 명령 참조의 describe-addresses-attribute를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyAddressAttribute 섹션을 참조하세요.

modify-availability-zone-group

다음 코드 예시는 modify-availability-zone-group의 사용 방법을 보여 줍니다.

AWS CLI

영역 그룹 활성화

다음 modify-availability-zone-group 예시에서는 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 Amazon EC2 사용 설명서에서 리전 및 가용 영역을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyAvailabilityZoneGroup 섹션을 참조하세요.

modify-capacity-reservation-fleet

다음 코드 예시는 modify-capacity-reservation-fleet의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 용량 예약 플릿의 총 목표 용량 수정

다음 modify-capacity-reservation-fleet 예시에서는 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
}
```

예시 2: 용량 예약 플릿의 종료 날짜 수정

다음 modify-capacity-reservation-fleet 예시에서는 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 예약에 대한 자세한 내용은 Amazon EC2 사용 설명서의 용량 예약 플릿을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyCapacityReservationFleet 섹션을 참조하세요.

modify-capacity-reservation

다음 코드 예시는 modify-capacity-reservation의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 기존 용량 예약에서 예약한 인스턴스 수 변경

다음 modify-capacity-reservation 예시에서는 용량 예약이 용량을 예약하는 인스턴스 수를 변경합니다.

```
aws ec2 modify-capacity-reservation \
    --capacity-reservation-id cr-1234abcd56EXAMPLE \
    --instance-count 5
```

출력:

```
{
    "Return": true
}
```

자세한 내용은 Amazon EC2 사용 설명서의 [용량 예약 수정]을 참조하세요.

예시 2: 기존 용량 예약의 종료 날짜 및 시간 변경

다음 `modify-capacity-reservation` 예시에서는 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 Amazon EC2 사용 설명서의 [용량 예약 수정]을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ModifyCapacityReservation] 섹션을 참조하세요.

`modify-client-vpn-endpoint`

다음 코드 예시는 `modify-client-vpn-endpoint`의 사용 방법을 보여 줍니다.

AWS CLI

Client VPN 엔드포인트 수정

다음 `modify-client-vpn-endpoint` 예시에서는 지정된 Client 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 AWS Client VPN 관리자 안내서의 Client VPN 엔드포인트를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyClientVpnEndpoint 섹션을 참조하세요.

modify-default-credit-specification

다음 코드 예시는 modify-default-credit-specification의 사용 방법을 보여 줍니다.

AWS CLI

기본 크레딧 옵션 수정

다음 modify-default-credit-specification 예시에서는 T2 인스턴스의 기본 크레딧 옵션을 수정합니다.

```
aws ec2 modify-default-credit-specification \
    --instance-family t2 \
    --cpu-credits unlimited
```

출력:

```
{
    "InstanceFamilyCreditSpecification": {
        "InstanceFamily": "t2",
        "CpuCredits": "unlimited"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ModifyDefaultCreditSpecification 섹션을 참조하세요.

modify-ebs-default-kms-key-id

다음 코드 예시는 modify-ebs-default-kms-key-id의 사용 방법을 보여 줍니다.

AWS CLI

EBS 암호화를 위한 기본 CMK 설정

다음 modify-ebs-default-kms-key-id 예시에서는 지정된 CMK를 현재 리전 내 AWS 계정의 EBS 암호화에 대한 기본 CMK로 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ModifyEbsDefaultKmsKeyId 섹션을 참조하세요.

modify-fleet

다음 코드 예시는 modify-fleet의 사용 방법을 보여 줍니다.

AWS CLI

EC2 플릿 크기 조정

다음 modify-fleet 예시에서는 지정된 EC2 플릿의 목표 용량을 수정합니다. 지정된 값이 현재 용량보다 크면 EC2 플릿이 추가 인스턴스를 시작합니다. 지정된 값이 현재 용량보다 작으면 EC2 플릿은 열려 있는 모든 요청을 취소하고, 종료 정책이 terminate인 경우 EC2 플릿은 새 목표 용량을 초과하는 모든 인스턴스를 종료합니다.

```
aws ec2 modify-fleet \
    --fleet-ids fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE \
    --target-capacity-specification TotalTargetCapacity=5
```

출력:

```
{
    "Return": true
}
```

자세한 내용은 Amazon EC2 사용 설명서의 EC2 플릿 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyFleet 섹션을 참조하세요.

modify-fpga-image-attribute

다음 코드 예시는 modify-fpga-image-attribute의 사용 방법을 보여 줍니다.

AWS CLI

Amazon FPGA 이미지의 속성 수정

이 예시에서는 지정된 AFI에 대한 계정 ID 123456789012에 대한 로드 권한을 추가합니다.

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"
          }
      ]
  }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ModifyFpgaImageAttribute 섹션을 참조하세요.

modify-hosts

다음 코드 예시는 modify-hosts의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 전용 호스트에 대한 자동 배치 활성화

다음 modify-hosts 예시에서는 전용 호스트에 대한 자동 배치를 활성화하여 인스턴스 유형 구성과 일치하는 모든 타겟팅되지 않은 인스턴스 시작을 허용합니다.

```
aws ec2 modify-hosts \
```

```
    --host-id h-06c2f189b4EXAMPLE \
    --auto-placement on
```

출력:

```
{
    "Successful": [
        "h-06c2f189b4EXAMPLE"
    ],
    "Unsuccessful": []
}
```

자세한 내용은 Amazon EC2 사용 설명서의 [전용 호스트의 자동 배치 설정 수정]을 참조하세요.

예시 2: 전용 호스트에 대한 호스트 복구 활성화

다음 modify-hosts 예시에서는 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 Amazon EC2 사용 설명서의 전용 호스트의 자동 배치 설정 수정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyHosts 섹션을 참조하세요.

modify-id-format

다음 코드 예시는 modify-id-format의 사용 방법을 보여 줍니다.

AWS CLI

리소스에 더 긴 ID 형식 활성화

다음 modify-id-format 예시에서는 instance 리소스 유형에 대해 더 긴 ID 형식을 활성화합니다.

```
aws ec2 modify-id-format \
    --resource instance \
    --use-long-ids
```

리소스의 더 긴 ID 형식 비활성화

다음 modify-id-format 예시에서는 instance 리소스 유형에 대해 더 긴 ID 형식을 비활성화합니다.

```
aws ec2 modify-id-format \
    --resource instance \
    --no-use-long-ids
```

다음 modify-id-format 예시에서는 옵트인 기간 내에 있는 지원되는 모든 리소스 유형에 대해 더 긴 ID 형식을 사용하도록 설정합니다.

```
aws ec2 modify-id-format \
    --resource all-current \
    --use-long-ids
```

- API 세부 정보는 AWS CLI 명령 참조의 ModifyIdFormat 섹션을 참조하세요.

modify-identity-id-format

다음 코드 예시는 modify-identity-id-format의 사용 방법을 보여 줍니다.

AWS CLI

리소스에 더 긴 ID를 사용하도록 IAM 역할 활성화

다음 modify-identity-id-format 예시에서는 AWS 계정의 IAM 역할 EC2Role이 instance 리소스 유형에 긴 ID 형식을 사용하도록 활성화합니다.

```
aws ec2 modify-identity-id-format \
    --principal-arn arn:aws:iam::123456789012:role/EC2Role \
    --resource instance \
    --use-long-ids
```

리소스에 더 긴 ID를 사용하도록 IAM 사용자 활성화

다음 `modify-identity-id-format` 예시에서는 AWS 계정의 IAM 사용자 `AdminUser`가 `volume` 리소스 유형에 긴 ID 형식을 사용하도록 활성화합니다.

```
aws ec2 modify-identity-id-format \
    --principal-arn arn:aws:iam::123456789012:user/AdminUser \
    --resource volume \
    --use-long-ids
```

다음 `modify-identity-id-format` 예시에서는 AWS 계정의 IAM 사용자 `AdminUser`가 옵트인 기간 내에 있는 지원되는 모든 리소스 유형에 대해 더 긴 ID 형식을 사용할 수 있도록 활성화합니다.

```
aws ec2 modify-identity-id-format \
    --principal-arn arn:aws:iam::123456789012:user/AdminUser \
    --resource all-current \
    --use-long-ids
```

- API 세부 정보는 AWS CLI 명령 참조의 [ModifyIdentityIdFormat](#) 섹션을 참조하세요.

`modify-image-attribute`

다음 코드 예시는 `modify-image-attribute`의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 퍼블릭 AMI로 설정

다음 `modify-instance-attribute` 예시에서는 지정된 AMI를 퍼블릭으로 설정합니다.

```
aws ec2 modify-image-attribute \
    --image-id ami-5731123e \
    --launch-permission "Add=[{Group=all}]"
```

이 명령은 출력을 생성하지 않습니다.

예시 2: 프라이빗 AMI로 설정

다음 `modify-instance-attribute` 예시에서는 지정된 AMI를 프라이빗으로 설정합니다.

```
aws ec2 modify-image-attribute \
    --image-id ami-5731123e \
    --launch-permission "Remove=[{Group=all}]"
```

이 명령은 출력을 생성하지 않습니다.

예시 3: AWS 계정에 시작 권한 부여

다음 `modify-instance-attribute` 예시에서는 지정한 AWS 계정에 특정 AMI의 시작 권한을 허용하는 데 사용됩니다.

```
aws ec2 modify-image-attribute \
    --image-id ami-5731123e \
    --launch-permission "Add=[{UserId=123456789012}]"
```

이 명령은 출력을 생성하지 않습니다.

예시 4: AWS 계정에서 시작 권한 제거

다음 `modify-instance-attribute` 예시에서는 지정된 AWS 계정에서 시작 권한을 제거합니다.

```
aws ec2 modify-image-attribute \
    --image-id ami-5731123e \
    --launch-permission "Remove=[{UserId=123456789012}]"
```

- API 세부 정보는 AWS CLI 명령 참조의 ModifyImageAttribute 섹션을 참조하세요.

`modify-instance-attribute`

다음 코드 예시는 `modify-instance-attribute`의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 인스턴스 유형 수정

다음 `modify-instance-attribute` 예시에서는 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.

예시 2: 인스턴스에서 향상된 네트워킹 활성화

다음 `modify-instance-attribute` 예시에서는 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.

예시 3: sourceDestCheck 속성 수정

다음 `modify-instance-attribute` 예시에서는 지정된 인스턴스의 `sourceDestCheck` 속성을 `true`로 설정합니다. 인스턴스가 VPC에 있어야 합니다.

```
aws ec2 modify-instance-attribute --instance-id i-1234567890abcdef0 --source-dest-check "{\"Value\": true}"
```

이 명령은 출력을 생성하지 않습니다.

예시 4: 루트 볼륨의 deleteOnTermination 속성 수정법

다음 `modify-instance-attribute` 예시에서는 지정된 Amazon EBS 지원 인스턴스의 루트 볼륨에 대한 `deleteOnTermination` 속성을 `false`로 설정합니다. 기본적으로 이 속성은 루트 볼륨의 경우 `true`입니다.

명령:

```
aws ec2 modify-instance-attribute \
  --instance-id i-1234567890abcdef0 \
  --block-device-mappings "[{\"DeviceName\": \"/dev/sda1\",\"Ebs\":{\"DeleteOnTermination\":false}}]"
```

이 명령은 출력을 생성하지 않습니다.

예시 5: 인스턴스에 연결된 사용자 데이터 수정

다음 `modify-instance-attribute` 예시에서는 `UserData.txt` 파일의 콘텐츠를 지정된 인스턴스에 대한 UserData로 추가합니다.

원본 파일 `UserData.txt`의 콘텐츠:

```
#!/bin/bash
yum update -y
service httpd start
chkconfig httpd on
```

파일의 콘텐츠는 base64로 인코딩되어야 합니다. 첫 번째 명령은 텍스트 파일을 base64로 변환하여 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 EC2 사용 설명서의 [사용자 데이터 및 AWS CLI]를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ModifyInstanceAttribute] 섹션을 참조하세요.

modify-instance-capacity-reservation-attributes

다음 코드 예시는 modify-instance-capacity-reservation-attributes의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 인스턴스의 용량 예약 대상 설정 수정

다음 modify-instance-capacity-reservation-attributes 예시는 인스턴스를 수정하여 특정 용량 예약을 대상으로 지정합니다.

```
aws ec2 modify-instance-capacity-reservation-attributes \
    --instance-id i-EXAMPLE8765abcd4e \
    --capacity-reservation-specification
 'CapacityReservationTarget={CapacityReservationId= cr-1234abcd56EXAMPLE }'
```

출력:

```
{
    "Return": true
}
```

자세한 내용은 Amazon EC2 사용 설명서의 인스턴스의 용량 예약 설정 수정을 참조하세요.

예시 2: 인스턴스의 용량 예약 대상 설정 수정

다음 modify-instance-capacity-reservation-attributes 예시에서는 지정된 용량 예약을 대상으로 하는 중지된 인스턴스가 일치하는 속성(인스턴스 유형, 플랫폼, 가용 영역)이 있고 열려 있는 인스턴스 일치 조건이 있는 모든 용량 예약에서 실행되도록 수정합니다.

```
aws ec2 modify-instance-capacity-reservation-attributes \
    --instance-id i-EXAMPLE8765abcd4e \
    --capacity-reservation-specification 'CapacityReservationPreference=open'
```

출력:

```
{
    "Return": true
}
```

자세한 내용은 Amazon EC2 사용 설명서의 인스턴스의 용량 예약 설정 수정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyInstanceCapacityReservationAttributes 섹션을 참조하세요.

modify-instance-credit-specification

다음 코드 예시는 `modify-instance-credit-specification`의 사용 방법을 보여 줍니다.

AWS CLI

인스턴스의 CPU 사용량에 대한 크레딧 옵션 수정

이 예시에서는 지정된 리전에서 지정된 인스턴스의 CPU 사용량에 대한 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": []
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ModifyInstanceCreditSpecification 섹션을 참조하세요.

modify-instance-event-start-time

다음 코드 예시는 `modify-instance-event-start-time`의 사용 방법을 보여 줍니다.

AWS CLI

인스턴스의 이벤트 시작 시간 수정

다음 modify-instance-event-start-time 명령은 지정된 인스턴스의 이벤트 시작 시간 수정을 보여줍니다. --instance-event-id 파라미터를 사용하여 이벤트 ID를 지정합니다. --not-before 파라미터를 사용하여 새 날짜와 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 재부팅을 위해 정기 인스턴스 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyInstanceEventStartTime 섹션을 참조하세요.

modify-instance-event-window

다음 코드 예시는 modify-instance-event-window의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 이벤트 기간의 시간 범위 수정

다음 modify-instance-event-window 예시에서는 이벤트 기간의 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 사항 섹션을 참조하세요.

예시 2: 이벤트 기간의 시간 범위 세트 수정

다음 modify-instance-event-window 예시에서는 이벤트 기간의 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 사항 섹션을 참조하세요.

예시 3: 이벤트 기간의 cron 표현식 수정

다음 `modify-instance-event-window` 예시에서는 이벤트 기간의 cron 표현식을 수정합니다. `cron-expression` 파라미터를 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 사항 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyInstanceEventWindow 섹션을 참조하세요.

modify-instance-maintenance-options

다음 코드 예시는 modify-instance-maintenance-options의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 인스턴스의 복구 동작 비활성

다음 modify-instance-maintenance-options 예시에서는 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 Amazon EC2 사용 설명서의 간소화된 자동 복구 구성을 참조하세요.

예시 2: 인스턴스의 복구 동작 기본값으로 설정

다음 modify-instance-maintenance-options 예시에서는 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 Amazon EC2 사용 설명서의 간소화된 자동 복구 구성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyInstanceMaintenanceOptions 섹션을 참조하세요.

modify-instance-metadata-options

다음 코드 예시는 modify-instance-metadata-options의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: IMDSv2 활성화

다음 modify-instance-metadata-options 예시에서는 지정된 인스턴스에서 IMDSv2 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 Amazon EC2 사용 설명서의 인스턴스 메타데이터를 참조하세요.

예시 2: 인스턴스 메타데이터 비활성화

다음 modify-instance-metadata-options 예시에서는 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 Amazon EC2 사용 설명서의 [인스턴스 메타데이터](#)를 참조하세요.

예시 3: 인스턴스에 대해 인스턴스 메타데이터 IPv6 엔드포인트 활성화

다음 `modify-instance-metadata-options` 예시에서는 인스턴스 메타데이터 서비스에 대해 IPv6 엔드포인트를 설정하는 방법을 보여줍니다. 기본적으로 IPv6 엔드포인트는 비활성화되어 있습니다. IPv6 전용 서브넷으로 인스턴스를 시작한 경우에도 마찬가지입니다. IMDS용 IPv6 엔드포인트는 Nitro System에 구축된 인스턴스에서만 액세스할 수 있습니다.

```
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"
    }
}
```

자세한 내용은 Amazon EC2 사용 설명서의 [인스턴스 메타데이터](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ModifyInstanceMetadataOptions](#) 섹션을 참조하세요.

modify-instance-placement

다음 코드 예시는 modify-instance-placement의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 전용 호스트와 인스턴스의 친화성 제거

다음 modify-instance-placement 예시에서는 인스턴스와 전용 호스트의 친화성을 제거하고 인스턴스 유형을 지원하는 계정에서 사용 가능한 모든 전용 호스트에서 인스턴스를 실행할 수 있도록 합니다.

```
aws ec2 modify-instance-placement \
    --instance-id i-0e6ddf6187EXAMPLE \
    --affinity default
```

출력:

```
{
    "Return": true
}
```

예시 2: 인스턴스와 지정된 전용 호스트 간에 친화성 설정

다음 modify-instance-placement 예시에서는 인스턴스와 전용 호스트 간에 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
}
```

예시 3: 배치 그룹으로 인스턴스 이동

다음 modify-instance-placement 예시에서는 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
```

예시 4: 배치 그룹에서 인스턴스 제거

다음 `modify-instance-placement` 예시에서는 인스턴스를 중지하고 인스턴스 배치를 수정한 다음 인스턴스를 다시 시작하여 배치 그룹에서 인스턴스를 제거합니다. 다음 예시에서는 배치 그룹 이름에 빈 문자열("")을 지정하여 인스턴스가 배치 그룹에 위치하지 않음을 나타냅니다.

인스턴스를 중지합니다.

```
aws ec2 stop-instances \
    --instance-ids i-0123a456700123456
```

배치 수정(Windows 명령 프롬프트):

```
aws ec2 modify-instance-placement \
    --instance-id i-0123a456700123456 \
    --group-name ""
```

배치 수정(Windows PowerShell, Linux 및 macOS):

```
aws ec2 modify-instance-placement `
    --instance-id i-0123a456700123456 `
    --group-name ''
```

인스턴스 다시 시작:

```
aws ec2 start-instances \
    --instance-ids i-0123a456700123456
```

출력:

```
{
```

```
    "Return": true
}
```

자세한 내용은 Amazon EC2 사용 설명서의 전용 호스트 테넌시 및 선호도 수정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyInstancePlacement 섹션을 참조하세요.

modify-ipam-pool

다음 코드 예시는 `modify-ipam-pool`의 사용 방법을 보여 줍니다.

AWS CLI

IPAM 풀 수정

다음 `modify-ipam-pool` 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyIpamPool 섹션을 참조하세요.

modify-ipam-resource-cidr

다음 코드 예시는 modify-ipam-resource-cidr의 사용 방법을 보여 줍니다.

AWS CLI

리소스에 할당된 CIDR 수정

다음 modify-ipam-resource-cidr 예시에서는 리소스 CIDR을 수정합니다.

(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 간에 리소스 CIDR 이동을 참조하세요.

모니터링 상태 변경에 대한 자세한 내용은 Amazon VPC IPAM 사용 설명서의 리소스 CIDR의 모니터링 상태 변경을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyIpamResourceCidr 섹션을 참조하세요.

modify-ipam-resource-discovery

다음 코드 예시는 modify-ipam-resource-discovery의 사용 방법을 보여 줍니다.

AWS CLI

리소스 검색의 운영 리전 수정

이 예시에서는 리소스 검색의 작동 리전을 수정하려는 IPAM 위임된 관리자입니다.

이 요청을 완료하는 방법:

기본 리소스 검색은 수정할 수 없으며 리소스 검색의 소유자여야 합니다. 리소스 검색 ID가 필요하며, 이는 describe-ipam-resource-discoveries를 통해 얻을 수 있습니다.

다음 modify-ipam-resource-discovery 예시에서는 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 설명서에서 리소스 검색 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyIpamResourceDiscovery 섹션을 참조하세요.

modify-ipam-scope

다음 코드 예시는 modify-ipam-scope의 사용 방법을 보여 줍니다.

AWS CLI

범위 대상 수정

이 시나리오에서는 IPAM 범위의 설명을 수정하려는 IPAM 위임된 관리자입니다.

이 요청을 완료하려면 describe-ipam-scopes로 가져올 수 있는 범위 ID가 필요합니다.

다음 modify-ipam-scope 예시에서는 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"
    }
}
```

자세한 내용은 Amazon VPC IPAM 사용 설명서의 IPAM 작동 방식을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyIpamScope 섹션을 참조하세요.

modify-ipam

다음 코드 예시는 modify-ipam의 사용 방법을 보여 줍니다.

AWS CLI

IPAM 수정

다음 modify-ipam 예시에서는 운영 리전을 추가하여 IPAM을 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [ModifyIpam](https://docs.aws.amazon.com/goto/aws-cli/ec2-2016-11-15/ModifyIpam) 섹션을 참조하세요.

`modify-launch-template`

다음 코드 예시는 `modify-launch-template`의 사용 방법을 보여 줍니다.

AWS CLI

기본 시작 템플릿 버전 변경

이 예시에서는 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ModifyLaunchTemplate 섹션을 참조하세요.

modify-managed-prefix-list

다음 코드 예시는 modify-managed-prefix-list의 사용 방법을 보여 줍니다.

AWS CLI

접두사 목록 수정

다음 modify-managed-prefix-list 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyManagedPrefixList 섹션을 참조하세요.

modify-network-interface-attribute

다음 코드 예시는 modify-network-interface-attribute의 사용 방법을 보여 줍니다.

AWS CLI

네트워크 인터페이스의 연결 속성 수정

이 예시 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 속성 수정

이 예시 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 속성 수정

이 예시 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 속성 수정

이 예시 명령은 지정된 네트워크 인터페이스의 sourceDestCheck 속성을 수정합니다.

명령:

```
aws ec2 modify-network-interface-attribute --network-interface-id eni-686ea200 --no-source-dest-check
```

- API 세부 정보는 AWS CLI 명령 참조의 ModifyNetworkInterfaceAttribute 섹션을 참조하세요.

modify-private-dns-name-options

다음 코드 예시는 modify-private-dns-name-options의 사용 방법을 보여 줍니다.

AWS CLI

인스턴스 호스트 이름에 대한 옵션 설명

다음 modify-private-dns-name-options 예시에서는 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 EC2 사용 설명서의 Amazon EC2 인스턴스 호스트 이름 유형을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyPrivateDnsNameOptions 섹션을 참조하세요.

modify-reserved-instances

다음 코드 예시는 modify-reserved-instances의 사용 방법을 보여 줍니다.

AWS CLI

예약 인스턴스 수정

이 예시 명령은 예약 인스턴스를 같은 리전 내의 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 플랫폼 수정

이 예시 명령은 EC2-Classic 예약 인스턴스를 EC2-VPC로 변환합니다.

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 크기 수정

이 예시 명령은 us-west-1c에 10개의 m1.small Linux/UNIX 인스턴스가 있는 예약 인스턴스를 수정하여 8개의 m1.small 인스턴스는 2개의 m1.large 인스턴스가 되고 나머지 2개의 m1.small 인스턴스는 동일한 가용 영역에서 1개의 m1.medium 인스턴스가 되도록 합니다. 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 예약 인스턴스 크기 수정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyReservedInstances 섹션을 참조하세요.

modify-security-group-rules

다음 코드 예시는 modify-security-group-rules의 사용 방법을 보여 줍니다.

AWS CLI

보안 그룹 규칙을 수정하여 규칙 설명, IP 프로토콜 및 CidrIpv4 주소 범위 업데이트

다음 modify-security-group-rules 예시에서는 지정된 보안 그룹 규칙의 설명, IP 프로토콜 및 IPV4 CIDR 범위를 업데이트합니다. security-group-rules 파라미터를 사용하여 지정된 보안 그룹 규칙에 대한 업데이트를 입력합니다. 은 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.

- API 세부 정보는 AWS CLI 명령 참조의 ModifySecurityGroupRules 섹션을 참조하세요.

modify-snapshot-attribute

다음 코드 예시는 modify-snapshot-attribute의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 스냅샷 속성 수정

다음 modify-snapshot-attribute 예시에서는 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
```

예시 2: 스냅샷을 퍼블릭으로 설정

다음 modify-snapshot-attribute 예시에서는 지정된 스냅샷을 퍼블릭으로 설정합니다.

```
aws ec2 modify-snapshot-attribute \
    --snapshot-id snap-1234567890abcdef0 \
    --attribute createVolumePermission \
    --operation-type add \
    --group-names all
```

- API 세부 정보는 AWS CLI 명령 참조의 ModifySnapshotAttribute 섹션을 참조하세요.

modify-snapshot-tier

다음 코드 예시는 modify-snapshot-tier의 사용 방법을 보여 줍니다.

AWS CLI

스냅샷을 아카이브하려면

다음 modify-snapshot-tier 예시에서는 지정된 스냅샷을 아카이브합니다. TieringStartTime 응답 파라미터는 아카이브 프로세스가 시작된 날짜 및 시간을 UTC 시간 형식(YYYY-MM-DDTHH:MM:SSZ)으로 나타냅니다.

```
aws ec2 modify-snapshot-tier \
    --snapshot-id snap-01234567890abcedf \
    --storage-tier archive
```

출력:

```
{
    "SnapshotId": "snap-01234567890abcedf",
```

```
    "TieringStartTime": "2021-09-15T16:44:37.574Z"
}
```

스냅샷 아카이브에 대한 자세한 내용은 Amazon EBS 사용 설명서의 Amazon EBS 스냅샷 아카이브를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifySnapshotTier 섹션을 참조하세요.

modify-spot-fleet-request

다음 코드 예시는 modify-spot-fleet-request의 사용 방법을 보여 줍니다.

AWS CLI

스팟 플릿 요청 수정

이 예시 명령은 지정된 스팟 플릿 요청의 목표 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 예시 명령은 결과적으로 스팟 인스턴스를 종료하지 않고 지정된 스팟 플릿 요청의 목표 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
```

```
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [ModifySpotFleetRequest](#) 섹션을 참조하세요.

modify-subnet-attribute

다음 코드 예시는 `modify-subnet-attribute`의 사용 방법을 보여 줍니다.

AWS CLI

서브넷의 퍼블릭 IPv4 주소 지정 동작 변경

이 예시에서는 서브넷-1a2b3c4d를 수정하여 이 서브넷으로 시작된 모든 인스턴스에 공용 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 동작 변경

이 예시에서는 이 서브넷으로 시작된 모든 인스턴스에 해당 서브넷 범위의 IPv6 주소가 할당되도록 subnet-1a2b3c4d를 수정합니다.

명령:

```
aws ec2 modify-subnet-attribute --subnet-id subnet-1a2b3c4d --assign-ipv6-address-on-creation
```

자세한 내용은 AWS Virtual Private Cloud 사용 설명서에서 VPC의 IP 주소 지정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ModifySubnetAttribute](#) 섹션을 참조하세요.

modify-traffic-mirror-filter-network-services

다음 코드 예시는 `modify-traffic-mirror-filter-network-services`의 사용 방법을 보여 줍니다.

AWS CLI

트래픽 미러 필터에 네트워크 서비스 추가

다음 `modify-traffic-mirror-filter-network-services` 예시에서는 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 AWS Traffic Mirroring 설명서의 [트래픽 미러 필터 네트워크 서비스 수정]을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ModifyTrafficMirrorFilterNetworkServices] 섹션을 참조하세요.

modify-traffic-mirror-filter-rule

다음 코드 예시는 modify-traffic-mirror-filter-rule의 사용 방법을 보여 줍니다.

AWS CLI

트래픽 미러 필터 규칙 수정

다음 modify-traffic-mirror-filter-rule 예시에서는 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 AWS Traffic Mirroring 설명서의 트래픽 미러 필터 규칙 수정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyTrafficMirrorFilterRule 섹션을 참조하세요.

modify-traffic-mirror-session

다음 코드 예시는 modify-traffic-mirror-session의 사용 방법을 보여 줍니다.

AWS CLI

트래픽 미러 세션 수정

다음 modify-traffic-mirror-session 예시에서는 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 Traffic Mirroring 설명서의 트래픽 미러 세션 조정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyTrafficMirrorSession 섹션을 참조하세요.

modify-transit-gateway-prefix-list-reference

다음 코드 예시는 modify-transit-gateway-prefix-list-reference의 사용 방법을 보여 줍니다.

AWS CLI

접두사 목록 참조 수정

다음 modify-transit-gateway-prefix-list-reference 예시에서는 트래픽이 라우팅되는 연결을 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-11111122222222333 \
    --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 설명서의 접두사 목록 참조를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyTransitGatewayPrefixListReference 섹션을 참조하세요.

modify-transit-gateway-vpc-attachment

다음 코드 예시는 modify-transit-gateway-vpc-attachment의 사용 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 VPC 연결 삭제

다음 modify-transit-gateway-vpc-attachment 예시에서는 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 설명서의 VPC에 대한 전송 게이트웨이 연결을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyTransitGatewayVpcAttachment 섹션을 참조하세요.

`modify-transit-gateway`

다음 코드 예시는 `modify-transit-gateway`의 사용 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 수정

다음 `modify-transit-gateway` 예시에서는 VPN 연결에 대한 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 Gateway 설명서의 전송 게이트웨이를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyTransitGateway 섹션을 참조하세요.

modify-verified-access-endpoint-policy

다음 코드 예시는 modify-verified-access-endpoint-policy의 사용 방법을 보여 줍니다.

AWS CLI

엔드포인트에 대한 Verified Access 정책 구성

다음 modify-verified-access-endpoint-policy 예시에서는 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 AWS Verified Access 사용 설명서의 Verified Access policies 정책을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyVerifiedAccessEndpointPolicy 섹션을 참조하세요.

modify-verified-access-endpoint

다음 코드 예시는 modify-verified-access-endpoint의 사용 방법을 보여 줍니다.

AWS CLI

Verified Access 엔드포인트의 구성 수정

다음 modify-verified-access-endpoint 예시에서는 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 AWS Verified Access 사용 설명서의 Verified Access endpoints 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyVerifiedAccessEndpoint 섹션을 참조하세요.

modify-verified-access-group-policy

다음 코드 예시는 `modify-verified-access-group-policy`의 사용 방법을 보여 줍니다.

AWS CLI

그룹에 대한 Verified Access 정책 구성

다음 `modify-verified-access-group-policy` 예시에서는 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 AWS Verified Access 사용 설명서의 Verified Access groups 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyVerifiedAccessGroupPolicy 섹션을 참조하세요.

modify-verified-access-group

다음 코드 예시는 modify-verified-access-group의 사용 방법을 보여 줍니다.

AWS CLI

Verified Access 그룹의 구성 수정

다음 modify-verified-access-group 예시에서는 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 AWS Verified Access 사용 설명서의 Verified Access groups 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyVerifiedAccessGroup 섹션을 참조하세요.

`modify-verified-access-instance-logging-configuration`

다음 코드 예시는 `modify-verified-access-instance-logging-configuration`의 사용 방법을 보여 줍니다.

AWS CLI

Verified Access 인스턴스에 대한 로깅 활성화

다음 `modify-verified-access-instance-logging-configuration` 예시에서는 지정된 Verified Access 인스턴스에 대한 액세스 로깅을 활성화합니다. 로그는 지정된 CloudWatch Logs 로그 그룹으로 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 AWS Verified Access 사용 설명서의 Verified Access logs 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyVerifiedAccessInstanceLoggingConfiguration 섹션을 참조하세요.

modify-verified-access-instance

다음 코드 예시는 modify-verified-access-instance의 사용 방법을 보여 줍니다.

AWS CLI

Verified Access 인스탄스의 구성 수정

다음 modify-verified-access-instance 예시에서는 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 AWS Verified Access 사용 설명서의 Verified Access instances 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyVerifiedAccessInstance 섹션을 참조하세요.

modify-verified-access-trust-provider

다음 코드 예시는 modify-verified-access-trust-provider의 사용 방법을 보여 줍니다.

AWS CLI

Verified·Access 디바이스 신뢰 공급자 구성 수정

다음 modify-verified-access-trust-provider 예시에서는 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 AWS Verified Access 사용 설명서의 Trust providers for Verified Access 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyVerifiedAccessTrustProvider 섹션을 참조하세요.

modify-volume-attribute

다음 코드 예시는 modify-volume-attribute의 사용 방법을 보여 줍니다.

AWS CLI

볼륨 속성 수정

이 예시에서는 ID가 vol-1234567890abcdef0인 볼륨의 autoEnableIo 속성을 true로 설정합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 modify-volume-attribute --volume-id vol-1234567890abcdef0 --auto-enable-io
```

- API 세부 정보는 AWS CLI 명령 참조의 ModifyVolumeAttribute 섹션을 참조하세요.

modify-volume

다음 코드 예시는 modify-volume의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 크기를 변경하여 볼륨 수정

다음 modify-volume 예시에서는 지정된 볼륨의 크기를 150GB로 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
    }
}
```

예시 2: 유형, 크기 및 IOPS 값을 변경하여 볼륨 수정

다음 `modify-volume` 예시에서는 볼륨 유형을 프로비저닝된 IOPS SSD로 변경하고, 목표 IOPS 속도를 10,000으로 설정하고, 볼륨 크기를 350GB로 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
```

```
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ModifyVolume 섹션을 참조하세요.

modify-vpc-attribute

다음 코드 예시는 modify-vpc-attribute의 사용 방법을 보여 줍니다.

AWS CLI

enableDnsSupport 속성 수정

다음 예시에서는 enableDnsSupport 속성을 수정합니다. 이 속성은 VPC에 DNS 확인이 활성화되어 있는지 여부를 나타냅니다. 이 속성이 true인 경우 Amazon DNS 서버는 인스턴스의 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 속성 수정

다음 예시에서는 enableDnsHostnames 속성을 수정합니다. 이 속성은 VPC에서 시작된 인스턴스가 DNS 호스트 이름을 가져오는지 나타냅니다. 이 속성이 true인 경우 VPC의 인스턴스가 DNS 호스트 이름을 가져오고, 그렇지 않으면 가져오지 않습니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 modify-vpc-attribute --vpc-id vpc-a01106c2 --enable-dns-hostnames "{\"Value
\":false}"
```

- API 세부 정보는 AWS CLI 명령 참조의 ModifyVpcAttribute 섹션을 참조하세요.

modify-vpc-endpoint-connection-notification

다음 코드 예시는 modify-vpc-endpoint-connection-notification의 사용 방법을 보여 줍니다.

AWS CLI

엔드포인트 연결 알림 수정

이 예시에서는 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ModifyVpcEndpointConnectionNotification 섹션을 참조하세요.

modify-vpc-endpoint-service-configuration

다음 코드 예시는 modify-vpc-endpoint-service-configuration의 사용 방법을 보여 줍니다.

AWS CLI

엔드포인트 서비스 구성 수정

이 예시에서는 지정된 엔드포인트 서비스에 대한 허용 요구 사항을 변경합니다.

명령:

```
aws ec2 modify-vpc-endpoint-service-configuration --service-id vpce-svc-09222513e6e77dc86 --no-acceptance-required
```

출력:

```
{
   "ReturnValue": true
```

```
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ModifyVpcEndpointServiceConfiguration 섹션을 참조하세요.

modify-vpc-endpoint-service-payer-responsibility

다음 코드 예시는 `modify-vpc-endpoint-service-payer-responsibility`의 사용 방법을 보여 줍니다.

AWS CLI

지불자 책임 수정

다음 `modify-vpc-endpoint-service-payer-responsibility` 예시에서는 지정된 엔드포인트 서비스의 지급인 책임을 수정합니다.

```
aws ec2 modify-vpc-endpoint-service-payer-responsibility \
    --service-id vpce-svc-071afff70666e61e0 \
    --payer-responsibility ServiceOwner
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyVpcEndpointServicePayerResponsibility 섹션을 참조하세요.

modify-vpc-endpoint-service-permissions

다음 코드 예시는 `modify-vpc-endpoint-service-permissions`의 사용 방법을 보여 줍니다.

AWS CLI

엔드포인트 서비스 권한 수정

이 예시에서는 지정된 엔드포인트 서비스에 연결할 수 있는 AWS 계정에 대한 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 예시에서는 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"]'
```

- API 세부 정보는 AWS CLI 명령 참조의 ModifyVpcEndpointServicePermissions 섹션을 참조하세요.

modify-vpc-endpoint

다음 코드 예시는 modify-vpc-endpoint의 사용 방법을 보여 줍니다.

AWS CLI

게이트웨이 엔드포인트 수정

이 예시에서는 rtb-aaa222bb 라우팅 테이블을 vpce-1a2b3c4d 엔드포인트와 연결하고 정책 문서를 재설정하여 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 엔드포인트 수정

이 예시에서는 `vpce-0fe5b17a0707d6fa5` 엔드포인트에 `subnet-d6fcaa8d` 서브넷을 추가하여 인터페이스 엔드포인트를 수정합니다.

명령:

```
aws ec2 modify-vpc-endpoint --vpc-endpoint-id vpce-0fe5b17a0707d6fa5 --add-subnet-id subnet-d6fcaa8d
```

출력:

```
{
  "Return": true
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ModifyVpcEndpoint 섹션을 참조하세요.

`modify-vpc-peering-connection-options`

다음 코드 예시는 `modify-vpc-peering-connection-options`의 사용 방법을 보여 줍니다.

AWS CLI

로컬 ClassicLink 연결에서 VPC 피어링 연결을 통한 통신 활성

이 예시에서는 피어링 연결 `pcx-aaaabbb`의 경우 요청자 VPC의 소유자가 VPC 피어링 연결 옵션을 수정하여 로컬 ClassicLink 연결이 피어 VPC와 통신할 수 있도록 합니다.

명령:

```
aws ec2 modify-vpc-peering-connection-options --vpc-peering-connection-id pcx-aaaabbbb --requester-peering-connection-options AllowEgressFromLocalClassicLinkToRemoteVpc=true
```

출력:

```
{
  "RequesterPeeringConnectionOptions": {
      "AllowEgressFromLocalClassicLinkToRemoteVpc": true
```

```
  }
}
```

로컬 VPC에서 원격 ClassicLink 연결로의 VPC 피어링 연결을 통한 통신 활성화

이 예시에서는 수락자 VPC의 소유자가 VPC 피어링 연결 옵션을 수정하여 로컬 VPC가 피어 VPC의 ClassicLink 연결과 통신할 수 있도록 합니다.

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 확인 지원 활성화

이 예시에서는 요청자 VPC의 소유자가 pcx-aaaabbbb에 대한 VPC 피어링 연결 옵션을 수정하여 피어 VPC의 인스턴스에서 쿼리할 때 로컬 VPC가 퍼블릭 DNS 호스트 이름을 프라이빗 IP 주소로 확인할 수 있도록 합니다.

명령:

```
aws ec2 modify-vpc-peering-connection-options --vpc-peering-connection-id pcx-
aaaabbbb --requester-peering-connection-options AllowDnsResolutionFromRemoteVpc=true
```

출력:

```
{
  "RequesterPeeringConnectionOptions": {
      "AllowDnsResolutionFromRemoteVpc": true
  }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ModifyVpcPeeringConnectionOptions 섹션을 참조하세요.

modify-vpc-tenancy

다음 코드 예시는 modify-vpc-tenancy의 사용 방법을 보여 줍니다.

AWS CLI

VPC의 테넌시 수정

이 예시에서는 VPC vpc-1a2b3c4d의 테넌시를 default로 수정합니다.

명령:

```
aws ec2 modify-vpc-tenancy --vpc-id vpc-1a2b3c4d --instance-tenancy default
```

출력:

```
{
  "Return": true
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ModifyVpcTenancy 섹션을 참조하세요.

modify-vpn-connection-options

다음 코드 예시는 modify-vpn-connection-options의 사용 방법을 보여 줍니다.

AWS CLI

VPN 연결 옵션 수정

다음 modify-vpn-connection-options 예시에서는 지정된 VPN 연결의 고객 게이트웨이 측에서 로컬 IPv4 CIDR을 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 Site-to-Site VPN 연결 옵션 수정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyVpnConnectionOptions을 참조하세요.

`modify-vpn-connection`

다음 코드 예시는 `modify-vpn-connection`의 사용 방법을 보여 줍니다.

AWS CLI

VPN 연결 수정

다음 `modify-vpn-connection` 예시에서는 VPN 연결 `vpn-12345678901234567`의 대상 게이트웨이를 가상 프라이빗 게이트웨이 `vgw-11223344556677889`로 변경합니다.

```
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"
            }
        ]
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ModifyVpnConnection을 참조하세요.

modify-vpn-tunnel-certificate

다음 코드 예시는 modify-vpn-tunnel-certificate의 사용 방법을 보여 줍니다.

AWS CLI

VPN 터널 인증서 교체

다음 modify-vpn-tunnel-certificate 예시에서는 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"
            }
        ]
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ModifyVpnTunnelCertificate을 참조하세요.

`modify-vpn-tunnel-options`

다음 코드 예시는 `modify-vpn-tunnel-options`의 사용 방법을 보여 줍니다.

AWS CLI

VPN 연결에 대한 터널 옵션 수정

다음 `modify-vpn-tunnel-options` 예시에서는 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"
            }
        ]
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [ModifyVpnTunnelOptions](#)을 참조하세요.

`monitor-instances`

다음 코드 예시는 `monitor-instances`의 사용 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"
          }
      }
  ]
}
```

- API 세부 정보는 AWS CLI 명령 참조에서 MonitorInstances를 참조하세요.

move-address-to-vpc

다음 코드 예시는 move-address-to-vpc의 사용 방법을 보여 줍니다.

AWS CLI

주소를 EC2-VPC로 이동

이 예시에서는 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 MoveAddressToVpc 섹션을 참조하세요.

move-byoip-cidr-to-ipam

다음 코드 예시는 move-byoip-cidr-to-ipam의 사용 방법을 보여 줍니다.

AWS CLI

IPAM으로 BYOIP CIDR 전송

다음 move-byoip-cidr-to-ipam 예시에서는 BYOIP CIDR을 IPAM으로 전송합니다.

(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 BYOIP IPv4 CIDR을 IPAM으로 전송을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 MoveByoipCidrToIpam을 참조하세요.

network-insights-access-scope

다음 코드 예시는 network-insights-access-scope의 사용 방법을 보여 줍니다.

AWS CLI

Network Insights 액세스 범위 생성

다음 create-network-insights-access-scope 예시에서는 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 설명서의 AWS CLI를 사용하여 Network Access Analyzer 시작하기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 NetworkInsightsAccessScope 섹션을 참조하세요.

provision-byoip-cidr

다음 코드 예시는 provision-byoip-cidr의 사용 방법을 보여 줍니다.

AWS CLI

주소 범위 프로비저닝

다음 provision-byoip-cidr 예시에서는 AWS에서 사용할 퍼블릭 IP 주소 범위를 프로비저닝합니다.

```
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 문자열을 만드는 방법에 대한 자세한 내용은 Amazon EC2 사용 설명서의 자체 IP 주소 가져오기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ProvisionByoipCidr 섹션을 참조하세요.

provision-ipam-pool-cidr

다음 코드 예시는 provision-ipam-pool-cidr의 사용 방법을 보여 줍니다.

AWS CLI

IPAM 풀에 CIDR 프로비저닝

다음 `provision-ipam-pool-cidr` 예시에서는 CIDR을 IPAM 풀에 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 풀에 CIDR 프로비저닝을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ProvisionIpamPoolCidr을 참조하세요.

`purchase-host-reservation`

다음 코드 예시는 `purchase-host-reservation`의 사용 방법을 보여 줍니다.

AWS CLI

전용 호스트 예약 구매

이 예시에서는 계정에서 지정한 전용 호스트에 대해 지정된 전용 호스트 예약 상품을 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 PurchaseHostReservation 섹션을 참조하세요.

purchase-reserved-instances-offering

다음 코드 예시는 purchase-reserved-instances-offering의 사용 방법을 보여 줍니다.

AWS CLI

예약 인스턴스 오퍼링 구매

이 예시 명령은 예약 인스턴스 오퍼링의 구매를 설명하며, 오퍼링 ID와 인스턴스 수를 지정합니다.

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"
```

```
}
```

- API 세부 정보는 AWS CLI 명령 참조의 PurchaseReservedInstancesOffering 섹션을 참조하세요.

`purchase-scheduled-instances`

다음 코드 예시는 `purchase-scheduled-instances`의 사용 방법을 보여 줍니다.

AWS CLI

정기 인스턴스 구매

이 예시에서는 정기 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 PurchaseScheduledInstances을 참조하세요.

reboot-instances

다음 코드 예시는 `reboot-instances`의 사용 방법을 보여 줍니다.

AWS CLI

Amazon EC2 인스턴스를 재부팅하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조에서 RebootInstances를 참조하세요.

register-image

다음 코드 예시는 `register-image`의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 매니페스트 파일을 사용하여 AMI 등록

다음 `register-image` 예시에서는 Amazon S3에서 지정된 매니페스트 파일을 사용하여 AMI를 등록합니다.

```
aws ec2 register-image \
    --name my-image \
    --image-location amzn-s3-demo-bucket/myimage/image.manifest.xml
```

출력:

```
{
    "ImageId": "ami-1234567890EXAMPLE"
}
```

자세한 내용은 Amazon EC2 사용 설명서에서 [Amazon Machine Image(AMI)](#)를 참조하세요.

예시 2: 루트 디바이스의 스냅샷을 사용하여 AMI 등록

다음 `register-image` 예시에서는 EBS 루트 볼륨의 지정된 스냅샷을 디바이스 `/dev/xvda`로 사용하여 AMI를 등록합니다. 블록 디바이스 매핑에는 빈 100기가바이트 EBS 볼륨도 디바이스 `/dev/xvdf`로 포함됩니다.

```
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 EC2 사용 설명서에서 [Amazon Machine Image(AMI)](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [RegisterImage](#) 섹션을 참조하세요.

register-instance-event-notification-attributes

다음 코드 예시는 register-instance-event-notification-attributes의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 이벤트 알림의 모든 태그 포함

다음 register-instance-event-notification-attributes 예시에서는 이벤트 알림의 모든 태그를 포함합니다.

```
aws ec2 register-instance-event-notification-attributes \
    --instance-tag-attribute IncludeAllTagsOfInstance=true
```

출력:

```
{
    "InstanceTagAttribute": {
        "InstanceTagKeys": [],
        "IncludeAllTagsOfInstance": true
    }
}
```

자세한 내용은 Amazon EC2 사용 설명서의 예약된 인스턴스 이벤트를 참조하세요.

예시 2: 이벤트 알림에 특정 태그 포함

다음 register-instance-event-notification-attributes 예시에서는 이벤트 알림에 지정된 태그를 포함합니다. IncludeAllTagsOfInstance가 true인 경우 태그를 지정할 수 없습니다.

```
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 Amazon EC2 사용 설명서의 [예약된 인스턴스 이벤트]를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [RegisterInstanceEventNotificationAttributes] 섹션을 참조하세요.

`register-transit-gateway-multicase-group-sources`

다음 코드 예시는 `register-transit-gateway-multicase-group-sources`의 사용 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 멀티캐스트 그룹에 소스 등록

다음 `register-transit-gateway-multicast-group-sources` 예시에서는 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 AWS Transit Gateways 사용 설명서의 [멀티캐스트 그룹에 소스 등록]을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [RegisterTransitGatewayMulticaseGroupSources](#) 섹션을 참조하세요.

register-transit-gateway-multicast-group-members

다음 코드 예시는 `register-transit-gateway-multicast-group-members`의 사용 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 멀티캐스트 도메인에 대한 연결 정보 보기

다음 `register-transit-gateway-multicast-group-members` 예시에서는 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](#)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [RegisterTransitGatewayMulticastGroupMembers](#) 섹션을 참조하세요.

register-transit-gateway-multicast-group-sources

다음 코드 예시는 `register-transit-gateway-multicast-group-sources`의 사용 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 멀티캐스트 그룹에 소스 등록

다음 register-transit-gateway-multicast-group-sources 예시에서는 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 설명서의 멀티캐스트 도메인 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RegisterTransitGatewayMulticastGroupSources 섹션을 참조하세요.

reject-transit-gateway-peering-attachment

다음 코드 예시는 reject-transit-gateway-peering-attachment의 사용 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 피어링 연결 거부

다음 reject-transit-gateway-peering-attachment 예시에서는 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 설명서의 Transit Gateway Peering Attachments 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RejectTransitGatewayPeeringAttachment 섹션을 참조하세요.

`reject-transit-gateway-vpc-attachment`

다음 코드 예시는 `reject-transit-gateway-vpc-attachment`의 사용 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 VPC 연결 거부

다음 `reject-transit-gateway-vpc-attachment` 예시에서는 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 설명서의 VPC에 대한 전송 게이트웨이 연결을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RejectTransitGatewayVpcAttachment 섹션을 참조하세요.

reject-transit-gateway-vpc-attachments

다음 코드 예시는 `reject-transit-gateway-vpc-attachments`의 사용 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 VPC 연결 거부

다음 `reject-transit-gateway-vpc-attachment` 예시에서는 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 설명서의 VPC에 대한 전송 게이트웨이 연결을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RejectTransitGatewayVpcAttachments 섹션을 참조하세요.

reject-vpc-endpoint-connections

다음 코드 예시는 reject-vpc-endpoint-connections의 사용 방법을 보여 줍니다.

AWS CLI

인터페이스 엔드포인트 연결 요청 거부

이 예시에서는 지정된 엔드포인트 서비스에 대해 지정된 엔드포인트 연결 요청을 거부합니다.

명령:

```
aws ec2 reject-vpc-endpoint-connections --service-id vpce-svc-03d5ebb7d9579a2b3 --
vpc-endpoint-ids vpce-0c1308d7312217abc
```

출력:

```
{
  "Unsuccessful": []
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [RejectVpcEndpointConnections](#) 섹션을 참조하세요.

`reject-vpc-peering-connection`

다음 코드 예시는 `reject-vpc-peering-connection`의 사용 방법을 보여 줍니다.

AWS CLI

VPC 피어링 연결 거부

이 예시에서는 지정된 VPC 피어링 연결 요청을 거부합니다.

명령:

```
aws ec2 reject-vpc-peering-connection --vpc-peering-connection-id pcx-1a2b3c4d
```

출력:

```
{
    "Return": true
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [RejectVpcPeeringConnection](#) 섹션을 참조하세요.

`release-address`

다음 코드 예시는 `release-address`의 사용 방법을 보여 줍니다.

AWS CLI

EC2-Classic의 탄력적 IP 주소를 해제하는 방법

자세한 내용은 EC2-Classic의 인스턴스에서 사용할 탄력적 IP 주소를 해제합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 release-address --public-ip 198.51.100.0
```

EC2-VPC의 탄력적 IP 주소를 해제하는 방법

이 예제에서는 VPC의 인스턴스에서 사용하도록 탄력적 IP 주소를 해제합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 release-address --allocation-id eipalloc-64d5890a
```

- API 세부 정보는 AWS CLI 명령 참조의 ReleaseAddress를 참조하세요.

release-hosts

다음 코드 예시는 `release-hosts`의 사용 방법을 보여 줍니다.

AWS CLI

계정에서 전용 호스트 해제

계정에서 전용 호스트를 해제합니다. 호스트에 있는 인스턴스는 호스트를 해제하기 전에 중지하거나 종료해야 합니다.

명령:

```
aws ec2 release-hosts --host-id=h-0029d6e3cacf1b3da
```

출력:

```
{
    "Successful":  [
        "h-0029d6e3cacf1b3da"
         ],
  "Unsuccessful": []

 }
```

- API 세부 정보는 AWS CLI 명령 참조의 ReleaseHosts 섹션을 참조하세요.

release-ipam-pool-allocation

다음 코드 예시는 `release-ipam-pool-allocation`의 사용 방법을 보여 줍니다.

AWS CLI

IPAM 풀 할당 해제

이 예시에서는 IPAM 풀을 삭제하려고 했지만 풀에 할당이 있는 동안에는 풀을 삭제할 수 없다는 오류가 표시되는 IPAM 위임된 관리자입니다. 이 명령을 사용하여 풀 할당을 해제합니다.

다음 사항에 유의하세요.

이 명령은 사용자 지정 할당의 경우에만 사용할 수 있습니다. 리소스를 삭제하지 않고 리소스에 대한 할당을 제거하려면 modify-ipam-resource-cidr 를 사용하여 모니터링된 상태를 false로 설정합니다. 이 요청을 완료하려면 describe-ipam-pools 로 가져올 수 있는 IPAM 풀 ID가 필요합니다. 또한 할당 ID가 필요하며, 이는 get-ipam-pool-allocations로 얻을 수 있습니다.할당을 하나씩 제거하지 않으려면 IPAM 풀을 삭제할 때 `--cascade option`을 사용하여 풀의 모든 할당을 삭제하기 전에 자동으로 해제할 수 있습니다.이 명령을 실행하기 전에 여러 가지 전제 조건이 있습니다. 자세한 내용은 Amazon VPC IPAM 사용 설명서의 할당 해제를 참조하세요. 이 명령을 실행하는 `--region`은 할당이 있는 IPAM 풀의 로캘이어야 합니다.

다음 `release-ipam-pool-allocation` 예시에서는 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 delete-ipam-pool을 실행할 수 있습니다.

- API 세부 정보는 AWS CLI 명령 참조의 ReleaseIpamPoolAllocation 섹션을 참조하세요.

`replace-iam-instance-profile-association`

다음 코드 예시는 `replace-iam-instance-profile-association`의 사용 방법을 보여 줍니다.

AWS CLI

인스턴스에 대한 IAM 인스턴스 프로파일을 바꾸는 방법

이 예제에서는 iip-assoc-060bae234aac2e7fa 연결로 표시되는 IAM 인스턴스 프로파일을 이름이 AdminRole인 IAM 인스턴스 프로파일로 바꿉니다.

```
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"
        }
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조에서 ReplaceIamInstanceProfileAssociation을 참조하세요.

replace-network-acl-association

다음 코드 예시는 replace-network-acl-association의 사용 방법을 보여 줍니다.

AWS CLI

서브넷과 연결된 네트워크 ACL 교체

이 예시에서는 지정한 네트워크 ACL을 지정한 네트워크 ACL 연결의 서브넷과 연결합니다.

명령:

```
aws ec2 replace-network-acl-association --association-id aclassoc-e5b95c8c --
network-acl-id acl-5fb85d36
```

출력:

```
{
    "NewAssociationId": "aclassoc-3999875b"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [ReplaceNetworkAclAssociation](#) 섹션을 참조하세요.

`replace-network-acl-entry`

다음 코드 예시는 `replace-network-acl-entry`의 사용 방법을 보여 줍니다.

AWS CLI

네트워크 ACL 항목 교체

이 예시에서는 지정된 네트워크 ACL에 대한 항목을 대체합니다. 새 규칙 100은 UDP 포트 53(DNS)의 203.0.113.12/24에서 연결된 모든 서브넷으로 유입되는 트래픽을 허용합니다.

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
```

- API 세부 정보는 AWS CLI 명령 참조의 [ReplaceNetworkAclEntry](#) 섹션을 참조하세요.

`replace-route-table-association`

다음 코드 예시는 `replace-route-table-association`의 사용 방법을 보여 줍니다.

AWS CLI

서브넷과 연결된 라우팅 테이블 교체

이 예시에서는 지정한 라우팅 테이블을 지정한 라우팅 테이블 연결을 위한 서브넷과 연결합니다.

명령:

```
aws ec2 replace-route-table-association --association-id rtbassoc-781d0d1a --route-table-id rtb-22574640
```

출력:

```
{
    "NewAssociationId": "rtbassoc-3a1f0f58"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ReplaceRouteTableAssociation 섹션을 참조하세요.

replace-route

다음 코드 예시는 replace-route의 사용 방법을 보여 줍니다.

AWS CLI

경로를 바꾸는 방법

이 예시에서는 지정된 라우팅 테이블에서 지정된 경로를 대체합니다. 새 라우팅은 지정된 CIDR과 일치하며 트래픽을 지정된 가상 사설 게이트웨이로 보냅니다. 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
```

- API 세부 정보는 AWS CLI 명령 참조의 ReplaceRoute 섹션을 참조하세요.

replace-transit-gateway-route

다음 코드 예시는 replace-transit-gateway-route의 사용 방법을 보여 줍니다.

AWS CLI

지정된 전송 게이트웨이 라우팅 테이블에서 지정된 라우팅 교체

다음 replace-transit-gateway-route 예시에서는 지정된 전송 게이트웨이 라우팅 테이블의 경로를 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 설명서의 전송 게이트웨이 라우팅 테이블을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ReplaceTransitGatewayRoute 섹션을 참조하세요.

report-instance-status

다음 코드 예시는 `report-instance-status`의 사용 방법을 보여 줍니다.

AWS CLI

인스턴스에 대한 상태 피드백 보고

이 예시 명령은 지정된 인스턴스에 대한 상태 피드백을 보고합니다.

명령:

```
aws ec2 report-instance-status --instances i-1234567890abcdef0 --status impaired --
reason-codes unresponsive
```

- API 세부 정보는 AWS CLI 명령 참조의 ReportInstanceStatus 섹션을 참조하세요.

`request-spot-fleet`

다음 코드 예시는 `request-spot-fleet`의 사용 방법을 보여 줍니다.

AWS CLI

서브넷에서 가장 저렴한 가격으로 스팟 플릿 요청

이 예시 명령은 서브넷만 다른 두 가지 출시 사양을 가진 스팟 플릿 요청을 생성합니다. 스팟 플릿은 지정된 서브넷에서 가장 낮은 가격으로 인스턴스를 시작합니다. 인스턴스가 기본 VPC로 시작되는 경우, 인스턴스는 기본적으로 퍼블릭 IP 주소를 받습니다. 인스턴스가 기본이 아닌 VPC로 시작되는 경우, 인스턴스는 기본적으로 퍼블릭 IP 주소를 받지 않습니다.

스팟 플릿 요청에서는 동일한 가용 영역에서 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 플릿 요청

이 예시 명령은 가용 영역만 다른 두 가지 출시 사양을 가진 스팟 플릿 요청을 생성합니다. 스팟 플릿은 지정된 가용 영역에서 가장 낮은 가격으로 인스턴스를 시작합니다. 계정이 EC2-VPC만 지원하는 경우, Amazon EC2는 가용 영역의 기본 서브넷에서 스팟 인스턴스를 실행합니다. 계정이 EC2-Classic을 지원하는 경우, Amazon EC2는 가용 영역의 EC2-Classic에서 인스턴스를 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 주소 할당

이 예시 명령은 기본값이 아닌 VPC에서 시작된 인스턴스에 퍼블릭 주소를 할당합니다. 네트워크 인터페이스를 지정할 때는 네트워크 인터페이스를 사용하는 서브넷 ID 및 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"
          }
      }
  ]
}
```

다양한 할당 전략을 사용하여 스팟 플릿 요청

이 예시 명령은 다양한 할당 전략을 사용하여 30개의 인스턴스를 실행하는 스팟 플릿 요청을 생성합니다. 시작 사양은 인스턴스 유형에 따라 다릅니다. 스팟 플릿은 각 유형에 10개의 인스턴스가 있도록 출시 사양에 따라 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.

- API 세부 정보는 AWS CLI 명령 참조의 [RequestSpotFleet](#) 섹션을 참조하세요.

`request-spot-instances`

다음 코드 예시는 `request-spot-instances`의 사용 방법을 보여 줍니다.

AWS CLI

스팟 인스턴스 요청

이 예시 명령은 지정된 가용 영역에서 5개의 인스턴스에 대한 일회성 스팟 인스턴스 요청을 생성합니다. 계정이 EC2-VPC만 지원하는 경우, Amazon EC2는 지정된 가용 영역의 기본 서브넷에서 인스턴스를 실행합니다. 계정이 EC2-Classic을 지원하는 경우, Amazon EC2는 지정된 가용 영역에서 EC2-Classic의 인스턴스를 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 예시 명령은 지정된 서브넷에서 5개의 인스턴스에 대한 일회성 스팟 인스턴스 요청을 생성합니다. Amazon EC2는 지정된 서브넷에서 인스턴스를 시작합니다. VPC가 기본값이 아닌 VPC인 경우 인스턴스는 기본적으로 퍼블릭 IP 주소를 받지 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 예시에서는 기본값이 아닌 VPC에서 실행하는 스팟 인스턴스에 퍼블릭 IP 주소를 할당합니다. 네트워크 인터페이스를 지정할 때는 네트워크 인터페이스를 사용하는 서브넷 ID 및 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"
  }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 RequestSpotInstances 섹션을 참조하세요.

`reset-address-attribute`

다음 코드 예시는 `reset-address-attribute`의 사용 방법을 보여 줍니다.

AWS CLI

탄력적 IP 주소와 연결된 도메인 이름 속성 재설정

다음 `reset-address-attribute` 예시에서는 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 AWS CLI 명령 참조의 describe-addresses-attribute를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ResetAddressAttribute을 참조하세요.

`reset-ebs-default-kms-key-id`

다음 코드 예시는 `reset-ebs-default-kms-key-id`의 사용 방법을 보여 줍니다.

AWS CLI

EBS 암호화를 위한 기본 CMK 재설정

다음 `reset-ebs-default-kms-key-id` 예시에서는 현재 리전 내 AWS 계정의 EBS 암호화에 대한 기본 CMK를 재설정합니다.

```
aws ec2 reset-ebs-default-kms-key-id
```

출력:

```
{
   "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/8c5b2c63-b9bc-45a3-a87a-5513eEXAMPLE"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ResetEbsDefaultKmsKeyId 섹션을 참조하세요.

`reset-fpga-image-attribute`

다음 코드 예시는 `reset-fpga-image-attribute`의 사용 방법을 보여 줍니다.

AWS CLI

Amazon FPGA 이미지의 속성 재설정

이 예시에서는 지정된 AFI에 대한 로드 권한을 재설정합니다.

명령:

```
aws ec2 reset-fpga-image-attribute --fpga-image-id afi-0d123e123bfc85abc --attribute loadPermission
```

출력:

```
{
  "Return": true
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ResetFpgaImageAttribute 섹션을 참조하세요.

reset-image-attribute

다음 코드 예시는 reset-image-attribute의 사용 방법을 보여 줍니다.

AWS CLI

launchPermission 속성 재설정

이 예시에서는 지정된 AMI의 launchPermission 속성을 기본값으로 재설정합니다. 기본적으로 AMI는 프라이빗으로 설정됩니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 reset-image-attribute --image-id ami-5731123e --attribute launchPermission
```

- API 세부 정보는 AWS CLI 명령 참조의 [ResetImageAttribute](#) 섹션을 참조하세요.

reset-instance-attribute

다음 코드 예시는 reset-instance-attribute의 사용 방법을 보여 줍니다.

AWS CLI

sourceDestCheck 속성 재설정

이 예시에서는 지정된 인스턴스의 sourceDestCheck 속성을 재설정합니다. 인스턴스가 VPC에 있어야 합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 reset-instance-attribute --instance-id i-1234567890abcdef0 --
attribute sourceDestCheck
```

커널 속성을 재설정하는 방법

이 예시에서는 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 속성 재설정

이 예시에서는 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
```

- API 세부 정보는 AWS CLI 명령 참조의 ResetInstanceAttribute 섹션을 참조하세요.

reset-network-interface-attribute

다음 코드 예시는 reset-network-interface-attribute의 사용 방법을 보여 줍니다.

AWS CLI

네트워크 인터페이스 속성 재설정

다음 reset-network-interface-attribute 예시에서는 소스/대상 검사 속성의 값을 true로 재설정합니다.

```
aws ec2 reset-network-interface-attribute \
    --network-interface-id eni-686ea200 \
    --source-dest-check
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 ResetNetworkInterfaceAttribute 섹션을 참조하세요.

reset-snapshot-attribute

다음 코드 예시는 reset-snapshot-attribute의 사용 방법을 보여 줍니다.

AWS CLI

스냅샷 속성 재설정

이 예시에서는 스냅샷 snap-1234567890abcdef0에 대한 볼륨 생성 권한을 초기화합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 reset-snapshot-attribute --snapshot-id snap-1234567890abcdef0 --
attribute createVolumePermission
```

- API 세부 정보는 AWS CLI 명령 참조의 ResetSnapshotAttribute 섹션을 참조하세요.

restore-address-to-classic

다음 코드 예시는 `restore-address-to-classic`의 사용 방법을 보여 줍니다.

AWS CLI

EC2-Classic으로 주소 복원

이 예시에서는 탄력적 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 RestoreAddressToClassic 섹션을 참조하세요.

restore-image-from-recycle-bin

다음 코드 예시는 `restore-image-from-recycle-bin`의 사용 방법을 보여 줍니다.

AWS CLI

휴지통에서 이미지 복원

다음 `restore-image-from-recycle-bin` 예시에서는 휴지통에서 AMI ami-0111222333444abcd를 복원합니다.

```
aws ec2 restore-image-from-recycle-bin \
```

```
    --image-id ami-0111222333444abcd
```

출력:

```
{
    "Return": true
}
```

자세한 내용은 Amazon EBS 사용 설명서의 휴지통에서 삭제된 AMI 복구를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RestoreImageFromRecycleBin 섹션을 참조하세요.

restore-managed-prefix-list-version

다음 코드 예시는 restore-managed-prefix-list-version의 사용 방법을 보여 줍니다.

AWS CLI

us-west-2**접두사 목록 버전을 복원하는 방법**

다음 restore-managed-prefix-list-version 명령은 지정된 접두사 목록의 버전 1에서 항목을 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.

- API 세부 정보는 AWS CLI 명령 참조의 RestoreManagedPrefixListVersion 섹션을 참조하세요.

restore-snapshot-from-recycle-bin

다음 코드 예시는 restore-snapshot-from-recycle-bin의 사용 방법을 보여 줍니다.

AWS CLI

휴지통에서 스냅샷 복원

다음 restore-snapshot-from-recycle-bin 명령은 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.

휴지통에 대한 자세한 내용은 Amazon EBS 사용 설명서의 휴지통에서 삭제된 스냅샷 복구를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RestoreSnapshotFromRecycleBin 섹션을 참조하세요.

restore-snapshot-tier

다음 코드 예시는 restore-snapshot-tier의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 아카이빙된 스냅샷 영구 복원

다음 restore-snapshot-tier 예시에서는 지정된 스냅샷을 영구적으로 복원합니다. --snapshot-id를 지정하고 permanent-restore 옵션을 포함합니다.

```
aws ec2 restore-snapshot-tier \
    --snapshot-id snap-01234567890abcedf \
    --permanent-restore
```

출력:

```
{
    "SnapshotId": "snap-01234567890abcedf",
    "IsPermanentRestore": true
}
```

스냅샷 아카이브에 대한 자세한 내용은 Amazon EBS 사용 설명서의 Amazon EBS 스냅샷 아카이브를 참조하세요.

예시 2: 아카이빙된 스냅샷 임시 복원

다음 `restore-snapshot-tier` 예시에서는 지정된 스냅샷을 일시적으로 복원합니다. `--permanent-restore` 옵션을 생략합니다. `--snapshot-id`를 지정하고 `temporary-restore-days`의 경우 스냅샷을 복원할 일 수를 지정합니다. `temporary-restore-days`는 일 단위로 지정해야 합니다. 허용되는 범위는 `1`~`180`입니다. 값을 지정하지 않으면 기본적으로 `1`일이 사용됩니다.

```
aws ec2 restore-snapshot-tier \
    --snapshot-id snap-01234567890abcedf \
    --temporary-restore-days 5
```

출력:

```
{
    "SnapshotId": "snap-01234567890abcedf",
    "RestoreDuration": 5,
    "IsPermanentRestore": false
}
```

스냅샷 아카이브에 대한 자세한 내용은 Amazon EBS 사용 설명서의 Amazon EBS 스냅샷 아카이브를 참조하세요.

예시 3: 복원 기간 수정

다음 `restore-snapshot-tier` 예시에서는 지정한 스냅샷의 복원 기간을 `10`일로 변경합니다.

```
aws ec2 restore-snapshot-tier \
    --snapshot-id snap-01234567890abcedf \
    --temporary-restore-days 10
```

출력:

```
{
    "SnapshotId": "snap-01234567890abcedf",
    "RestoreDuration": 10,
    "IsPermanentRestore": false
}
```

스냅샷 아카이브에 대한 자세한 내용은 Amazon EBS 사용 설명서의 Amazon EBS 스냅샷 아카이브를 참조하세요.

예시 4: 복원 유형 수정

다음 `restore-snapshot-tier` 예시에서는 지정한 스냅샷의 복원 유형을 임시에서 영구로 변경합니다.

```
aws ec2 restore-snapshot-tier \
    --snapshot-id snap-01234567890abcedf
    --permanent-restore
```

출력:

```
{
    "SnapshotId": "snap-01234567890abcedf",
    "IsPermanentRestore": true
}
```

스냅샷 아카이브에 대한 자세한 내용은 Amazon EBS 사용 설명서의 Amazon EBS 스냅샷 아카이브를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RestoreSnapshotTier 섹션을 참조하세요.

`revoke-client-vpn-ingress`

다음 코드 예시는 `revoke-client-vpn-ingress`의 사용 방법을 보여 줍니다.

AWS CLI

Client VPN 엔드포인트에 권한 부여 규칙 취소

다음 `revoke-client-vpn-ingress` 예시에서는 모든 그룹에 대한 인터넷 액세스(`0.0.0.0/0`) 규칙을 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 Client VPN 관리자 안내서의 권한 부여 규칙을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RevokeClientVpnIngress 섹션을 참조하세요.

revoke-security-group-egress

다음 코드 예시는 revoke-security-group-egress의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 특정 주소 범위로 아웃바운드 트래픽을 허용하는 규칙 제거

다음 revoke-security-group-egress 예시 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.

예시 2: 특정 보안 그룹에 대한 아웃바운드 트래픽을 허용하는 규칙 제거

다음 revoke-security-group-egress 예시 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.

- API 세부 정보는 AWS CLI 명령 참조의 RevokeSecurityGroupEgress 섹션을 참조하세요.

`revoke-security-group-ingress`

다음 코드 예시는 `revoke-security-group-ingress`의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 보안 그룹에서 규칙 제거

다음 `revoke-security-group-ingress` 예시에서는 기본 VPC의 지정된 보안 그룹에서 `203.0.113.0/24` 주소 범위에 대한 TCP 포트 22 액세스를 제거합니다.

```
aws ec2 revoke-security-group-ingress \
    --group-name mySecurityGroup
    --protocol tcp \
    --port 22 \
    --cidr 203.0.113.0/24
```

이 명령은 성공 시 출력을 생성하지 않습니다.

자세한 내용은 Amazon EC2 사용 설명서의 보안 그룹을 참조하세요.

예시 2: IP 권한 세트를 사용하여 규칙 제거

다음 `revoke-security-group-ingress` 예시에서는 `ip-permissions` 파라미터를 사용하여 ICMP 메시지 `Destination Unreachable: Fragmentation Needed and Don't Fragment was Set`(유형 3, 코드 4)를 허용하는 인바운드 규칙을 제거합니다.

```
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 성공 시 출력을 생성하지 않습니다.

자세한 내용은 Amazon EC2 사용 설명서의 보안 그룹을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RevokeSecurityGroupIngress 섹션을 참조하세요.

run-instances

다음 코드 예시는 run-instances의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 기본 서브넷에서 인스턴스를 시작하는 방법

다음 run-instances 예제에서는 현재 리전의 기본 서브넷에서 t2.micro 유형의 단일 인스턴스를 시작하고 이를 해당 리전에서 기본 VPC에 대한 기본 서브넷에 연결합니다. 키 페어는 SSH(Linux) 또는 RDP(Windows)를 사용하여 인스턴스에 연결할 계획이 없는 경우 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 볼륨 및 인스턴스 스토어 볼륨 모두를 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 예제에서는 크기가 100GiB인 빈 EBS 볼륨(/dev/sdh)을 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 설명서에서 [블록 디바이스 매핑](#)을 참조하세요.

예제 4: 인스턴스를 시작하고 생성 시 태그를 추가하는 방법

다음 run-instances 예제에서는 키가 production이고 값이 webserver인 태그를 인스턴스에 추가합니다. 이 명령은 또 생성되는 EBS 볼륨(이 경우에는 루트 볼륨)에 키가 cost-center이고 값이 cc123인 태그를 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 설명서에서 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-specification을 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.

성능 버스트 기능이 있는 인스턴스에 대한 자세한 내용은 Amazon EC2 사용 설명서에서 성능 버스트 기능이 있는 인스턴스를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RunInstances를 참조하세요.

run-scheduled-instances

다음 코드 예시는 run-scheduled-instances의 사용 방법을 보여 줍니다.

AWS CLI

정기 인스턴스 시작

이 예시에서는 VPC에서 지정된 정기 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",
  "InstanceType": "c4.large",
  "NetworkInterfaces": [
    {
        "DeviceIndex": 0,
        "SubnetId": "subnet-12345678",
        "AssociatePublicIpAddress": true,
        "Groups": ["sg-12345678"]
    }
  ],
  "IamInstanceProfile": {
      "Name": "my-iam-role"
  }
}
```

출력:

```
{
  "InstanceIdSet": [
      "i-1234567890abcdef0"
  ]
}
```

이 예시에서는 EC2-Classic에서 지정된 정기 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"
  ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 RunScheduledInstances 섹션을 참조하세요.

search-local-gateway-routes

다음 코드 예시는 search-local-gateway-routes의 사용 방법을 보여 줍니다.

AWS CLI

로컬 게이트웨이 라우팅 테이블에서 라우팅 검색

다음 search-local-gateway-routes 예시에서는 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [SearchLocalGatewayRoutes](https://docs.aws.amazon.com/goto/aws-cli/ec2-2016-11-15/SearchLocalGatewayRoutes) 섹션을 참조하세요.

search-transit-gateway-multicast-groups

다음 코드 예시는 search-transit-gateway-multicast-groups의 사용 방법을 보여 줍니다.

AWS CLI

하나 이상의 전송 게이트웨이 멀티캐스트 그룹 검색 및 그룹 멤버십 정보 반환

다음 search-transit-gateway-multicast-groups 예시에서는 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 설명서의 전송 게이트웨이의 멀티캐스트를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 SearchTransitGatewayMulticastGroups 섹션을 참조하세요.

search-transit-gateway-routes

다음 코드 예시는 search-transit-gateway-routes의 사용 방법을 보여 줍니다.

AWS CLI

지정된 전송 게이트웨이 라우팅 테이블에서 라우팅 검색

다음 search-transit-gateway-routes 예시에서는 지정된 라우팅 테이블에 있는 static 유형의 모든 라우팅을 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 설명서의 전송 게이트웨이 라우팅 테이블을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 SearchTransitGatewayRoutes 섹션을 참조하세요.

send-diagnostic-interrupt

다음 코드 예시는 send-diagnostic-interrupt의 사용 방법을 보여 줍니다.

AWS CLI

진단 인터럽트 전송

다음 send-diagnostic-interrupt 예시에서는 지정된 인스턴스에 진단 인터럽트를 보냅니다.

```
aws ec2 send-diagnostic-interrupt \
    --instance-id i-1234567890abcdef0
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 SendDiagnosticInterrupt 섹션을 참조하세요.

start-instances

다음 코드 예시는 start-instances의 사용 방법을 보여 줍니다.

AWS CLI

Amazon EC2 인스턴스를 시작하는 방법

다음 예제에서는 지정된 Amazon EBS 지원 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.

- API 세부 정보는 AWS CLI 명령 참조의 StartInstances를 참조하세요.

start-network-insights-access-scope-analysis

다음 코드 예시는 start-network-insights-access-scope-analysis의 사용 방법을 보여 줍니다.

AWS CLI

Network Insights 액세스 범위 분석 시작

다음 start-network-insights-access-scope-analysis 예시에서는 AWS 계정의 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 설명서의 AWS CLI를 사용하여 Network Access Analyzer 시작하기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartNetworkInsightsAccessScopeAnalysis 섹션을 참조하세요.

start-network-insights-analysis

다음 코드 예시는 start-network-insights-analysis의 사용 방법을 보여 줍니다.

AWS CLI

경로 분석

다음 start-network-insights-analysis 예시에서는 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 설명서의 AWS CLI를 사용하여 시작하기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartNetworkInsightsAnalysis을 참조하세요.

start-vpc-endpoint-service-private-dns-verification

다음 코드 예시는 start-vpc-endpoint-service-private-dns-verification의 사용 방법을 보여 줍니다.

AWS CLI

DNS 확인 프로세스 시작

다음 start-vpc-endpoint-service-private-dns-verification 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 StartVpcEndpointServicePrivateDnsVerification 섹션을 참조하세요.

stop-instances

다음 코드 예시는 stop-instances의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: Amazon EC2 인스턴스를 중지하는 방법

다음 stop-instances 예제에서는 Amazon EBS 지원 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 인스턴스에서 최대 절전 모드를 적용하는 방법

다음 `stop-instances` 예제에서는 인스턴스에서 최대 절전 모드가 활성화되고 인스턴스가 최대 절전 모드 사전 조건을 충족하는 경우 Amazon EBS 지원 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 전환]을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [StopInstances]를 참조하세요.

terminate-client-vpn-connections

다음 코드 예시는 `terminate-client-vpn-connections`의 사용 방법을 보여 줍니다.

AWS CLI

Client VPN 엔드포인트에 대한 연결 종료

다음 `terminate-client-vpn-connections` 예시에서는 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 AWS Client VPN 관리자 안내서의 클라이언트 구성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TerminateClientVpnConnections을 참조하세요.

terminate-instances

다음 코드 예시는 `terminate-instances`의 사용 방법을 보여 줍니다.

AWS CLI

Amazon EC2 인스턴스를 종료하는 방법

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 Command Line Interface 사용 설명서에서 Amazon EC2 인스턴스 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조에서 TerminateInstances를 참조하세요.

`unassign-ipv6-addresses`

다음 코드 예시는 `unassign-ipv6-addresses`의 사용 방법을 보여 줍니다.

AWS CLI

네트워크 인터페이스에서 IPv6 주소 할당 해제

이 예시에서는 지정된 네트워크 인터페이스에서 지정된 IPv6 주소의 할당을 해제합니다.

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 세부 정보는 AWS CLI 명령 참조의 UnassignIpv6Addresses을 참조하세요.

`unassign-private-ip-addresses`

다음 코드 예시는 `unassign-private-ip-addresses`의 사용 방법을 보여 줍니다.

AWS CLI

네트워크 인터페이스에서 보조 프라이빗 IP 주소 할당 해제

이 예시에서는 지정된 네트워크 인터페이스에서 지정된 프라이빗 IP 주소의 할당을 해제합니다. 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
```

- API 세부 정보는 AWS CLI 명령 참조의 UnassignPrivateIpAddresses 섹션을 참조하세요.

`unassign-private-nat-gateway-address`

다음 코드 예시는 `unassign-private-nat-gateway-address`의 사용 방법을 보여 줍니다.

AWS CLI

프라이빗 NAT 게이트웨이에서 프라이빗 IP 주소 할당 취소

다음 `unassign-private-nat-gateway-address` 예시에서는 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 정보는 Amazon VPC 사용 설명서의 NAT 게이트웨이 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UnassignPrivateNatGatewayAddress 섹션을 참조하세요.

`unlock-snapshot`

다음 코드 예시는 `unlock-snapshot`의 사용 방법을 보여 줍니다.

AWS CLI

스냅샷 잠금 해제

다음 `unlock-snapshot` 예시에서는 지정된 스냅샷을 잠금 해제합니다.

```
aws ec2 unlock-snapshot \
    --snapshot-id snap-0b5e733b4a8df6e0d
```

출력:

```
{
    "SnapshotId": "snap-0b5e733b4a8df6e0d"
}
```

자세한 내용은 Amazon EBS 사용 설명서의 Snapshot Lock을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UnlockSnapshot 섹션을 참조하세요.

unmonitor-instances

다음 코드 예시는 `unmonitor-instances`의 사용 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"
          }
      }
  ]
}
```

- API 세부 정보는 AWS CLI 명령 참조에서 UnmonitorInstances를 참조하세요.

update-security-group-rule-descriptions-egress

다음 코드 예시는 update-security-group-rule-descriptions-egress의 사용 방법을 보여 줍니다.

AWS CLI

아웃바운드 보안 그룹 규칙의 설명 업데이트

다음 update-security-group-rule-descriptions-egress 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateSecurityGroupRuleDescriptionsEgress 섹션을 참조하세요.

update-security-group-rule-descriptions-ingress

다음 코드 예시는 update-security-group-rule-descriptions-ingress의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 인바운드 보안 그룹 규칙에 대한 설명을 CIDR 소스로 업데이트

다음 update-security-group-rule-descriptions-ingress 예시에서는 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.

예시 2: 인바운드 보안 그룹 규칙에 대한 설명을 접두사 목록 소스로 업데이트

다음 `update-security-group-rule-descriptions-ingress` 예시에서는 지정된 포트 및 접두사 목록에 대한 보안 그룹 규칙의 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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateSecurityGroupRuleDescriptionsIngress 섹션을 참조하세요.

`withdraw-byoip-cidr`

다음 코드 예시는 `withdraw-byoip-cidr`의 사용 방법을 보여 줍니다.

AWS CLI

주소 범위 광고 중지

다음 `withdraw-byoip-cidr` 예시에서는 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [WithdrawByoipCidr](#) 섹션을 참조하세요.

AWS CLI를 사용한 Amazon EC2 인스턴스 연결 예제

다음 코드 예제에서는 Amazon EC2 Instance Connect에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여 줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`send-ssh-public-key`

다음 코드 예제에서는 `send-ssh-public-key`의 사용 방법을 보여줍니다.

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.

- API 세부 정보는 AWS CLI 명령 참조의 SendSshPublicKey를 참조하세요.

AWS CLI를 사용한 Amazon ECR 예제

다음 코드 예제에서는 Amazon ECR에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`batch-check-layer-availability`

다음 코드 예제에서는 `batch-check-layer-availability`의 사용 방법을 보여줍니다.

AWS CLI

계층의 가용성을 확인하려면

다음 batch-check-layer-availability 예제에서는 cluster-autoscaler 리포지토리에 다이제스트 sha256:6171c7451a50945f8ddd72f7732cc04d7a0d1f48138a426b2e64387fdeb834ed가 있는 계층의 가용성을 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": []
}
```

- API 세부 정보는 AWS CLI 명령 참조의 BatchCheckLayerAvailability를 참조하세요.

batch-delete-image

다음 코드 예제에서는 batch-delete-image의 사용 방법을 보여줍니다.

AWS CLI

예제 1: 항목을 삭제하려면

다음 batch-delete-image 예제에서는 계정의 기본 레지스트리에서 지정된 리포지토리에 태그 precise가 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 리포지토리에서 prod 및 team1로 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.

- API 세부 정보는 AWS CLI 명령 참조의 [BatchDeleteImage](https://docs.aws.amazon.com/goto/aws-cli/ecr-2015-09-21/BatchDeleteImage)를 참조하세요.

batch-get-image

다음 코드 예제에서는 batch-get-image의 사용 방법을 보여줍니다.

AWS CLI

예제 1: 이미지를 가져오려면

다음 batch-get-image 예제에서는 계정의 기본 레지스트리에서 cluster-autoscaler라는 리포지토리에 태그 v1.13.6가 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 리포지토리에 prod 및 team1로 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.

- API 세부 정보는 AWS CLI 명령 참조의 BatchGetImage를 참조하세요.

`complete-layer-upload`

다음 코드 예제에서는 `complete-layer-upload`의 사용 방법을 보여줍니다.

AWS CLI

이미지 계층 업로드를 완료하려면

다음 `complete-layer-upload` 예제에서는 `layer-test` 리포지토리에 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CompleteLayerUpload를 참조하세요.

`create-repository`

다음 코드 예제에서는 `create-repository`의 사용 방법을 보여줍니다.

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 VPC 사용 설명서의 리포지토리 생성을 참조하세요.

예제 2: 이미지 태그 변경 불가능으로 구성된 리포지토리를 생성하려면

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 변경 가능성을 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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateRepository를 참조하세요.

delete-lifecycle-policy

다음 코드 예제에서는 `delete-lifecycle-policy`의 사용 방법을 보여줍니다.

AWS CLI

리포지토리의 수명 주기 정책을 삭제하려면

다음 `delete-lifecycle-policy` 예제에서는 `hello-world` 리포지토리의 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteLifecyclePolicy를 참조하세요.

delete-repository-policy

다음 코드 예제에서는 delete-repository-policy의 사용 방법을 보여줍니다.

AWS CLI

리포지토리에 대한 리포지토리 정책을 삭제하려면

다음 delete-repository-policy 예제에서는 cluster-autoscaler 리포지토리에 대한 리포지토리 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}"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteRepositoryPolicy를 참조하세요.

delete-repository

다음 코드 예제에서는 delete-repository의 사용 방법을 보여줍니다.

AWS CLI

리포지토리 삭제

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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteRepository를 참조하세요.

describe-image-scan-findings

다음 코드 예제에서는 describe-image-scan-findings의 사용 방법을 보여줍니다.

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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeImageScanFindings를 참조하세요.

describe-images

다음 코드 예제에서는 describe-images의 사용 방법을 보여줍니다.

AWS CLI

리포지토리의 이미지를 설명하려면

다음 describe-images 예제에서는 태그가 v1.13.6인 cluster-autoscaler 리포지토리의 이미지에 대한 세부 정보를 표시합니다.

```
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
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeImages](#)를 참조하세요.

`describe-repositories`

다음 코드 예제에서는 `describe-repositories`의 사용 방법을 보여줍니다.

AWS CLI

레지스트리의 리포지토리를 설명하려면

이 예제에서는 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeRepositories](#)를 참조하세요.

get-authorization-token

다음 코드 예제에서는 get-authorization-token의 사용 방법을 보여줍니다.

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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetAuthorizationToken을 참조하세요.

get-download-url-for-layer

다음 코드 예제에서는 get-download-url-for-layer의 사용 방법을 보여줍니다.

AWS CLI

계층의 다운로드 URL을 가져오려면

다음 get-download-url-for-layer 예제에서는 cluster-autoscaler 리포지토리에 다이제스트 sha256:6171c7451a50945f8ddd72f7732cc04d7a0d1f48138a426b2e64387fdeb834ed가 있는 계층의 다운로드 URL을 표시합니다.

```
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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetDownloadUrlForLayer를 참조하세요.

get-lifecycle-policy-preview

다음 코드 예제에서는 get-lifecycle-policy-preview의 사용 방법을 보여줍니다.

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.

- API 세부 정보는 AWS CLI 명령 참조의 GetLifecyclePolicyPreview를 참조하세요.

get-lifecycle-policy

다음 코드 예제에서는 get-lifecycle-policy의 사용 방법을 보여줍니다.

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.

- API 세부 정보는 AWS CLI 명령 참조의 GetLifecyclePolicy를 참조하세요.

get-login-password

다음 코드 예제에서는 get-login-password의 사용 방법을 보여줍니다.

AWS CLI

레지스트리에 인증할 암호를 검색하려면

다음 get-login-password에서는 IAM 보안 주체가 액세스할 수 있는 Amazon ECR 레지스트리에 인증하는 데 선택한 컨테이너 클라이언트와 함께 사용할 수 있는 암호를 표시합니다.

```
aws ecr get-login-password
```

출력:

```
<password>
```

Docker CLI와 함께 사용하려면 get-login-password 명령의 출력을 docker login 명령에 파이프로 연결합니다. 암호를 검색할 때 Amazon ECR 레지스트리가 있는 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 RDS 사용 설명서의 레지스트리 인증을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetLoginPassword를 참조하세요.

get-login

다음 코드 예제에서는 get-login의 사용 방법을 보여줍니다.

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>
```

- API 세부 정보는 AWS CLI 명령 참조의 GetLogin을 참조하세요.

`get-repository-policy`

다음 코드 예제에서는 `get-repository-policy`의 사용 방법을 보여줍니다.

AWS CLI

리포지토리에 대한 리포지토리 정책을 검색하려면

다음 `get-repository-policy` 예제에서는 `cluster-autoscaler` 리포지토리의 리포지토리 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}"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetRepositoryPolicy를 참조하세요.

`initiate-layer-upload`

다음 코드 예제에서는 `initiate-layer-upload`의 사용 방법을 보여줍니다.

AWS CLI

이미지 계층 업로드를 시작하려면

다음 `initiate-layer-upload` 예제에서는 `layer-test` 리포지토리에 이미지 계층 업로드를 시작합니다.

```
aws ecr initiate-layer-upload \
    --repository-name layer-test
```

출력:

```
{
    "partSize": 10485760,
    "uploadId": "6cb64b8a-9378-0e33-2ab1-b780fab8a9e9"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 InitiateLayerUpload를 참조하세요.

`list-images`

다음 코드 예제에서는 `list-images`의 사용 방법을 보여줍니다.

AWS CLI

리포지토리의 이미지를 나열하려면

다음 list-images 예제에서는 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListImages를 참조하세요.

list-tags-for-resource

다음 코드 예제에서는 list-tags-for-resource의 사용 방법을 보여줍니다.

AWS CLI

리포지토리 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [ListTagsForResource](#) 섹션을 참조하세요.

put-image-scanning-configuration

다음 코드 예제에서는 `put-image-scanning-configuration`의 사용 방법을 보여줍니다.

AWS CLI

리포지토리의 이미지 검사 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.

- API 세부 정보는 AWS CLI 명령 참조의 PutImageScanningConfiguration을 참조하세요.

put-image-tag-mutability

다음 코드 예제에서는 put-image-tag-mutability의 사용 방법을 보여줍니다.

AWS CLI

리포지토리에 대한 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 않습니다.

```
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 변경 가능성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutImageTagMutability를 참조하세요.

put-image

다음 코드 예제에서는 put-image의 사용 방법을 보여줍니다.

AWS CLI

매니페스트를 사용하여 이미지에 다시 태그를 지정하려면

다음 put-image 예제에서는 기존 이미지 매니페스트를 사용하여 hello-world 리포지토리에 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 PutImage를 참조하세요.

put-lifecycle-policy

다음 코드 예제에서는 put-lifecycle-policy의 사용 방법을 보여줍니다.

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.

- API 세부 정보는 AWS CLI 명령 참조의 PutLifeCyclePolicy를 참조하세요.

set-repository-policy

다음 코드 예제에서는 set-repository-policy의 사용 방법을 보여줍니다.

AWS CLI

리포지토리에 대한 리포지토리 정책을 설정하려면

다음 set-repository-policy 예제에서는 파일에 포함된 리포지토리 정책을 cluster-autoscaler 리포지토리에 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}"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 SetRepositoryPolicy를 참조하세요.

start-image-scan

다음 코드 예제에서는 start-image-scan의 사용 방법을 보여줍니다.

AWS CLI

이미지 취약성 스캔을 시작하려면

다음 start-image-scan 예제에서는 지정된 리포지토리의 이미지 다이제스트에 의해 지정되고 이를 위한 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.

- API 세부 정보는 AWS CLI 명령 참조의 StartImageScan를 참조하세요.

`start-lifecycle-policy-preview`

다음 코드 예제에서는 `start-lifecycle-policy-preview`의 사용 방법을 보여줍니다.

AWS CLI

수명 주기 정책 미리 보기를 생성하려면

다음 `start-lifecycle-policy-preview` 예제에서는 지정된 리포지토리에 대해 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.json`의 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 StartLifecyclePolicyPreview를 참조하세요.

`tag-resource`

다음 코드 예제에서는 `tag-resource`의 사용 방법을 보여줍니다.

AWS CLI

리포지토리에 태그를 추가하려면

다음 `tag-resource` 예제에서는 보고서에 `Stage` 키 및 값 `Integ`로 태그를 추가합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

`untag-resource`

다음 코드 예제에서는 `untag-resource`의 사용 방법을 보여줍니다.

AWS CLI

리포지토리의 태그를 해제하려면

다음 `untag-resource` 예제에서는 `hello-world` 리포지토리에서 키 `Stage`가 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.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

`upload-layer-part`

다음 코드 예제에서는 `upload-layer-part`의 사용 방법을 보여줍니다.

AWS CLI

계층 부분을 업로드하려면

다음 `upload-layer-part`에서는 이미지 계층 부분을 `layer-test` 리포지토리에 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 UploadLayerPart를 참조하세요.

AWS CLI를 사용한 Amazon ECR Public 예제

다음 코드 예제에서는 Amazon ECR Public에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여 줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

batch-delete-image

다음 코드 예시는 batch-delete-image의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 이미지 다이제스트 ID를 사용하여 이미지를 삭제하려면 이미지와 모든 태그가 퍼블릭 레지스트리의 리포지토리 내에서 삭제됩니다.

다음 batch-delete-image 예제에서는 이미지 다이제스트를 지정하여 이미지를 삭제합니다.

```
aws ecr-public batch-delete-image \
    --repository-name project-a/nginx-web-app \
    --image-
ids imageDigest=sha256:b1f9deb5fe3711a3278379ebbcaefbc5d70a2263135db86bd27a0dae150546c2
```

출력:

```
{
```

```
"imageIds": [
    {
        "imageDigest":
 "sha256:b1f9deb5fe3711a3278379ebbcaefbc5d70a2263135db86bd27a0dae150546c2",
        "imageTag": "latest"
    }
],
"failures": []
}
```

자세한 내용은 Amazon ECR Public 사용 설명서의 [퍼블릭 리포지토리에서 이미지 삭제](#)를 참조하세요.

예제 2: 리포지토리에서 삭제하려는 이미지와 연결된 태그를 지정하여 이미지 삭제.

다음 `batch-delete-image` 예제에서는 퍼블릭 레지스트리에 `project-a/nginx-web-app` 이름이 지정된 이미지 리포지토리와 연결된 태그를 지정하여 이미지를 삭제합니다. 태그가 하나뿐이고 이 명령을 실행하면 이미지가 제거됩니다. 그렇지 않으면 동일한 이미지에 대해 여러 태그가 있는 경우 하나를 지정하면 태그만 리포지토리에서 제거되고 이미지는 제거되지 않습니다.

```
aws ecr-public batch-delete-image \
    --repository-name project-a/nginx-web-app \
    --image-ids imageTag=_temp
```

출력:

```
{
    "imageIds": [
        {
            "imageDigest":
 "sha256:f7a86a0760e2f8d7eff07e515fc87bf4bac45c35376c06f9a280f15ecad6d7e0",
            "imageTag": "_temp"
        }
    ],
    "failures": []
}
```

자세한 내용은 Amazon ECR Public 사용 설명서의 [퍼블릭 리포지토리에서 이미지 삭제](#)를 참조하세요.

예제 3: 여러 이미지를 삭제하려면 퍼블릭 레지스트리의 리포지토리에 대한 요청에서 여러 이미지 태그 또는 이미지 다이제스트를 지정할 수 있습니다.

다음 `batch-delete-image` 예제에서는 요청에 여러 이미지 태그 또는 이미지 다이제스트를 지정하여 project-a/nginx-web-app이라는 리포지토리에서 여러 이미지를 삭제합니다.

```
aws ecr-public batch-delete-image \
    --repository-name project-a/nginx-web-app \
    --image-ids imageTag=temp2.0
  imageDigest=sha256:47ba980bc055353d9c0af89b1894f68faa43ca93856917b8406316be86f01278
```

출력:

```
{
     "imageIds": [
         {
             "imageDigest":
 "sha256:47ba980bc055353d9c0af89b1894f68faa43ca93856917b8406316be86f01278"
         },
         {
             "imageDigest":
 "sha256:f7a86a0760e2f8d7eff07e515fc87bf4bac45c35376c06f9a280f15ecad6d7e0",
             "imageTag": "temp2.0"
         }
     ],
     "failures": []
 }
```

자세한 내용은 Amazon ECR Public 사용 설명서의 [퍼블릭 리포지토리에서 이미지 삭제]를 참조하세요.

예제 4: 레지스트리 ID 및 이미지 다이제스트 ID를 사용하여 교차 AWS 계정의 이미지를 삭제하려면 이미지와 모든 태그가 퍼블릭 레지스트리의 리포지토리 내에서 삭제됩니다.

다음 `batch-delete-image` 예제에서는 교차 AWS 계정에서 이미지 다이제스트를 지정하여 이미지를 삭제합니다.

```
aws ecr-public batch-delete-image \
    --registry-id 123456789098 \
    --repository-name project-a/nginx-web-app \
    --image-
ids imageDigest=sha256:b1f9deb5fe3711a3278379ebbcaefbc5d70a2263135db86bd27a0dae150546c2
 \
    --region us-east-1
```

출력:

```
{
    "imageIds": [
        {
            "imageDigest":
 "sha256:b1f9deb5fe3711a3278379ebbcaefbc5d70a2263135db86bd27a0dae150546c2",
            "imageTag": "temp2.0"
        }
    ],
    "failures": []
}
```

자세한 내용은 Amazon ECR Public 사용 설명서의 [퍼블릭 리포지토리에서 이미지 삭제]를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [BatchDeleteImage]를 참조하세요.

`create-repository`

다음 코드 예시는 `create-repository`의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 퍼블릭 레지스트리에서 리포지토리를 생성하는 방법

다음 `create-repository` 예제에서는 퍼블릭 레지스트리에 project-a/nginx-web-app이라는 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 ECR Public 사용 설명서의 Creating a public repository를 참조하세요.

예제 2: 리포지토리의 이미지가 호환되는 리포지토리의 내용, 시스템 및 운영 아키텍처에 대한 간단한 설명과 함께 퍼블릭 레지스트리에 리포지토리를 만드는 방법

다음 `create-repository` 예제에서는 리포지토리의 내용, 리포지토리의 이미지가 호환되는 시스템 및 운영 아키텍처에 대한 간략한 설명과 함께 퍼블릭 레지스트리에 project-a/nginx-web-app이라는 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 ECR Public 사용 설명서의 퍼블릭 리포지토리 생성을 참조하세요.

예제 3: 퍼블릭 레지스트리에 리포지토리를 만드는 방법과 logoImageBlob, aboutText, usageText 및 태그 정보

다음 `create-repository` 예제에서는 logoImageBlob, aboutText, usageText 및 태그 정보와 함께 퍼블릭 레지스트리에 project-a/nginx-web-app이라는 이름의 리포지토리를 생성합니다.

```
aws ecr-public create-repository \
    --cli-input-json file://myfile.json
```

`myfile.json`의 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 &lt;package&gt;` 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 ECR Public 사용 설명서의 퍼블릭 리포지토리 생성과 Amazon ECR Public 사용 설명서의 리포지토리 카탈로그 데이터를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateRepository를 참조하세요.

`delete-repository-policy`

다음 코드 예시는 `delete-repository-policy`의 사용 방법을 보여 줍니다.

AWS CLI

퍼블릭 레지스트리에서 리포지토리 정책을 삭제하려면

다음 `delete-repository-policy` 예제에서는 AWS 계정의 ECR Public 리포지토리에 대한 리포지토리 정책을 삭제합니다.

```
aws ecr-public delete-repository-policy \
     --repository-name project-a/nginx-web-app \
     --region us-east-1
```

출력:

```
{
    "registryId": "123456789012",
    "repositoryName": "project-a/nginx-web-app",
    "policyText": "{\n  \"Version\" : \"2008-10-17\",\n  \"Statement\" : [ {\n
    \"Sid\" : \"AllowPush\",\n    \"Effect\" : \"Allow\",\n    \"Principal
\" : {\n      \"AWS\" : [ \"arn:aws:iam:"123456789012":user/eksuser1\",
 \"arn:aws:iam:"123456789012":user/admin\" ]\n    },\n    \"Action\" :
 [ \"ecr-public:BatchCheckLayerAvailability\", \"ecr-public:PutImage\",
 \"ecr-public:InitiateLayerUpload\", \"ecr-public:UploadLayerPart\", \"ecr-
public:CompleteLayerUpload\" ]\n  } ]\n}"
}
```

자세한 내용은 Amazon ECR Public 사용 설명서의 퍼블릭 리포지토리 정책 명령문 삭제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteRepositoryPolicy를 참조하세요.

`delete-repository`

다음 코드 예시는 `delete-repository`의 사용 방법을 보여 줍니다.

AWS CLI

퍼블릭 레지스트리에서 리포지토리를 삭제하는 방법

다음 `delete-repository` 예제에서는 퍼블릭 레지스트리에서 이름이 `project-a/nginx-web-app`인 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 ECR Public 사용 설명서의 퍼블릭 리포지토리 삭제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteRepository를 참조하세요.

describe-image-tags

다음 코드 예시는 `describe-image-tags`의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 퍼블릭 리포지토리의 이미지 태그 세부 정보 설명

다음 `describe-image-tags` 예제에서는 `project-a/nginx-web-app` 샘플 리포지토리의 이미지 태그를 설명합니다.

```
aws ecr-public describe-image-tags \
    --repository-name project-a/nginx-web-app \
    --region us-east-1
```

출력:

```
{
    "imageTagDetails": [
        {
            "imageTag": "latest",
            "createdAt": "2024-07-10T22:29:00-05:00",
            "imageDetail": {
                "imageDigest":
 "sha256:b1f9deb5fe3711a3278379ebbcaefbc5d70a2263135db86bd27a0dae150546c2",
                "imageSizeInBytes": 121956548,
                "imagePushedAt": "2024-07-10T22:29:00-05:00",
```

```
                "imageManifestMediaType": "application/
vnd.docker.distribution.manifest.v2+json",
                "artifactMediaType": "application/
vnd.docker.container.image.v1+json"
            }
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeImageTags를 참조하세요.

describe-images

다음 코드 예시는 `describe-images`의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 퍼블릭 레지스트리 리포지토리의 이미지 설명

다음 `describe-images` 예제에서는 퍼블릭 레지스트리에서 `project-a/nginx-web-app`이라는 리포지토리의 imagesDetails를 설명합니다.

```
aws ecr-public describe-images \
    --repository-name project-a/nginx-web-app \
    --region us-east-1
```

출력:

```
{
    "imageDetails": [
        {
            "registryId": "123456789012",
            "repositoryName": "project-a/nginx-web-app",
            "imageDigest":
 "sha256:0d8c93e72e82fa070d49565c00af32abbe8ddfd7f75e39f4306771ae0628c7e8",
            "imageTags": [
                "temp1.0"
            ],
            "imageSizeInBytes": 123184716,
            "imagePushedAt": "2024-07-23T11:32:49-05:00",
            "imageManifestMediaType": "application/
vnd.docker.distribution.manifest.v2+json",
```

```
            "artifactMediaType": "application/vnd.docker.container.image.v1+json"
        },
        {
            "registryId": "123456789012",
            "repositoryName": "project-a/nginx-web-app",
            "imageDigest":
 "sha256:b1f9deb5fe3711a3278379ebbcaefbc5d70a2263135db86bd27a0dae150546c2",
            "imageTags": [
                "temp2.0"
            ],
            "imageSizeInBytes": 121956548,
            "imagePushedAt": "2024-07-23T11:39:38-05:00",
            "imageManifestMediaType": "application/
vnd.docker.distribution.manifest.v2+json",
            "artifactMediaType": "application/vnd.docker.container.image.v1+json"
        },
        {
            "registryId": "123456789012",
            "repositoryName": "project-a/nginx-web-app",
            "imageDigest":
 "sha256:f7a86a0760e2f8d7eff07e515fc87bf4bac45c35376c06f9a280f15ecad6d7e0",
            "imageTags": [
                "temp3.0",
                "latest"
            ],
            "imageSizeInBytes": 232108879,
            "imagePushedAt": "2024-07-22T00:54:34-05:00",
            "imageManifestMediaType": "application/
vnd.docker.distribution.manifest.v2+json",
            "artifactMediaType": "application/vnd.docker.container.image.v1+json"
        }
    ]
}
```

자세한 내용은 Amazon ECR Public 사용 설명서의 퍼블릭 리포지토리의 이미지 설명을 참조하세요.

예제 2: imageTags & imagePushedAt를 정렬하여 리포지토리의 이미지 설명

다음 `describe-images` 예제에서는 퍼블릭 레지스트리의 project-a/nginx-web-app이라는 리포지토리 내의 이미지를 설명합니다.

```
aws ecr-public describe-images \
```

```
    --repository-name project-a/nginx-web-app \
    --query 'sort_by(imageDetails,& imagePushedAt)[*].imageTags[*]' \
    --output text
```

출력:

```
temp3.0 latest
temp1.0
temp2.0
```

예제 3: 리포지토리에서 푸시된 마지막 이미지 태그 2개를 생성하기 위해 리포지토리의 이미지를 설명

다음 `describe-images` 예제에서는 퍼블릭 레지스트리의 `project-a/nginx-web-app`이라는 리포지토리에서 이미지 태그 세부 정보를 가져오고 결과를 쿼리하여 처음 두 레코드만 표시합니다.

```
aws ecr-public describe-images \
    --repository-name project-a/nginx-web-app  \
    --query 'sort_by(imageDetails,& imagePushedAt)[*].imageTags[*] | [0:2]' \
    --output text
```

출력:

```
temp3.0 latest
temp1.0
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeImages를 참조하세요.

`describe-registries`

다음 코드 예시는 `describe-registries`의 사용 방법을 보여 줍니다.

AWS CLI

퍼블릭 레지스트리의 모든 레지스트리를 설명하려면

다음 `describe-registries` 예제에서는 계정의 모든 레지스트리를 설명합니다.

```
aws ecr-public describe-registries
```

출력:

```
{
 "registries": [
     {
         "registryId": "123456789012",
         "registryArn": "arn:aws:ecr-public::123456789012:registry/123456789012",
         "registryUri": "public.ecr.aws/publicregistrycustomalias",
         "verified": false,
         "aliases": [
             {
                 "name": "publicregistrycustomalias",
                 "status": "ACTIVE",
                 "primaryRegistryAlias": true,
                 "defaultRegistryAlias": true
             }
         ]
     }
     ]
 }
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeRegistries를 참조하세요.

describe-repository

다음 코드 예시는 `describe-repository`의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 퍼블릭 레지스트리의 리포지토리 설명

다음 `describe-repositories` 예제에서는 퍼블릭 레지스트리의 `project-a/nginx-web-app`이라는 리포지토리를 설명합니다.

```
aws ecr-public describe-repositories \
    --repository-name project-a/nginx-web-app
```

출력:

```
{
    "repositories": [
```

```
        {
            "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": "2024-07-07T00:07:56.526000-05:00"
        }
    ]
}
```

예제 2: 테이블의 퍼블릭 레지스트리에 있는 모든 리포지토리 설명

다음 `describe-repositories` 예제에서는 퍼블릭 레지스트리의 모든 리포지토리를 설명한 다음 리포지토리 이름을 테이블 형식으로 출력합니다.

```
aws ecr-public describe-repositories \
    --region us-east-1 \
    --output table \
    --query "repositories[*].repositoryName"
```

출력:

```
-----------------------------
|   DescribeRepositories    |
+---------------------------+
|  project-a/nginx-web-app  |
|  nginx                    |
|  myfirstrepo1             |
|  helm-test-chart          |
|  test-ecr-public          |
|  nginx-web-app            |
|  sample-repo              |
+---------------------------+
```

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeRepository](#) 섹션을 참조하세요.

`get-authorization-token`

다음 코드 예시는 `get-authorization-token`의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: IAM 보안 주체가 액세스할 수 있는 Amazon ECR 퍼블릭 레지스트리에 대한 권한 부여 토큰 검색

다음 get-authorization-token 예제에서는 AWS CLI를 사용하여 권한 부여 토큰을 가져와 환경 변수로 설정합니다.

```
aws ecr-public get-authorization-token \
    --region us-east-1
```

출력:

```
{
    "authorizationData": {
        "authorizationToken":
 "QVdTOmV5SndZWGxzYWJjJKJFHDSFKJHERWUY65IOU36TRYEGFNSDLRIUOTUYTHJKLDFGOcmFUQk9OSFV2UVV4a0x6Sm1
        "expiresAt": "2024-07-25T21:37:26.301000-04:00"
    }
}
```

자세한 내용은 Amazon ECR Public 사용 설명서의 Amazon ECR 퍼블릭 레지스트리를 참조하세요.

예제 2: IAM 위탁자가 액세스할 수 있는 Amazon ECR 퍼블릭 레지스트리에 대한 권한 부여 토큰 검색

다음 get-authorization-token 예제에서는 AWS CLI를 사용하여 권한 부여 토큰을 가져와 환경 변수로 설정합니다.

```
aws ecr-public get-authorization-token \
    --region us-east-1 \
    --output=text \
    --query 'authorizationData.authorizationToken'
```

출력:

```
QVdTOmV5SndZWGxzYWJjJKJFHDSFKJHERWUY65IOU36TRYEGFNSDLRIUOTUYTHJKLDFGOcmFUQk9OSFV2UVV4a0x6Sm1ZV
```

자세한 내용은 Amazon ECR Public 사용 설명서의 Amazon ECR 퍼블릭 레지스트리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetAuthorizationToken을 참조하세요.

get-login-password

다음 코드 예시는 get-login-password의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: Amazon ECR 퍼블릭 레지스트리에 대한 Docker 인증

다음 get-login-password 예제에서는 Amazon ECR 퍼블릭 레지스트리에 인증하는 데 사용할 수 있는 GetAuthorizationToken API를 사용하여 인증 토큰을 검색하고 표시합니다.

```
aws ecr-public get-login-password \
    --region us-
east-1
| docker login \
    --username AWS \
    --password-stdin public.ecr.aws
```

이 명령은 터미널에 출력을 생성하지 않고 대신 출력을 Docker에 파이프합니다.

자세한 내용은 Amazon ECR Public 사용 설명서의 퍼블릭 레지스트리에 인증 섹션을 참조하세요.

예제 2: 사용자 지정 AmazonECR 퍼블릭 레지스트리에 대한 Docker 인증

다음 get-login-password 예제에서는 사용자 지정 Amazon ECR 퍼블릭 레지스트리에 인증하는 데 사용할 수 있는 GetAuthorizationToken API를 사용하여 인증 토큰을 검색하고 표시합니다.

```
 aws ecr-public get-login-password \
    --region us-east-1 \
| docker login \
    --username AWS \
    --password-stdin public.ecr.aws/<your-public-registry-custom-alias>
```

이 명령은 터미널에 출력을 생성하지 않고 대신 출력을 Docker에 파이프합니다.

자세한 내용은 Amazon ECR Public 사용 설명서의 자체 Amazon ECR 퍼블릭에 인증을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetLoginPassword를 참조하세요.

`get-registry-catalog-data`

다음 코드 예시는 `get-registry-catalog-data`의 사용 방법을 보여 줍니다.

AWS CLI

퍼블릭 ECR 레지스트리의 카탈로그 메타데이터를 검색하려면

다음 `get-registry-catalog-data` 예제에서는 ECR 퍼블릭 레지스트리의 카탈로그 메타데이터를 검색합니다.

```
aws ecr-public get-registry-catalog-data \
    --region us-east-1
```

출력:

```
{
    "registryCatalogData": {
        "displayName": "YourCustomPublicRepositoryalias"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetRegistryCatalogData 섹션을 참조하세요.

`get-repository-catalog-data`

다음 코드 예시는 `get-repository-catalog-data`의 사용 방법을 보여 줍니다.

AWS CLI

퍼블릭 레지스트리의 리포지토리에 대한 카탈로그 메타데이터를 검색하려면

다음 `get-repository-catalog-data` 예제에서는 퍼블릭 레지스트리의 `project-a/nginx-web-app` 리포지토리에 대한 카탈로그 메타데이터를 나열합니다.

```
aws ecr-public get-repository-catalog-data \
    --repository-name project-a/nginx-web-app \
    --region us-east-1
```

출력:

```
{
```

```
    "catalogData": {
        "description": "My project-a ECR Public Repository",
        "architectures": [
            "ARM",
            "ARM 64",
            "x86",
            "x86-64"
        ],
        "operatingSystems": [
            "Linux"
        ],
        "logoUrl": "https://d3g9o9u8re44ak.cloudfront.net/logo/491d3846-8f33-4d8b-a10c-c2ce271e6c0d/4f09d87c-2569-4916-a932-5c296bf6f88a.png",
        "aboutText": "## Quick reference\n\nMaintained <truncated>",
        "usageText": "## Supported architectures\n\namd64, arm64v8\n\n## <truncated>"
    }
}
```

자세한 내용은 Amazon ECR Public 사용 설명서의 리포지토리 카탈로그 데이터를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetRepositoryCatalogData 섹션을 참조하세요.

`get-repository-policy`

다음 코드 예시는 `get-repository-policy`의 사용 방법을 보여 줍니다.

AWS CLI

리포지토리와 연결된 리포지토리 정책을 가져오려면

다음 `get-repository-policy` 예제에서는 리포지토리와 연결된 리포지토리 정책을 가져옵니다.

```
aws ecr-public get-repository-policy \
    --repository-name project-a/nginx-web-app \
    --region us-east-1
```

출력:

```
{
    "registryId": "123456789012",
```

```
    "repositoryName": "project-a/nginx-web-app",
    "policyText": "{\n  \"Version\" : \"2008-10-17\",\n  \"Statement\" : [ {\n
    \"Sid\" : \"AllowPush\",\n    \"Effect\" : \"Allow\",\n    \"Principal
\" : {\n      \"AWS\" : [ \"arn:aws:iam::123456789012:user/eksuser1\",
 \"arn:aws:iam::123456789012:user/admin\" ]\n    },\n    \"Action\" :
 [ \"ecr-public:BatchCheckLayerAvailability\", \"ecr-public:PutImage\",
 \"ecr-public:InitiateLayerUpload\", \"ecr-public:UploadLayerPart\", \"ecr-
public:CompleteLayerUpload\" ]\n  } ]\n}"
}
```

자세한 내용은 Amazon ECR Public 사용 설명서의 AWS SDK 또는 CLI에서 GetRepositoryPolicy 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetRepositoryPolicy를 참조하세요.

`list-tags-for-resource`

다음 코드 예시는 `list-tags-for-resource`의 사용 방법을 보여 줍니다.

AWS CLI

퍼블릭 레지스트리에서 퍼블릭 리포지토리의 태그를 나열하려면

다음 `list-tags-for-resource` 예제에서는 퍼블릭 레지스트리에 `project-a/nginx-web-app`이라는 리소스의 태그를 나열합니다.

```
aws ecr-public list-tags-for-resource \
    --resource-arn arn:aws:ecr-public::123456789012:repository/project-a/nginx-web-
app \
    --region us-east-1
```

출력:

```
{
    "tags": [
        {
            "Key": "Environment",
            "Value": "Prod"
        },
        {
            "Key": "stack",
            "Value": "dev1"
```

```
        },
        {
            "Key": "Name",
            "Value": "project-a/nginx-web-app"
        }
    ]
}
```

자세한 내용은 Amazon ECR Public 사용 설명서의 [퍼블릭 리포지토리의 태그 나열](#)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ListTagsForResource](#)를 참조하세요.

put-registry-catalog-data

다음 코드 예시는 `put-registry-catalog-data`의 사용 방법을 보여 줍니다.

AWS CLI

퍼블릭 ECR 레지스트리에 대한 카탈로그 메타데이터를 생성하거나 업데이트하려면

다음 `put-registry-catalog-data`에서는 ECR 퍼블릭 레지스트리에 대한 카탈로그 메타데이터를 생성하거나 업데이트합니다. 확인된 계정 배지가 있는 계정만 레지스트리 표시 이름을 가질 수 있습니다.

```
aws ecr-public put-registry-catalog-data \
    --region us-east-1 \
    --display-name <YourCustomPublicRepositoryalias>
```

출력:

```
{
    "registryCatalogData": {
        "displayName": "YourCustomPublicRepositoryalias"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [PutRegistryCatalogData](#)를 참조하세요.

put-repository-catalog-data

다음 코드 예시는 `put-repository-catalog-data`의 사용 방법을 보여 줍니다.

AWS CLI

퍼블릭 레지스트리의 리포지토리에 대한 카탈로그 데이터를 생성하거나 업데이트하려면

다음 `put-repository-catalog-data` 예제에서는 logoImageBlob, aboutText, usageText 및 태그 정보와 함께 퍼블릭 레지스트리에 project-a/nginx-web-app이라는 이름의 리포지토리의 카탈로그 데이터를 생성하거나 업데이트합니다.

```
aws ecr-public put-repository-catalog-data \
    --repository-name project-a/nginx-web-app \
    --cli-input-json file://repository-catalog-data.json \
    --region us-east-1
```

`repository-catalog-data.json`의 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.",
        "usageText": "## Supported architectures are as follows."
    }
}
```

출력:

```
{
    "catalogData": {
        "description": "My project-a ECR Public Repository",
        "architectures": [
            "ARM",
            "ARM 64",
            "x86",
```

```
            "x86-64"
        ],
        "operatingSystems": [
            "Linux"
        ],
        "logoUrl": "https://d3g9o9u8re44ak.cloudfront.net/logo/df86cf58-ee60-4061-b804-0be24d97ccb1/4a9ed9b2-69e4-4ede-b924-461462d20ef0.png",
        "aboutText": "## Quick reference.",
        "usageText": "## Supported architectures are as follows."
    }
}
```

자세한 내용은 Amazon ECR Public 사용 설명서의 리포지토리 카탈로그 데이터를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutRepositoryCatalogData를 참조하세요.

set-repository-policy

다음 코드 예시는 set-repository-policy의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 리포지토리에서 풀을 허용하도록 리포지토리 정책 설정

다음 set-repository-policy 예제에서는 ECR 퍼블릭 리포지토리 정책을 지정된 리포지토리에 적용하여 액세스 권한을 제어합니다.

```
aws ecr-public set-repository-policy \
    --repository-name project-a/nginx-web-app \
    --policy-text file://my-repository-policy.json
```

my-repository-policy.json의 콘텐츠:

```
{
    "Version" : "2008-10-17",
    "Statement" : [
        {
            "Sid" : "allow public pull",
            "Effect" : "Allow",
            "Principal" : "*",
            "Action" : [
                "ecr:BatchCheckLayerAvailability",
                "ecr:BatchGetImage",
```

```
                "ecr:GetDownloadUrlForLayer"
            ]
        }
    ]
}
```

출력:

```
{
    "registryId": "12345678901",
    "repositoryName": "project-a/nginx-web-app",
    "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 내용은 Amazon ECR Public 사용 설명서의 리포지토리 정책 명령문 설정을 참조하세요.

예제 2: 계정 내 IAM 사용자가 이미지를 푸시할 수 있도록 리포지토리 정책 설정

다음 `set-repository-policy` 예제에서는 계정 내 IAM 사용자가 정책 텍스트로 `file://my-repository-policy.json`이라는 입력 파일을 사용하여 AWS 계정의 ECR 리포지토리로 이미지를 푸시할 수 있도록 허용합니다.

```
aws ecr-public set-repository-policy \
    --repository-name project-a/nginx-web-app \
    --policy-text file://my-repository-policy.json
```

`my-repository-policy.json`의 콘텐츠:

```
{
    "Version": "2008-10-17",
    "Statement": [
        {
            "Sid": "AllowPush",
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::account-id:user/push-pull-user-1",
                    "arn:aws:iam::account-id:user/push-pull-user-2"
                ]
```

```
            },
            "Action": [
                "ecr-public:BatchCheckLayerAvailability",
                "ecr-public:PutImage",
                "ecr-public:InitiateLayerUpload",
                "ecr-public:UploadLayerPart",
                "ecr-public:CompleteLayerUpload"
            ]
        }
    ]
}
```

출력:

```
{
    "registryId": "12345678901",
    "repositoryName": "project-a/nginx-web-app",
    "policyText": "{\n  \"Version\" : \"2008-10-17\",\n  \"Statement\" :
 [ {\n    \"Sid\" : \"AllowPush\",\n    \"Effect\" : \"Allow\",\n
 \"Principal\" : {\n      \"AWS\" : [ \"arn:aws:iam::12345678901:user/admin
\", \"arn:aws:iam::12345678901:user/eksuser1\" ]\n    },\n    \"Action\" :
 [ \"ecr-public:BatchCheckLayerAvailability\", \"ecr-public:PutImage\",
 \"ecr-public:InitiateLayerUpload\", \"ecr-public:UploadLayerPart\", \"ecr-
public:CompleteLayerUpload\" ]\n  } ]\n}"
}
```

자세한 내용은 Amazon ECR Public 사용 설명서의 리포지토리 정책 명령문 설정을 참조하세요.

예제 3: 다른 계정의 IAM 사용자가 이미지를 푸시할 수 있도록 리포지토리 정책 설정

다음 `set-repository-policy` 예제에서는 특정 계정이 AWS 계정에서 cli 입력 file://my-repository-policy.json을 사용하여 이미지를 푸시할 수 있도록 허용합니다.

```
aws ecr-public set-repository-policy \
    --repository-name project-a/nginx-web-app \
    --policy-text file://my-repository-policy.json
```

`my-repository-policy.json`의 콘텐츠:

```
 {
     "Version": "2008-10-17",
     "Statement": [
         {
```

```
            "Sid": "AllowCrossAccountPush",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::other-or-same-account-id:role/RoleName"
            },
            "Action": [
                "ecr-public:BatchCheckLayerAvailability",
                "ecr-public:PutImage",
                "ecr-public:InitiateLayerUpload",
                "ecr-public:UploadLayerPart",
                "ecr-public:CompleteLayerUpload"
            ]
        }
    ]
}
```

출력:

```
{
    "registryId": "12345678901",
    "repositoryName": "project-a/nginx-web-app",
    "policyText": "{\n  \"Version\" : \"2008-10-17\",\n  \"Statement\" : [ {\n
 \"Sid\" : \"AllowCrossAccountPush\",\n    \"Effect\" : \"Allow\",\n    \"Principal
\" : {\n      \"AWS\" : \"arn:aws:iam::12345678901:role/RoleName\"\n    },\n
 \"Action\" : [ \"ecr-public:BatchCheckLayerAvailability\", \"ecr-public:PutImage
\", \"ecr-public:InitiateLayerUpload\", \"ecr-public:UploadLayerPart\", \"ecr-
public:CompleteLayerUpload\" ]\n  } ]\n}"
}
```

자세한 내용은 Amazon ECR Public 사용 설명서의 퍼블릭 리포지토리 정책 예제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 SetRepositoryPolicy를 참조하세요.

`tag-resource`

다음 코드 예시는 `tag-resource`의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 퍼블릭 레지스트리의 기존 퍼블릭 리포지토리에 태그 지정

다음 `tag-resource` 예제는 퍼블릭 레지스트리에서 `project-a/nginx-web-app`이라는 리포지토리에 태그를 지정합니다.

```
aws ecr-public tag-resource \
    --resource-arn arn:aws:ecr-public::123456789012:repository/project-a/nginx-web-
app \
    --tags Key=stack,Value=dev \
    --region us-east-1
```

자세한 내용은 Amazon ECR Public 사용 설명서의 퍼블릭 리포지토리에 태그 사용을 참조하세요.

예제 2: 퍼블릭 레지스트리에서 여러 태그로 기존 퍼블릭 리포지토리에 태그 지정.

다음 `tag-resource` 예제에서는 여러 태그로 기존 리포지토리에 태그를 지정합니다.

```
aws ecr-public tag-resource \
    --resource-arn arn:aws:ecr-public::890517186334:repository/project-a/nginx-web-
app  \
    --tags Key=key1,Value=value1 Key=key2,Value=value2 Key=key3,Value=value3 \
    --region us-east-1
```

자세한 내용은 Amazon ECR Public 사용 설명서의 퍼블릭 리포지토리에 태그 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

`untag-resource`

다음 코드 예시는 `untag-resource`의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 퍼블릭 레지스트리에서 기존 퍼블릭 리포지토리의 태그 해제

다음 `untag-resource` 예제는 퍼블릭 레지스트리에서 `project-a/nginx-web-app`이라는 리포지토리에 태그를 지정합니다.

```
aws ecr-public untag-resource \
    --resource-arn arn:aws:ecr-public::123456789012:repository/project-a/nginx-web-
app \
    --tag-keys stack \
    --region us-east-1
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon ECR Public 사용 설명서의 퍼블릭 리포지토리에 태그 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

AWS CLI를 사용한 Amazon ECS 예제

다음 코드 예제는 Amazon ECS와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

capacity-provider-update

다음 코드 예시는 `capacity-provider-update`의 사용 방법을 보여 줍니다.

AWS CLI

ECS 클러스터에서 용량 공급자 업데이트

다음 `update-capacity-provider` 예제는 ECS 클러스터에서 용량 공급자의 파라미터를 수정하는 방법을 보여줍니다.

```
aws ecs update-capacity-provider \
    --name Infra-ECS-Cluster-ECS-project-update-cluster-d6bb6d5b-
EC2CapacityProvider-3fIpdkLywwFt \
    --auto-scaling-group-
provider "managedScaling={status=DISABLED,targetCapacity=50,minimumScalingStepSize=2,maximum
```

출력:

```
{
    "capacityProvider": {
```

```
        "capacityProviderArn": "arn:aws:ecs:us-west-2:123456789012:capacity-provider/Infra-ECS-Cluster-ECS-project-update-cluster-d6bb6d5b-EC2CapacityProvider-3fIpdkLywwFt",
        "name": "Infra-ECS-Cluster-ECS-project-update-cluster-d6bb6d5b-EC2CapacityProvider-3fIpdkLywwFt",
        "status": "ACTIVE",
        "autoScalingGroupProvider": {
            "autoScalingGroupArn": "arn:aws:autoscaling:us-west-2:123456789012:autoScalingGroup:424941d1-b43f-4a17-adbb-08b6a6e397e1:autoScalingGroupName/Infra-ECS-Cluster-ECS-project-update-cluster-d6bb6d5b-ECSAutoScalingGroup-f44jrQHS2nRB",
            "managedScaling": {
                "status": "ENABLED",
                "targetCapacity": 100,
                "minimumScalingStepSize": 1,
                "maximumScalingStepSize": 10000,
                "instanceWarmupPeriod": 300
            },
            "managedTerminationProtection": "DISABLED",
            "managedDraining": "ENABLED"
        },
        "updateStatus": "UPDATE_IN_PROGRESS",
        "tags": []
    }
}
```

용량 공급자에 대한 자세한 내용은 Amazon ECS 개발자 안내서의 EC2 시작 유형에 대한 Amazon ECS 용량 공급자를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CapacityProviderUpdate를 참조하세요.

`create-capacity-provider`

다음 코드 예시는 `create-capacity-provider`의 사용 방법을 보여 줍니다.

AWS CLI

용량 공급자를 만들려면

다음 create-capacity-provider 예시에서는 MyASG라는 Auto Scaling 그룹을 사용하고 관리형 규모 조정 및 관리형 종료 보호가 활성화된 용량 공급자를 만듭니다. 이 구성은 Amazon ECS 클러스터 Auto Scaling에 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 ECS 개발자 안내서의 Amazon ECS 클러스터 오토 스케일링 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateCapacityProvider 섹션을 참조하세요.

`create-cluster`

다음 코드 예시는 `create-cluster`의 사용 방법을 보여 줍니다.

AWS CLI

예 1: 새 클러스터를 생성하는 방법

다음 `create-cluster` 예제에서는 `MyCluster`라는 클러스터를 만들고 향상된 관찰성을 갖춘 CloudWatch Container Insights를 사용 설정합니다.

```
aws ecs create-cluster \
    --cluster-name MyCluster \
    --settings name=containerInsights,value=enhanced
```

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": [],
        "settings": [
            {
                "name": "containerInsights",
                "value": "enhanced"
            }
        ],
        "tags": []
    }
}
```

자세한 내용은 Amazon ECS 개발자 안내서의 [클러스터 생성]을 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 공급자

를 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,
```

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 [Cluster capacity providers](클러스터 쿼리 언어)를 참조하세요.

예 3: 여러 태그가 포함된 새 클러스터를 생성하는 방법

다음 `create-cluster` 예시에서는 여러 태그가 있는 클러스터를 만듭니다. 간편 구문을 사용하여 태그를 추가하는 방법에 대한 자세한 내용은 AWSCLI 사용 설명서의 AWS Command Line Interface에서 간편 구문 사용을 참조하세요.

```
aws ecs create-cluster \
    --cluster-name MyCluster \
    --tags key=key1,value=value1 key=key2,value=value2
```

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"
            }
        ]
     }
 }
```

자세한 내용은 Amazon ECS 개발자 안내서의 클러스터 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateCluster를 참조하세요.

`create-service`

다음 코드 예시는 `create-service`의 사용 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],assignPublicIp
 \
    --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"
    }
}
```

자세한 정보는 Amazon ECS 개발자 안내서의 서비스 생성하기를 참조하세요.

예 2: EC2 시작 유형을 사용하여 서비스를 생성하는 방법

다음 `create-service` 예시에서는 EC2 시작 유형을 사용하는 태스크로 `ecs-simple-service`라는 서비스를 호출하는 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"
    }
}
```

자세한 정보는 Amazon ECS 개발자 안내서의 서비스 생성하기를 참조하세요.

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"
    }
}
```


자세한 정보는 Amazon ECS 개발자 안내서의 서비스 생성하기를 참조하세요.

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 예시에서는 `--cli-input-json` 옵션과 다음 콘텐츠가 포함된 `ecs-simple-service-elb.json`이라는 JSON 입력 파일을 사용합니다.

```
aws ecs create-service \
    --cluster MyCluster \
    --service-name ecs-simple-service-elb \
    --cli-input-json file://ecs-simple-service-elb.json
```

`ecs-simple-service-elb.json`의 콘텐츠:

```
 {
    "serviceName": "ecs-simple-service-elb",
    "taskDefinition": "ecs-demo",
    "loadBalancers": [
        {
            "loadBalancerName": "EC2Contai-EcsElast-123456789012",
            "containerName": "simple-demo",
            "containerPort": 80
        }
    ],
    "desiredCount": 10,
    "role": "ecsServiceRole"
}
```

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 로드 밸런싱을 사용하여 Amazon ECS 서비스 트래픽 분산을 참조하세요.

예시 5: 서비스 생성 시 Amazon EBS 볼륨 구성

다음 `create-service` 예시에서는 서비스에서 관리하는 각 작업에 대해 Amazon EBS 볼륨을 구성하는 방법을 보여줍니다. `AmazonECSInfrastructureRolePolicyForVolumes` 관리형 정책과 연결된 Amazon ECS 인프라 역할이 구성되어 있어야 합니다. `create-service` 요청에서와 동일한 볼륨 이름으로 작업 정의를 지정해야 합니다. 이 예시에서는 `--cli-input-json` 옵션과 다음 콘텐츠가 포함된 `ecs-simple-service-ebs.json`이라는 JSON 입력 파일을 사용합니다.

```
aws ecs create-service \
    --cli-input-json file://ecs-simple-service-ebs.json
```

`ecs-simple-service-ebs.json`의 콘텐츠:

```
{
    "cluster": "mycluster",
    "taskDefinition": "mytaskdef",
```

```
    "serviceName": "ecs-simple-service-ebs",
    "desiredCount": 2,
    "launchType": "FARGATE",
    "networkConfiguration":{
        "awsvpcConfiguration":{
            "assignPublicIp": "ENABLED",
            "securityGroups": ["sg-12344321"],
            "subnets":["subnet-12344321"]
        }
    },
    "volumeConfigurations": [
        {
            "name": "myEbsVolume",
            "managedEBSVolume": {
                "roleArn":"arn:aws:iam::123456789012:role/ecsInfrastructureRole",
                "volumeType": "gp3",
                "sizeInGiB": 100,
                "iops": 3000,
                "throughput": 125,
                "filesystemType": "ext4"
            }
        }
   ]
}
```

출력:

```
{
    "service": {
        "serviceArn": "arn:aws:ecs:us-west-2:123456789012:service/mycluster/ecs-
simple-service-ebs",
        "serviceName": "ecs-simple-service-ebs",
        "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/mycluster",
        "loadBalancers": [],
        "serviceRegistries": [],
        "status": "ACTIVE",
        "desiredCount": 2,
        "runningCount": 0,
        "pendingCount": 0,
        "launchType": "EC2",
        "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-definition/
mytaskdef:3",
        "deploymentConfiguration": {
```

```
            "deploymentCircuitBreaker": {
                "enable": false,
                "rollback": false
            },
            "maximumPercent": 200,
            "minimumHealthyPercent": 100
        },
        "deployments": [
            {
                "id": "ecs-svc/7851020056849183687",
                "status": "PRIMARY",
                "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-
definition/mytaskdef:3",
                "desiredCount": 0,
                "pendingCount": 0,
                "runningCount": 0,
                "failedTasks": 0,
                "createdAt": "2025-01-21T11:32:38.034000-06:00",
                "updatedAt": "2025-01-21T11:32:38.034000-06:00",
                "launchType": "EC2",
                "networkConfiguration": {
                    "awsvpcConfiguration": {
                        "subnets": [
                            "subnet-12344321"
                        ],
                        "securityGroups": [
                            "sg-12344321"
                        ],
                        "assignPublicIp": "DISABLED"
                    }
                },
                "rolloutState": "IN_PROGRESS",
                "rolloutStateReason": "ECS deployment ecs-svc/7851020056849183687 in
 progress.",
                "volumeConfigurations": [
                    {
                        "name": "myEBSVolume",
                        "managedEBSVolume": {
                            "volumeType": "gp3",
                            "sizeInGiB": 100,
                            "iops": 3000,
                            "throughput": 125,
                            "roleArn": "arn:aws:iam::123456789012:role/
ecsInfrastructureRole",
```

```
                            "filesystemType": "ext4"
                        }
                    }
                ]
            }
        ],
        "roleArn": "arn:aws:iam::123456789012:role/aws-service-role/
ecs.amazonaws.com/AWSServiceRoleForECS",
        "events": [],
        "createdAt": "2025-01-21T11:32:38.034000-06:00",
        "placementConstraints": [],
        "placementStrategy": [],
        "networkConfiguration": {
            "awsvpcConfiguration": {
                "subnets": [
                    "subnet-12344321"
                ],
                "securityGroups": [
                    "sg-12344321"
                ],
                "assignPublicIp": "DISABLED"
            }
        },
        "healthCheckGracePeriodSeconds": 0,
        "schedulingStrategy": "REPLICA",
        "deploymentController": {
            "type": "ECS"
        },
        "createdBy": "arn:aws:iam::123456789012:user/AIDACKCEVSQ6C2EXAMPLE",
        "enableECSManagedTags": false,
        "propagateTags": "NONE",
        "enableExecuteCommand": false,
        "availabilityZoneRebalancing": "DISABLED"
    }
}
```

자세한 내용은 Amazon ECS 개발자 안내서의 Amazon ECS에서 Amazon EBS 볼륨 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateService를 참조하세요.

`create-task-set`

다음 코드 예시는 `create-task-set`의 사용 방법을 보여 줍니다.

AWS CLI

작업 세트를 만들려면

다음 `create-task-set` 예시에서는 외부 배포 컨트롤러를 사용하는 서비스에서 작업 세트를 만듭니다.

```
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
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateTaskSet 섹션을 참조하세요.

delete-account-setting

다음 코드 예시는 delete-account-setting의 사용 방법을 보여 줍니다.

AWS CLI

특정 IAM 사용자 또는 IAM 역할의 계정 설정을 삭제하려면

다음 delete-account-setting 예시에서는 특정 IAM 사용자 또는 IAM 역할에 대한 계정 설정을 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 ECS 개발자 안내서의 Amazon 리소스 이름(ARN) 및 ID 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteAccountSetting 섹션을 참조하세요.

delete-attributes

다음 코드 예시는 delete-attributes의 사용 방법을 보여 줍니다.

AWS CLI

Amazon ECS 리소스에서 하나 이상의 사용자 지정 속성을 삭제하려면

다음 delete-attributes는 컨테이너 인스턴스에서 이름이 stack인 속성을 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteAttributes 섹션을 참조하세요.

delete-capacity-provider

다음 코드 예시는 delete-capacity-provider의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: Amazon 리소스 이름(ARN)을 사용하여 용량 공급자를 삭제하려면

다음 delete-capacity-provider 예시에서는 용량 공급자의 Amazon 리소스 이름(ARN)을 지정하여 용량 공급자를 삭제합니다. describe-capacity-providers 명령을 사용하여 용량 공급자 삭제 상태뿐만 아니라 ARN도 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 Cluster capacity providers(클러스터 쿼리 언어)를 참조하세요.

예시 2: 이름을 사용하여 용량 공급자를 삭제하려면

다음 `delete-capacity-provider` 예시에서는 용량 공급자의 짧은 이름을 지정하여 용량 공급자를 삭제합니다. `describe-capacity-providers` 명령을 사용하여 용량 공급자 삭제 상태뿐만 아니라 짧은 이름도 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 Cluster capacity providers(클러스터 쿼리 언어)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteCapacityProvider 섹션을 참조하세요.

delete-cluster

다음 코드 예시는 delete-cluster의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteCluster를 참조하세요.

delete-service

다음 코드 예시는 delete-service의 사용 방법을 보여 줍니다.

AWS CLI

서비스 삭제

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 정보는 Amazon ECS 개발자 안내서의 서비스 삭제하기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteService를 참조하세요.

delete-task-definitions

다음 코드 예시는 delete-task-definitions의 사용 방법을 보여 줍니다.

AWS CLI

작업 정의를 삭제하려면

다음 `delete-task-definitions` 예시에서는 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 ECS 개발자 안내서의 Amazon ECS 태스크 정의를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteTaskDefinitions 섹션을 참조하세요.

delete-task-set

다음 코드 예시는 delete-task-set의 사용 방법을 보여 줍니다.

AWS CLI

작업 세트를 삭제하려면

다음 delete-task-set 예시에서는 작업 세트를 삭제하는 방법을 보여줍니다. 작업 세트가 0으로 규모 조정되지 않은 경우에도 --force 파라미터를 포함하여 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
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteTaskSet 섹션을 참조하세요.

deregister-container-instance

다음 코드 예시는 deregister-container-instance의 사용 방법을 보여 줍니다.

AWS CLI

클러스터에서 컨테이너 인스턴스 등록을 취소하려면

다음 `deregister-container-instance` 예시에서는 지정된 클러스터에서 컨테이너 인스턴스를 등록 취소합니다. 컨테이너 인스턴스에서 아직 실행 중인 작업이 있는 경우 등록을 취소하기 전에 해당 작업을 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 Deregister a Container Instance 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeregisterContainerInstance 섹션을 참조하세요.

`deregister-task-definition`

다음 코드 예시는 `deregister-task-definition`의 사용 방법을 보여 줍니다.

AWS CLI

작업 정의를 등록 취소하려면

다음 `deregister-task-definition` 예시에서는 기본 리전에서 `curler` 작업 정의의 첫 번째 개정을 등록 취소합니다.

```
aws ecs deregister-task-definition --task-definition curler:1
```

결과 출력에서 작업 정의 상태는 `INACTIVE`를 표시합니다.

```
{
    "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 ECS 개발자 안내서의 Amazon ECS 태스크 정의를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeregisterTaskDefinition 섹션을 참조하세요.

describe-capacity-providers

다음 코드 예시는 describe-capacity-providers의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 모든 용량 공급자를 설명하려면

다음 describe-capacity-providers 예시에서는 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 [Cluster capacity providers](클러스터 쿼리 언어)를 참조하세요.

예시 2: 특정 용량 공급자를 설명하려면

다음 `describe-capacity-providers` 예시에서는 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 Cluster capacity providers(클러스터 쿼리 언어)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeCapacityProviders 섹션을 참조하세요.

describe-clusters

다음 코드 예시는 `describe-clusters`의 사용 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 ECS 개발자 안내서의 Amazon ECS 클러스터를 참조하세요.

예 2: 첨부 파일 옵션을 사용하여 클러스터를 설명하는 방법

다음 `describe-clusters` 예시에서는 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 크기 조정 정책과 같은 리소스는 asp 또는 as_policy ATCHEMENTS로 표시됩니다.

```
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 ECS 개발자 안내서의 Amazon ECS 클러스터를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeClusters를 참조하세요.

describe-container-instances

다음 코드 예시는 describe-container-instances의 사용 방법을 보여 줍니다.

AWS CLI

컨테이너 인스턴스를 설명하려면

다음 describe-container-instances 예시에서는 컨테이너 인스턴스 UUID를 식별자로 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 ECS 개발자 안내서의 Amazon ECS 컨테이너 인스턴스 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeContainerInstances 섹션을 참조하세요.

describe-service-deployments

다음 코드 예시는 describe-service-deployments의 사용 방법을 보여 줍니다.

AWS CLI

서비스 배포 세부 정보를 설명하려면

다음 describe-service-deployments 예제에서는 arn:aws:ecs:us-east-1:123456789012:service-deployment/example-cluster/example-service/ejGvqq2ilnbKT9qj0vLJe ARN을 사용한 서비스 배포에 대한 서비스 배포 세부 정보를 반환합니다.

```
aws ecs describe-service-deployments \
    --service-deployment-arn arn:aws:ecs:us-east-1:123456789012:service-deployment/
example-cluster/example-service/ejGvqq2ilnbKT9qj0vLJe
```

출력:

```
{
    "serviceDeployments": [
        {
            "serviceDeploymentArn": "arn:aws:ecs:us-east-1:123456789012:service-
deployment/example-cluster/example-service/ejGvqq2ilnbKT9qj0vLJe",
            "serviceArn": "arn:aws:ecs:us-east-1:123456789012:service/example-
cluster/example-service",
            "clusterArn": "arn:aws:ecs:us-east-1:123456789012:cluster/example-
cluster",
            "createdAt": "2024-10-31T08:03:30.917000-04:00",
            "startedAt": "2024-10-31T08:03:32.510000-04:00",
            "finishedAt": "2024-10-31T08:05:04.527000-04:00",
            "updatedAt": "2024-10-31T08:05:04.527000-04:00",
            "sourceServiceRevisions": [],
            "targetServiceRevision": {
                "arn": "arn:aws:ecs:us-east-1:123456789012:service-revision/example-
cluster/example-service/1485800978477494678",
                "requestedTaskCount": 1,
```

```
                "runningTaskCount": 1,
                "pendingTaskCount": 0
            },
            "status": "SUCCESSFUL",
            "deploymentConfiguration": {
                "deploymentCircuitBreaker": {
                    "enable": true,
                    "rollback": true
                },
                "maximumPercent": 200,
                "minimumHealthyPercent": 100,
                "alarms": {
                    "alarmNames": [],
                    "rollback": false,
                    "enable": false
                }
            },
            "deploymentCircuitBreaker": {
                "status": "MONITORING_COMPLETE",
                "failureCount": 0,
                "threshold": 3
            },
            "alarms": {
                "status": "DISABLED"
            }
        }
    ],
    "failures": []
}
```

자세한 내용은 Amazon ECS 개발자 안내서의 Amazon ECS 서비스 배포를 사용하여 서비스 기록 보기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeServiceDeployments 섹션을 참조하세요.

describe-service-revisions

다음 코드 예시는 describe-service-revisions의 사용 방법을 보여 줍니다.

AWS CLI

서비스 개정 세부 정보를 설명하려면

다음 describe-service-revisions 예제에서는 arn:aws:ecs:us-east-1:123456789012:service-revision/example-cluster/example-service/1485800978477494678 ARN을 사용하여 서비스 개정에 대한 서비스 개정 세부 정보를 반환합니다.

```
aws ecs describe-service-revisions \
    --service-revision-arns arn:aws:ecs:us-east-1:123456789012:service-revision/
example-cluster/example-service/1485800978477494678
```

출력:

```
{
    "serviceRevisions": [
        {
            "serviceRevisionArn": "arn:aws:ecs:us-east-1:123456789012:service-
revision/example-cluster/example-service/1485800978477494678",
            "serviceArn": "arn:aws:ecs:us-east-1:123456789012:service/example-
cluster/example-service",
            "clusterArn": "arn:aws:ecs:us-east-1:123456789012:cluster/example-
cluster",
            "taskDefinition": "arn:aws:ecs:us-east-1:123456789012:task-definition/
webserver:5",
            "capacityProviderStrategy": [
                {
                    "capacityProvider": "FARGATE",
                    "weight": 1,
                    "base": 0
                }
            ],
            "platformVersion": "1.4.0",
            "platformFamily": "Linux",
            "networkConfiguration": {
                "awsvpcConfiguration": {
                    "subnets": [
                        "subnet-0d0eab1bb38d5ca64",
                        "subnet-0db5010045995c2d5"
                    ],
                    "securityGroups": [
                        "sg-02556bf85a191f59a"
                    ],
                    "assignPublicIp": "ENABLED"
                }
```

```
            },
            "containerImages": [
                {
                    "containerName": "aws-otel-collector",
                    "imageDigest": 
 "sha256:7a1b3560655071bcacd66902c20ebe9a69470d5691fe3bd36baace7c2f3c4640",
                    "image": "public.ecr.aws/aws-observability/aws-otel-
collector:v0.32.0"
                },
                {
                    "containerName": "web",
                    "imageDigest": 
 "sha256:28402db69fec7c17e179ea87882667f1e054391138f77ffaf0c3eb388efc3ffb",
                    "image": "nginx"
                }
            ],
            "guardDutyEnabled": false,
            "serviceConnectConfiguration": {
                "enabled": false
            },
            "createdAt": "2024-10-31T08:03:29.302000-04:00"
        }
    ],
    "failures": []
}
```

자세한 내용은 Amazon ECS 개발자 안내서의 Amazon ECS 서비스 개정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeServiceRevisions 섹션을 참조하세요.

describe-services

다음 코드 예시는 describe-services의 사용 방법을 보여 줍니다.

AWS CLI

서비스를 설명하려면

다음 describe-services 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeServices를 참조하세요.

describe-task-definition

다음 코드 예시는 describe-task-definition의 사용 방법을 보여 줍니다.

AWS CLI

작업 정의를 설명하려면

다음 describe-task-definition 예시에서는 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 ECS 개발자 안내서의 Amazon ECS 태스크 정의를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeTaskDefinition 섹션을 참조하세요.

describe-task-sets

다음 코드 예시는 describe-task-sets의 사용 방법을 보여 줍니다.

AWS CLI

작업 세트를 설명하려면

다음 describe-task-sets 예시에서는 외부 배포자를 사용하는 서비스에서 설정된 작업을 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": []
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeTaskSets 섹션을 참조하세요.

describe-tasks

다음 코드 예시는 describe-tasks의 사용 방법을 보여 줍니다.

AWS CLI

예 1: 단일 태스크를 설명하는 방법

다음 describe-tasks 예시에서는 클러스터의 태스크 세부 정보를 검색합니다. 태스크의 ID 또는 전체 ARN을 사용하여 태스크를 지정할 수 있습니다. 이 예시에서는 태스크의 전체 ARN을 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 ECS 개발자 안내서의 Amazon ECS 태스크 정의를 참조하세요.

예 2: 여러 태스크를 설명하는 방법

다음 describe-tasks 예시에서는 클러스터에 있는 여러 태스크의 세부 정보를 검색합니다. 태스크의 ID 또는 전체 ARN을 사용하여 태스크를 지정할 수 있습니다. 이 예시에서는 태스크의 전체 ID를 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 ECS 개발자 안내서의 Amazon ECS 태스크 정의를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeTasks를 참조하세요.

execute-command

다음 코드 예시는 execute-command의 사용 방법을 보여 줍니다.

AWS CLI

대화형 /bin/sh 명령을 실행하려면

다음 execute-command 예시에서는 MyContainer라는 컨테이너에 대해 대화형 /bin/sh 명령을 실행하여 ID가 arn:aws:ecs:us-east-1:123456789012:task/MyCluster/d789e94343414c25b9f6bd59eEXAMPLE인 작업에 대해 실행합니다.

```
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 ECS 개발자 안내서의 디버깅에 Amazon ECS Exec 사용 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ExecuteCommand 섹션을 참조하세요.

get-task-protection

다음 코드 예시는 get-task-protection의 사용 방법을 보여 줍니다.

AWS CLI

ECS 서비스에서 작업의 보호 상태 검색

다음 get-task-protection에서는 Amazon ECS 서비스에 속하는 ECS 작업의 보호 상태를 제공합니다.

```
aws ecs get-task-protection \
    --cluster ECS-project-update-cluster \
    --tasks c43ed3b1331041f289316f958adb6a24
```

출력:

```
{
    "protectedTasks": [
        {
            "taskArn": "arn:aws:ecs:us-west-2:123456789012:task/
c43ed3b1331041f289316f958adb6a24",
            "protectionEnabled": false
        }
    ],
    "failures": []
}
```

작업 보호에 대한 자세한 내용은 Amazon ECS 개발자 안내서의 스케일 인 이벤트로 인해 Amazon ECS 작업이 종료되지 않도록 보호를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetTaskProtection을 참조하세요.

list-account-settings

다음 코드 예시는 list-account-settings의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 계정의 계정 설정을 보려면

다음 list-account-settings 예시에서는 계정에 대한 유효 계정 설정을 표시합니다.

```
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"
        }
    ]
}
```

예시 2: 특정 IAM 사용자 또는 IAM 역할에 대한 계정 설정을 보려면

다음 `list-account-settings` 예시에서는 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 ECS 개발자 안내서의 Amazon 리소스 이름(ARN) 및 ID 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListAccountSettings 섹션을 참조하세요.

`list-attributes`

다음 코드 예시는 `list-attributes`의 사용 방법을 보여 줍니다.

AWS CLI

특정 속성을 포함하는 컨테이너 인스턴스를 나열하려면

다음 예시에서는 기본 클러스터에 `stack=production` 속성이 있는 컨테이너 인스턴스의 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 ECS 개발자 안내서의 Amazon ECS 컨테이너 에이전트 구성 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListAttributes 섹션을 참조하세요.

`list-clusters`

다음 코드 예시는 `list-clusters`의 사용 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 ECS 개발자 안내서의 Amazon ECS 클러스터를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListClusters를 참조하세요.

`list-container-instances`

다음 코드 예시는 `list-container-instances`의 사용 방법을 보여 줍니다.

AWS CLI

클러스터의 컨테이너 인스턴스를 나열하려면

다음 `list-container-instances` 예시에서는 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 ECS 개발자 안내서의 [Amazon ECS 컨테이너 인스턴스](#) 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ListContainerInstances](#) 섹션을 참조하세요.

list-service-deployments

다음 코드 예시는 `list-service-deployments`의 사용 방법을 보여 줍니다.

AWS CLI

서비스 배포를 나열하려면

다음 `list-service-deployments` 예제에서는 `example-service`라는 서비스에 대한 서비스 배포를 검색합니다.

```
aws ecs list-service-deployments \
    --service arn:aws:ecs:us-east-1:123456789012:service/example-cluster/example-service
```

출력:

```
{
    "serviceDeployments": [
        {
            "serviceDeploymentArn": "arn:aws:ecs:us-east-1:123456789012:service-deployment/example-cluster/example-service/ejGvqq2ilnbKT9qj0vLJe",
            "serviceArn": "arn:aws:ecs:us-east-1:123456789012:service/example-cluster/example-service",
            "clusterArn": "arn:aws:ecs:us-east-1:123456789012:cluster/example-cluster",
            "startedAt": "2024-10-31T08:03:32.510000-04:00",
            "createdAt": "2024-10-31T08:03:30.917000-04:00",
            "finishedAt": "2024-10-31T08:05:04.527000-04:00",
            "targetServiceRevisionArn": "arn:aws:ecs:us-east-1:123456789012:service-revision/example-cluster/example-service/1485800978477494678",
            "status": "SUCCESSFUL"
        }
    ]
```

```
}
```

자세한 내용은 Amazon ECS 개발자 안내서의 Amazon ECS 서비스 배포를 사용하여 서비스 기록 보기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListServiceDeployments를 참조하세요.

list-services-by-namespace

다음 코드 예시는 list-services-by-namespace의 사용 방법을 보여 줍니다.

AWS CLI

네임스페이스에 있는 서비스를 나열하려면

다음 list-services-by-namespace 예시에서는 기본 리전에서 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 Connect 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListServicesByNamespace 섹션을 참조하세요.

list-services

다음 코드 예시는 list-services의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListServices를 참조하세요.

`list-tags-for-resource`

다음 코드 예시는 `list-tags-for-resource`의 사용 방법을 보여 줍니다.

AWS CLI

리소스의 태그를 나열하려면

다음 `list-tags-for-resource` 예시에서는 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [ListTagsForResource](#) 섹션을 참조하세요.

`list-task-definition-families`

다음 코드 예시는 `list-task-definition-families`의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 등록된 작업 정의 패밀리를 나열하려면

다음 `list-task-definition-families` 예시에서는 등록된 모든 작업 정의 패밀리를 나열합니다.

```
aws ecs list-task-definition-families
```

출력:

```
{
    "families": [
        "node-js-app",
        "web-timer",
        "hpcc",
        "hpcc-c4-8xlarge"
    ]
}
```

예시 2: 등록된 작업 정의 패밀리를 필터링하려면

다음 `list-task-definition-families` 예시에서는 'hpcc'로 시작하는 작업 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 파라미터 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTaskDefinitionFamilies 섹션을 참조하세요.

list-task-definitions

다음 코드 예시는 `list-task-definitions`의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 등록된 작업 정의를 나열하려면

다음 `list-task-definitions` 예시에서는 등록된 모든 작업 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"
    ]
}
```

예시 2: 패밀리에 등록된 작업 정의를 나열하려면

다음 list-task-definitions 예시에서는 지정된 패밀리의 작업 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 ECS 개발자 안내서의 Amazon ECS 태스크 정의를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTaskDefinitions 섹션을 참조하세요.

list-tasks

다음 코드 예시는 `list-tasks`의 사용 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 예시에서는 컨테이너 인스턴스 UUID를 필터로 사용하여 컨테이너 인스턴스의 태스크를 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 ECS 개발자 안내서의 Amazon ECS 태스크 정의를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTasks를 참조하세요.

put-account-setting-default

다음 코드 예시는 put-account-setting-default의 사용 방법을 보여 줍니다.

AWS CLI

기본 계정 설정을 수정하려면

다음 put-account-setting-default 예시에서는 계정의 모든 IAM 사용자 또는 역할에 대한 기본 계정 설정을 수정합니다. IAM 사용자 또는 역할이 이러한 설정을 명시적으로 재정의하는 경우를 제외하고 이러한 변경 사항이 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 ECS 개발자 안내서의 Amazon 리소스 이름(ARN) 및 ID 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutAccountSettingDefault 섹션을 참조하세요.

put-account-setting

다음 코드 예시는 put-account-setting의 사용 방법을 보여 줍니다.

AWS CLI

IAM 사용자 계정의 계정 설정을 수정하려면

다음 put-account-setting 예제에서는 IAM 사용자 계정의 enhanced에 대한 containerInsights 계정 설정을 구성합니다. 이렇게 하면 향상된 관찰성을 갖춘 Container Insights가 켜집니다.

```
aws ecs put-account-setting \
    --name containerInsights \
    --value enhanced
```

출력:

```
{
    "setting": {
        "name": "containerInsights",
        "value": "enhanced",
        "principalArn": "arn:aws:iam::123456789012:user/johndoe",
        "type": "user"
    }
}
```

자세한 내용은 Amazon ECS 개발자 안내서의 계정 설정 수정 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutAccountSetting 섹션을 참조하세요.

put-account-settings

다음 코드 예시는 put-account-settings의 사용 방법을 보여 줍니다.

AWS CLI

IAM 사용자 또는 IAM 역할의 계정 설정을 수정하려면

다음 put-account-setting 예시에서는 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 PutAccountSettings 섹션을 참조하세요.

put-attributes

다음 코드 예시는 put-attributes의 사용 방법을 보여 줍니다.

AWS CLI

속성을 만들고 Amazon ECS 리소스와 연결하려면

다음 put-attributes는 컨테이너 인스턴스에 스택 및 값 생산이라는 이름의 속성을 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 PutAttributes 섹션을 참조하세요.

put-cluster-capacity-providers

다음 코드 예시는 put-cluster-capacity-providers의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 클러스터에 기존 용량 공급자를 추가하려면

다음 put-cluster-capacity-providers 예시에서는 클러스터에 기존 용량 공급자를 추가합니다. create-capacity-provider 명령을 사용하여 용량 공급자를 생성합니다. describe-clusters 명령은 클러스터와 관련된 현재 용량 공급자 및 기본 용량 공급자 전략을 설명하는 데 사용됩니다. 클러스터에 새 용량 공급자를 추가할 때는 클러스터와 연결하려는 새 용량 공급자 외에 기존의 모든 용량 공급자를 지정해야 합니다. 또한 클러스터와 연결할 기본 용량 공급자 전략을 지정해야 합니다. 이 예시에서는 MyCluster 클러스터에 연결된 MyCapacityProvider1 용량 공급자가 있으며, MyCapacityProvider2 용량 공급자를 추가하고 기본 용량 제공업체 전략에 포함하여 작업이 두 용량 공급자에 균등하게 분산되도록 합니다.

```
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 [Cluster capacity providers](클러스터 쿼리 언어)를 참조하세요.

예시 2: 클러스터에서 용량 공급자를 제거하려면

다음 put-cluster-capacity-providers 예시에서는 클러스터에서 용량 공급자를 제거합니다. describe-clusters 명령은 클러스터와 연결된 현재 용량 공급자를 설명하는 데 사용됩니다. 클러스터에서 용량 공급자를 제거할 때는 클러스터와의 연결 상태를 유지할 용량 공급자와 클러스터와 연결할 기본 용량 공급자 전략을 지정해야 합니다. 이 예시에서는 클러스터에 MyCapacityProvider1 및 MyCapacityProvider2 용량 공급자가 연결되어 있고 MyCapacityProvider2 용량 공급자를 제거해야 하므로 업데이트된 기본 용량 공급자 전략과 함께 명령에 MyCapacityProvider1만 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 [Cluster capacity providers](클러스터 쿼리 언어)를 참조하세요.

예시 3: 클러스터에서 모든 용량 공급자 제거

다음 `put-cluster-capacity-providers` 예시에서는 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 Cluster capacity providers(클러스터 쿼리 언어)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutClusterCapacityProviders 섹션을 참조하세요.

`register-task-definition`

다음 코드 예시는 `register-task-definition`의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: JSON 파일에 작업 정의를 등록하려면

다음 `register-task-definition` 예시에서는 지정된 패밀리에 작업 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.json`의 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 예시 섹션을 참조하세요.

예시 2: JSON 문자열 파라미터로 작업 정의 등록

다음 `register-task-definition` 예시에서는 이스케이프된 큰따옴표로 묶인 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 예시와 동일합니다.

자세한 내용은 Amazon ECS 개발자 안내서의 작업 정의 생성 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RegisterTaskDefinition 섹션을 참조하세요.

run-task

다음 코드 예시는 run-task의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 기본 클러스터에서 작업 실행

다음 run-task 예시에서는 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 애플리케이션을 독립 실행형 작업으로 실행을 참조하세요.

예시 2: 독립 실행형 작업에 대한 Amazon EBS 볼륨 구성

다음 `run-task` 예시에서는 기본 클러스터의 Fargate 작업에 대해 암호화된 Amazon EBS 볼륨을 구성합니다. `AmazonECSInfrastructureRolePolicyForVolumes` 관리형 정책과 연결된 Amazon ECS 인프라 역할이 구성되어 있어야 합니다. `run-task` 요청에서와 동일한 볼륨 이름으로 작업 정의를 지정해야 합니다. 이 예시에서는 `--cli-input-json` 옵션과 `ebs.json`이라는 JSON 입력 파일을 사용합니다.

```
aws ecs run-task \
    --cli-input-json file://ebs.json
```

`ebs.json`의 콘텐츠:

```
{
   "cluster": "default",
   "taskDefinition": "mytaskdef",
   "launchType": "FARGATE",
   "networkConfiguration":{
        "awsvpcConfiguration":{
            "assignPublicIp": "ENABLED",
            "securityGroups": ["sg-12344321"],
            "subnets":["subnet-12344321"]
        }
    },
   "volumeConfigurations": [
        {
            "name": "myEBSVolume",
            "managedEBSVolume": {
                "volumeType": "gp3",
                "sizeInGiB": 100,
                "roleArn":"arn:aws:iam::1111222333:role/ecsInfrastructureRole",
                "encrypted": true,
                "kmsKeyId":
 "arn:aws:kms:region:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
            }
        }
    ]
}
```

출력:

```
{
```

```
    "tasks": [
        {
            "attachments": [
                {
                    "id": "ce868693-15ca-4083-91ac-f782f64000c9",
                    "type": "ElasticNetworkInterface",
                    "status": "PRECREATED",
                    "details": [
                        {
                        "name": "subnetId",
                        "value": "subnet-070982705451dad82"
                        }
                    ]
                },
                {
                    "id": "a17ed863-786c-4372-b5b3-b23e53f37877",
                    "type": "AmazonElasticBlockStorage",
                    "status": "CREATED",
                    "details": [
                        {
                            "name": "roleArn",
                            "value": "arn:aws:iam::123456789012:role/
ecsInfrastructureRole"
                        },
                        {
                            "name": "volumeName",
                            "value": "myEBSVolume"
                        },
                        {
                            "name": "deleteOnTermination",
                            "value": "true"
                        }
                    ]
                }
            ],
            "attributes": [
                {
                    "name": "ecs.cpu-architecture",
                    "value": "x86_64"
                }
            ],
            "availabilityZone": "us-west-2b",
            "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/default",
            "containers": [
```

```
                {
                    "containerArn": "arn:aws:ecs:us-west-2:123456789012:container/
default/7f1fbd3629434cc4b82d72d2f09b67c9/e21962a2-f328-4699-98a3-5161ac2c186a",
                    "taskArn": "arn:aws:ecs:us-west-2:123456789012:task/
default/7f1fbd3629434cc4b82d72d2f09b67c9",
                    "name": "container-using-ebs",
                    "image": "amazonlinux:2",
                    "lastStatus": "PENDING",
                    "networkInterfaces": [],
                    "cpu": "0"
                }
            ],
            "cpu": "1024",
            "createdAt": "2025-01-23T10:29:46.650000-06:00",
            "desiredStatus": "RUNNING",
            "enableExecuteCommand": false,
            "group": "family:mytaskdef",
            "lastStatus": "PROVISIONING",
            "launchType": "FARGATE",
            "memory": "3072",
            "overrides": {
                "containerOverrides": [
                    {
                        "name": "container-using-ebs"
                    }
                ],
                "inferenceAcceleratorOverrides": []
            },
            "platformVersion": "1.4.0",
            "platformFamily": "Linux",
            "tags": [],
            "taskArn": "arn:aws:ecs:us-west-2:123456789012:task/
default/7f1fbd3629434cc4b82d72d2f09b67c9",
            "taskDefinitionArn": "arn:aws:ecs:us-west-2:123456789012:task-
definition/mytaskdef:4",
            "version": 1,
            "ephemeralStorage": {
                "sizeInGiB": 20
            },
            "fargateEphemeralStorage": {
                "sizeInGiB": 20
            }
        }
    ],
```

```
    "failures": []
}
```

자세한 내용은 Amazon ECS 개발자 안내서의 Amazon ECS에서 Amazon EBS 볼륨 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RunTask 섹션을 참조하세요.

start-task

다음 코드 예시는 start-task의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 새 작업 시작

다음 start-task 예시에서는 기본 클러스터의 지정된 컨테이너 인스턴스에서 sleep360 작업 정의의 최신 버전을 사용하여 작업을 시작합니다.

```
aws ecs start-task \
    --task-definition sleep360 \
    --container-instances 765936fadbdd46b5991a4bd70c2a43d4
```

출력:

```
{
    "tasks": [
        {
            "taskArn": "arn:aws:ecs:us-west-2:123456789012:task/
default/666fdccc2e2d4b6894dd422f4eeee8f8",
            "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/default",
            "taskDefinitionArn": "arn:aws:ecs:us-west-2:123456789012:task-
definition/sleep360:3",
            "containerInstanceArn": "arn:aws:ecs:us-west-2:123456789012:container-
instance/default/765936fadbdd46b5991a4bd70c2a43d4",
            "overrides": {
                "containerOverrides": [
                    {
                        "name": "sleep"
                    }
                ]
            },
            "lastStatus": "PENDING",
```

```
            "desiredStatus": "RUNNING",
            "cpu": "128",
            "memory": "128",
            "containers": [
                {
                    "containerArn": "arn:aws:ecs:us-
west-2:123456789012:container/75f11ed4-8a3d-4f26-a33b-ad1db9e02d41",
                    "taskArn": "arn:aws:ecs:us-west-2:123456789012: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 내용은 Amazon ECS 개발자 안내서의 Amazon ECS에서 컨테이너 예약을 참조하세요.

예시 2: 작업 시작 시 Amazon EBS 볼륨 구성

다음 `start-task` 예시에서는 지정된 컨테이너 인스턴스의 작업에 대해 암호화된 Amazon EBS 볼륨을 구성합니다. `AmazonECSInfrastructureRolePolicyForVolumes` 관리형 정책과 연결된 Amazon ECS 인프라 역할이 구성되어 있어야 합니다. `start-task` 요청에서와 동일한 볼륨 이름으로 작업 정의를 지정해야 합니다. 이 예시에서는 `--cli-input-json` 옵션과 다음 콘텐츠가 포함된 `ebs.json`이라는 JSON 입력 파일을 사용합니다.

```
aws ecs start-task \
    --cli-input-json file://ebs.json \
    --container-instances 765936fadbdd46b5991a4bd70c2a43d4
```

`ebs.json`의 콘텐츠:

```
{
   "cluster": "default",
   "taskDefinition": "mytaskdef",
   "networkConfiguration":{
        "awsvpcConfiguration":{
            "assignPublicIp": "ENABLED",
            "securityGroups": ["sg-12344321"],
            "subnets":["subnet-12344321"]
        }
    },
   "volumeConfigurations": [
        {
            "name": "myEBSVolume",
            "managedEBSVolume": {
                "volumeType": "gp3",
                "sizeInGiB": 100,
                "roleArn":"arn:aws:iam::123456789012:role/ecsInfrastructureRole",
                "encrypted": true,
                "kmsKeyId":
 "arn:aws:kms:region:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab"
            }
        }
    ]
}
```

출력:

```
{
    "tasks": [
        {
            "attachments": [
                {
                    "id": "aea29489-9dcd-49f1-8164-4d91566e1113",
                    "type": "ElasticNetworkInterface",
                    "status": "PRECREATED",
                    "details": [
                        {
                            "name": "subnetId",
                            "value": "subnet-12344321"
                        }
                    ]
                },
                {
```

```
                    "id": "f29e1222-9a1e-410f-b499-a12a7cd6d42e",
                    "type": "AmazonElasticBlockStorage",
                    "status": "CREATED",
                    "details": [
                        {
                            "name": "roleArn",
                            "value": "arn:aws:iam::123456789012:role/
ecsInfrastructureRole"
                        },
                        {
                            "name": "volumeName",
                            "value": "myEBSVolume"
                        },
                        {
                            "name": "deleteOnTermination",
                            "value": "true"
                        }
                    ]
                }
            ],
            "attributes": [
                {
                    "name": "ecs.cpu-architecture",
                    "value": "arm64"
                }
            ],
            "availabilityZone": "us-west-2c",
            "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/default",
            "containerInstanceArn": "arn:aws:ecs:us-west-2:123456789012:container-
instance/default/765936fadbdd46b5991a4bd70c2a43d4",
            "containers": [
                {
                    "containerArn": "arn:aws:ecs:us-west-2:123456789012:container/
default/bb122ace3ed84add92c00a351a03c69e/a4a9ed10-51c7-4567-9653-50e71b94f867",
                    "taskArn": "arn:aws:ecs:us-west-2:123456789012:task/default/
bb122ace3ed84add92c00a351a03c69e",
                    "name": "container-using-ebs",
                    "image": "amazonlinux:2",
                    "lastStatus": "PENDING",
                    "networkInterfaces": [],
                    "cpu": "0"
                }
            ],
            "cpu": "1024",
```

```
            "createdAt": "2025-01-23T14:51:05.191000-06:00",
            "desiredStatus": "RUNNING",
            "enableExecuteCommand": false,
            "group": "family:mytaskdef",
            "lastStatus": "PROVISIONING",
            "launchType": "EC2",
            "memory": "3072",
            "overrides": {
                "containerOverrides": [
                    {
                        "name": "container-using-ebs"
                    }
                ],
                "inferenceAcceleratorOverrides": []
            },
             "tags": [],
            "taskArn": "arn:aws:ecs:us-west-2:123456789012:task/default/
bb122ace3ed84add92c00a351a03c69e",
            "taskDefinitionArn": "arn:aws:ecs:us-west-2:123456789012:task-
definition/mytaskdef:4",
            "version": 1
        }
    ],
    "failures": []
}
```

자세한 내용은 Amazon ECS 개발자 안내서의 Amazon ECS에서 Amazon EBS 볼륨 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartTask 섹션을 참조하세요.

stop-task

다음 코드 예시는 stop-task의 사용 방법을 보여 줍니다.

AWS CLI

작업을 중지하려면

다음 stop-task는 지정된 작업이 기본 클러스터에서 실행되지 않도록 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": []
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 StopTask 섹션을 참조하세요.

tag-resource

다음 코드 예시는 `tag-resource`의 사용 방법을 보여 줍니다.

AWS CLI

리소스에 태그를 지정하려면

다음 `tag-resource` 예시에서는 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` 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

`untag-resource`

다음 코드 예시에서는 `untag-resource`의 사용 방법을 보여줍니다.

AWS CLI

리소스에서 태그를 제거하려면

다음 `untag-resource` 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

update-cluster-settings

다음 코드 예시는 update-cluster-settings의 사용 방법을 보여 줍니다.

AWS CLI

클러스터의 설정을 수정하려면

다음 update-cluster-settings 예제에서는 MyCluster라는 클러스터를 만들고 향상된 관찰성을 갖춘 CloudWatch Container Insights를 사용 설정합니다.

```
aws ecs update-cluster-settings \
    --cluster MyCluster \
    --settings name=containerInsights,value=enhanced
```

출력:

```
{
    "cluster": {
        "clusterArn": "arn:aws:ecs:us-esat-1:123456789012:cluster/MyCluster",
        "clusterName": "default",
        "status": "ACTIVE",
        "registeredContainerInstancesCount": 0,
        "runningTasksCount": 0,
        "pendingTasksCount": 0,
        "activeServicesCount": 0,
        "statistics": [],
        "tags": [],
        "settings": [
            {
                "name": "containerInsights",
                "value": "enhanced"
            }
        ]
    }
}
```

자세한 내용은 Amazon ECS 개발자 안내서의 계정 설정 수정 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateClusterSettings 섹션을 참조하세요.

`update-cluster`

다음 코드 예시는 `update-cluster`의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: containerInsights 활성화하는 ECS 클러스터 업데이트

다음 `update-cluster`에서는 이미 생성된 클러스터에서 containerInsights 값을 `enabled`로 업데이트합니다. 기본적으로는 비활성화되어 있습니다.

```
aws ecs update-cluster \
    --cluster ECS-project-update-cluster \
    --settings name=containerInsights,value=enabled
```

출력:

```
"cluster": {
    "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/ECS-project-update-
cluster",
    "clusterName": "ECS-project-update-cluster",
    "status": "ACTIVE",
    "registeredContainerInstancesCount": 0,
    "runningTasksCount": 0,
    "pendingTasksCount": 0,
    "activeServicesCount": 0,
    "statistics": [],
    "tags": [],
    "settings": [
        {
            "name": "containerInsights",
            "value": "enabled"
        }
    ],
    "capacityProviders": [
        "Infra-ECS-Cluster-ECS-project-update-cluster-d6bb6d5b-
EC2CapacityProvider-3fIpdkLywwFt"
    ],
    "defaultCapacityProviderStrategy": [
        {
            "capacityProvider": "Infra-ECS-Cluster-ECS-project-update-cluster-
d6bb6d5b-EC2CapacityProvider-3fIpdkLywwFt",
            "weight": 1,
```

```
            "base": 0
        }
    ],
    "attachments": [
        {
            "id": "069d002b-7634-42e4-b1d4-544f4c8f6380",
            "type": "as_policy",
            "status": "CREATED",
            "details": [
                {
                    "name": "capacityProviderName",
                    "value": "Infra-ECS-Cluster-ECS-project-update-cluster-d6bb6d5b-
EC2CapacityProvider-3fIpdkLywwFt"
                },
                {
                    "name": "scalingPolicyName",
                    "value": "ECSManagedAutoScalingPolicy-152363a6-8c65-484c-
b721-42c3e070ae93"
                }
            ]
        },
        {
            "id": "08b5b6ca-45e9-4209-a65d-e962a27c490a",
            "type": "managed_draining",
            "status": "CREATED",
            "details": [
                {
                    "name": "capacityProviderName",
                    "value": "Infra-ECS-Cluster-ECS-project-update-cluster-d6bb6d5b-
EC2CapacityProvider-3fIpdkLywwFt"
                },
                {
                    "name": "autoScalingLifecycleHookName",
                    "value": "ecs-managed-draining-termination-hook"
                }
            ]
        },
        {
            "id": "45d0b36f-8cff-46b6-9380-1288744802ab",
            "type": "sc",
            "status": "ATTACHED",
            "details": []
        }
    ],
```

```
    "attachmentsStatus": "UPDATE_COMPLETE",
    "serviceConnectDefaults": {
        "namespace": "arn:aws:servicediscovery:us-west-2:123456789012:namespace/ns-igwrsylmy3kwvcdx"
    }
}
```

예제 2: ECS 클러스터를 업데이트하여 기본 Service Connect 네임스페이스 설정

다음 `update-cluster`에서는 기본 Service Connect 네임스페이스를 설정하여 ECS 클러스터를 업데이트합니다.

```
aws ecs update-cluster \
    --cluster ECS-project-update-cluster \
    --service-connect-defaults namespace=test
```

출력:

```
 {
     "cluster": {
         "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/ECS-project-update-cluster",
         "clusterName": "ECS-project-update-cluster",
         "status": "ACTIVE",
         "registeredContainerInstancesCount": 0,
         "runningTasksCount": 0,
         "pendingTasksCount": 0,
         "activeServicesCount": 0,
         "statistics": [],
         "tags": [],
         "settings": [
             {
                 "name": "containerInsights",
                 "value": "enabled"
             }
         ],
         "capacityProviders": [
             "Infra-ECS-Cluster-ECS-project-update-cluster-d6bb6d5b-EC2CapacityProvider-3fIpdkLywwFt"
         ],
         "defaultCapacityProviderStrategy": [
             {
```

```
                "capacityProvider": "Infra-ECS-Cluster-ECS-project-update-cluster-
d6bb6d5b-EC2CapacityProvider-3fIpdkLywwFt",
                "weight": 1,
                "base": 0
            }
        ],
        "attachments": [
            {
                "id": "069d002b-7634-42e4-b1d4-544f4c8f6380",
                "type": "as_policy",
                "status": "CREATED",
                "details": [
                    {
                        "name": "capacityProviderName",
                        "value": "Infra-ECS-Cluster-ECS-project-update-cluster-
d6bb6d5b-EC2CapacityProvider-3fIpdkLywwFt"
                    },
                    {
                        "name": "scalingPolicyName",
                        "value": "ECSManagedAutoScalingPolicy-152363a6-8c65-484c-
b721-42c3e070ae93"
                    }
                ]
            },
            {
                "id": "08b5b6ca-45e9-4209-a65d-e962a27c490a",
                "type": "managed_draining",
                "status": "CREATED",
                "details": [
                    {
                        "name": "capacityProviderName",
                        "value": "Infra-ECS-Cluster-ECS-project-update-cluster-
d6bb6d5b-EC2CapacityProvider-3fIpdkLywwFt"
                    },
                    {
                        "name": "autoScalingLifecycleHookName",
                        "value": "ecs-managed-draining-termination-hook"
                    }
                ]
            },
            {
                "id": "45d0b36f-8cff-46b6-9380-1288744802ab",
                "type": "sc",
                "status": "DELETED",
```

```
                    "details": []
                },
                {
                    "id": "3e6890c3-609c-4832-91de-d6ca891b3ef1",
                    "type": "sc",
                    "status": "ATTACHED",
                    "details": []
                },
                {
                    "id": "961b8ec1-c2f1-4070-8495-e669b7668e90",
                    "type": "sc",
                    "status": "DELETED",
                    "details": []
                }
            ],
            "attachmentsStatus": "UPDATE_COMPLETE",
            "serviceConnectDefaults": {
                "namespace": "arn:aws:servicediscovery:us-
west-2:123456789012:namespace/ns-dtjmxqpfi46ht7dr"
            }
        }
}
```

Service Connect에 대한 자세한 내용은 Amazon ECS 개발자 안내서의 Service Connect를 사용하여 Amazon ECS 서비스를 짧은 이름으로 연결을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateCluster를 참조하세요.

update-container-agent

다음 코드 예시는 update-container-agent의 사용 방법을 보여 줍니다.

AWS CLI

Amazon ECS 컨테이너 인스턴스에서 컨테이너 에이전트를 업데이트하려면

다음 update-container-agent 예시에서는 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 ECS 개발자 안내서의 Amazon ECS 컨테이너 에이전트 업데이트 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateContainerAgent 섹션을 참조하세요.

update-container-instances-state

다음 코드 예시는 `update-container-instances-state`의 사용 방법을 보여 줍니다.

AWS CLI

컨테이너 인스턴스의 상태를 업데이트하려면

다음 `update-container-instances-state`는 지정된 컨테이너 인스턴스의 상태를 `DRAINING`으로 업데이트하여 해당 인스턴스가 등록되어 있는 클러스터에서 제거합니다.

```
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": []
}
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateContainerInstancesState 섹션을 참조하세요.

`update-service-primary-task-set`

다음 코드 예시는 `update-service-primary-task-set`의 사용 방법을 보여 줍니다.

AWS CLI

서비스의 기본 작업 세트를 업데이트하려면

다음 `update-service-primary-task-set` 예시에서는 지정된 서비스에 대한 기본 작업 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
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateServicePrimaryTaskSet 섹션을 참조하세요.

update-service

다음 코드 예시는 update-service의 사용 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 \
    --cluster test \
    --service my-http-service \
    --task-definition amazon-ecs-sample
```

출력:

```
{
    "service": {
        "serviceArn": "arn:aws:ecs:us-west-2:123456789012:service/test/my-http-service",
        "serviceName": "my-http-service",
        "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/test",
        "loadBalancers": [],
        "serviceRegistries": [],
        "status": "ACTIVE",
        "desiredCount": 2,
        "runningCount": 2,
        "pendingCount": 0,
        "launchType": "FARGATE",
        "platformVersion": "1.4.0",
```

```
        "platformFamily": "Linux",
        "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-definition/
amazon-ecs-sample:2",
        "deploymentConfiguration": {
            "deploymentCircuitBreaker": {
                "enable": true,
                "rollback": true
            },
            "maximumPercent": 200,
            "minimumHealthyPercent": 100,
            "alarms": {
                "alarmNames": [],
                "rollback": false,
                "enable": false
            }
        },
        "deployments": [
            {
                "id": "ecs-svc/7419115625193919142",
                "status": "PRIMARY",
                "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-
definition/amazon-ecs-sample:2",
                "desiredCount": 0,
                "pendingCount": 0,
                "runningCount": 0,
                "failedTasks": 0,
                "createdAt": "2025-02-21T13:26:02.734000-06:00",
                "updatedAt": "2025-02-21T13:26:02.734000-06:00",
                "launchType": "FARGATE",
                "platformVersion": "1.4.0",
                "platformFamily": "Linux",
                "networkConfiguration": {
                    "awsvpcConfiguration": {
                        "subnets": [
                            "subnet-12344321"
                        ],
                        "securityGroups": [
                            "sg-12344321"
                        ],
                        "assignPublicIp": "ENABLED"
                    }
                },
                "rolloutState": "IN_PROGRESS",
```

```
                "rolloutStateReason": "ECS deployment ecs-svc/7419115625193919142 in
 progress."
            },
            {
                "id": "ecs-svc/1709597507655421668",
                "status": "ACTIVE",
                "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-
definition/old-amazon-ecs-sample:4",
                "desiredCount": 2,
                "pendingCount": 0,
                "runningCount": 2,
                "failedTasks": 0,
                "createdAt": "2025-01-24T11:13:07.621000-06:00",
                "updatedAt": "2025-02-02T16:11:30.838000-06:00",
                "launchType": "FARGATE",
                "platformVersion": "1.4.0",
                "platformFamily": "Linux",
                "networkConfiguration": {
                    "awsvpcConfiguration": {
                        "subnets": [
                            "subnet-12344321"
                        ],
                        "securityGroups": [
                             "sg-12344321"
                        ],
                        "assignPublicIp": "ENABLED"
                    }
                },
                "rolloutState": "COMPLETED",
                "rolloutStateReason": "ECS deployment ecs-svc/1709597507655421668
 completed."
            }
        ],
        "roleArn": "arn:aws:iam::123456789012:role/aws-service-role/
ecs.amazonaws.com/AWSServiceRoleForECS",
        "events": [
            {
                "id": "e40b4d1c-80d9-4834-aaf3-6a268e530e17",
                "createdAt": "2025-02-21T10:31:26.037000-06:00",
                "message": "(my-http-service) has reached a steady state."
            },
            {
                "id": "6ac069ad-fc8b-4e49-a35d-b5574a964c8e",
                "createdAt": "2025-02-21T04:31:22.703000-06:00",
```

```
                "message": "(my-http-service) has reached a steady state."
            },
            {
                "id": "265f7d37-dfd1-4880-a846-ec486f341919",
                "createdAt": "2025-02-20T22:31:22.514000-06:00",
                "message": "(my-http-service) has reached a steady state."
            }
        ],
        "createdAt": "2024-10-30T17:12:43.218000-05:00",
        "placementConstraints": [],
        "placementStrategy": [],
        "networkConfiguration": {
            "awsvpcConfiguration": {
                "subnets": [
                    "subnet-12344321",
                ],
                "securityGroups": [
                    "sg-12344321"
                ],
                "assignPublicIp": "ENABLED"
            }
        },
        "healthCheckGracePeriodSeconds": 0,
        "schedulingStrategy": "REPLICA",
        "deploymentController": {
            "type": "ECS"
        },
        "createdBy": "arn:aws:iam::123456789012:role/AIDACKCEVSQ6C2EXAMPLE",
        "enableECSManagedTags": true,
        "propagateTags": "NONE",
        "enableExecuteCommand": false,
        "availabilityZoneRebalancing": "DISABLED"
    }
}
```

자세한 내용은 Amazon ECS 개발자 안내서의 [콘솔을 사용하여 Amazon ECS 서비스 업데이트](#)를 참조하세요.

예 2: 서비스의 태스크 수를 변경하는 방법

다음 `update-service` 예시에서는 `my-http-service` 서비스의 원하는 작업 수를 2로 업데이트합니다.

```
aws ecs update-service \
    --cluster MyCluster \
    --service my-http-service \
    --desired-count 2
```

출력:


```
{
    "service": {
        "serviceArn": "arn:aws:ecs:us-east-1:123456789012:service/MyCluster/my-http-service",
        "serviceName": "my-http-service",
        "clusterArn": "arn:aws:ecs:us-east-1:123456789012:cluster/MyCluster",
        "loadBalancers": [],
        "serviceRegistries": [],
        "status": "ACTIVE",
        "desiredCount": 2,
        "runningCount": 1,
        "pendingCount": 0,
        "capacityProviderStrategy": [
            {
                "capacityProvider": "FARGATE",
                "weight": 1,
                "base": 0
            }
        ],
        "platformVersion": "LATEST",
        "platformFamily": "Linux",
        "taskDefinition": "arn:aws:ecs:us-east-1:123456789012:task-definition/MyTaskDefinition",
        "deploymentConfiguration": {
            "deploymentCircuitBreaker": {
                "enable": true,
                "rollback": true
            },
            "maximumPercent": 200,
            "minimumHealthyPercent": 100,
            "alarms": {
                "alarmNames": [],
                "rollback": false,
                "enable": false
            }
        },
```

```
        "deployments": [
            {
                "id": "ecs-svc/1976744184940610707",
                "status": "PRIMARY",
                "taskkDefinition": "arn:aws:ecs:us-east-1:123456789012:task-
definition/MyTaskDefinition",
                "desiredCount": 1,
                "pendingCount": 0,
                "runningCount": 1,
                "failedTasks": 0,
                "createdAt": "2024-12-03T16:24:25.225000-05:00",
                "updatedAt": "2024-12-03T16:25:15.837000-05:00",
                "capacityProviderStrategy": [
                    {
                        "capacityProvider": "FARGATE",
                        "weight": 1,
                        "base": 0
                    }
                ],
                "platformVersion": "1.4.0",
                "platformFamily": "Linux",
                "networkConfiguration": {
                    "awsvpcConfiguration": {
                        "subnets": [
                            "subnet-0d0eab1bb38d5ca64",
                            "subnet-0db5010045995c2d5"
                        ],
                        "securityGroups": [
                            "sg-02556bf85a191f59a"
                        ],
                        "assignPublicIp": "ENABLED"
                    }
                },
                "rolloutState": "COMPLETED",
                "rolloutStateReason": "ECS deployment ecs-svc/1976744184940610707
 completed."
            }
        ],
        "roleArn": "arn:aws:iam::123456789012:role/aws-service-role/
ecs.amazonaws.com/AWSServiceRoleForECS",
        "events": [
            {
                "id": "f27350b9-4b2a-4e2e-b72e-a4b68380de45",
                "createdAt": "2024-12-30T13:24:07.345000-05:00",
```

```
                "message": "(service my-http-service) has reached a steady state."
            },
            {
                "id": "e764ec63-f53f-45e3-9af2-d99f922d2957",
                "createdAt": "2024-12-30T12:32:21.600000-05:00",
                "message": "(service my-http-service) has reached a steady state."
            },
            {
                "id": "28444756-c2fa-47f8-bd60-93a8e05f3991",
                "createdAt": "2024-12-08T19:26:10.367000-05:00",
                "message": "(service my-http-service) has reached a steady state."
            }
        ],
        "createdAt": "2024-12-03T16:24:25.225000-05:00",
        "placementConstraints": [],
        "placementStrategy": [],
        "networkConfiguration": {
            "awsvpcConfiguration": {
                "subnets": [
                    "subnet-0d0eab1bb38d5ca64",
                    "subnet-0db5010045995c2d5"
                ],
                "securityGroups": [
                    "sg-02556bf85a191f59a"
                ],
                "assignPublicIp": "ENABLED"
            }
        },
        "healthCheckGracePeriodSeconds": 0,
        "schedulingStrategy": "REPLICA",
        "deploymentController": {
            "type": "ECS"
        },
        "createdBy": "arn:aws:iam::123456789012:role/Admin",
        "enableECSManagedTags": true,
        "propagateTags": "NONE",
        "enableExecuteCommand": false,
        "availabilityZoneRebalancing": "ENABLED"
    }
}
```

자세한 내용을 알아보려면 Amazon ECS 개발자 안내서의 [콘솔을 사용하여 Amazon ECS 서비스 업데이트](#) 섹션을 참조하시기 바랍니다.

예시 3: 서비스 업데이트 시 연결하도록 Amazon EBS 볼륨 구성

다음 update-service 예시에서는 Amazon EBS 볼륨을 사용하도록 my-http-service 서비스를 업데이트합니다. AmazonECSInfrastructureRolePolicyForVolumes 관리형 정책과 연결된 Amazon ECS 인프라 역할이 구성되어 있어야 합니다. 또한 update-service 요청에서와 동일한 볼륨 이름을 사용하고 configuredAtLaunch를 true로 설정하여 작업 정의를 지정해야 합니다. 이 예시에서는 --cli-input-json 옵션과 ebs.json이라는 JSON 입력 파일을 사용합니다.

```
aws ecs update-service \
    --cli-input-json file://ebs.json
```

ebs.json의 콘텐츠:

```
{
   "cluster": "mycluster",
   "taskDefinition": "mytaskdef",
   "service": "my-http-service",
   "desiredCount": 2,
   "volumeConfigurations": [
        {
            "name": "myEbsVolume",
            "managedEBSVolume": {
                "roleArn":"arn:aws:iam::123456789012:role/ecsInfrastructureRole",
                "volumeType": "gp3",
                "sizeInGiB": 100,
                "iops": 3000,
                "throughput": 125,
                "filesystemType": "ext4"
            }
        }
   ]
}
```

출력:

```
{
    "service": {
        "serviceArn": "arn:aws:ecs:us-west-2:123456789012:service/mycluster/my-http-
service",
        "serviceName": "my-http-service",
        "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/mycluster",
        "loadBalancers": [],
```

```
        "serviceRegistries": [],
        "status": "ACTIVE",
        "desiredCount": 2,
        "runningCount": 2,
        "pendingCount": 0,
        "launchType": "FARGATE",
        "platformVersion": "LATEST",
        "platformFamily": "Linux",
        "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-definition/
mytaskdef:1",
        "deploymentConfiguration": {
            "deploymentCircuitBreaker": {
                "enable": true,
                "rollback": true
            },
            "maximumPercent": 200,
            "minimumHealthyPercent": 100,
            "alarms": {
                "alarmNames": [],
                "rollback": false,
                "enable": false
            }
        },
        "deployments": [
            {
                "id": "ecs-svc/2420458347226626275",
                "status": "PRIMARY",
                "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-
definition/mytaskdef:1",
                "desiredCount": 0,
                "pendingCount": 0,
                "runningCount": 0,
                "failedTasks": 0,
                "createdAt": "2025-02-21T15:07:20.519000-06:00",
                "updatedAt": "2025-02-21T15:07:20.519000-06:00",
                "launchType": "FARGATE",
                "platformVersion": "1.4.0",
                "platformFamily": "Linux",
                "networkConfiguration": {
                    "awsvpcConfiguration": {
                        "subnets": [
                            "subnet-12344321",
                        ],
                        "securityGroups": [
```

```
                            "sg-12344321"
                        ],
                        "assignPublicIp": "ENABLED"
                    }
                },
                "rolloutState": "IN_PROGRESS",
                "rolloutStateReason": "ECS deployment ecs-svc/2420458347226626275 in
 progress.",
                "volumeConfigurations": [
                    {
                        "name": "ebs-volume",
                        "managedEBSVolume": {
                            "volumeType": "gp3",
                            "sizeInGiB": 100,
                            "iops": 3000,
                            "throughput": 125,
                            "roleArn": "arn:aws:iam::123456789012:role/
ecsInfrastructureRole",
                            "filesystemType": "ext4"
                        }
                    }
                ]
            },
            {
                "id": "ecs-svc/5191625155316533644",
                "status": "ACTIVE",
                "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-
definition/mytaskdef:2",
                "desiredCount": 2,
                "pendingCount": 0,
                "runningCount": 2,
                "failedTasks": 0,
                "createdAt": "2025-02-21T14:54:48.862000-06:00",
                "updatedAt": "2025-02-21T14:57:22.502000-06:00",
                "launchType": "FARGATE",
                "platformVersion": "1.4.0",
                "platformFamily": "Linux",
                "networkConfiguration": {
                    "awsvpcConfiguration": {
                        "subnets": [
                            "subnet-12344321"
                        ],
                        "securityGroups": [
                            "sg-12344321"
```

```
                        ],
                        "assignPublicIp": "ENABLED"
                    }
                },
                "rolloutState": "COMPLETED",
                "rolloutStateReason": "ECS deployment ecs-svc/5191625155316533644
 completed."
            }
        ],
        "roleArn": "arn:aws:iam::123456789012:role/aws-service-role/
ecs.amazonaws.com/AWSServiceRoleForECS",
        "events": [
            {
                "id": "b5823113-c2c5-458e-9649-8c2ed38f23a5",
                "createdAt": "2025-02-21T14:57:22.508000-06:00",
                "message": "(service my-http-service) has reached a steady state."
            },
            {
                "id": "b05a48e8-da35-4074-80aa-37ceb3167357",
                "createdAt": "2025-02-21T14:57:22.507000-06:00",
                "message": "(service my-http-service) (deployment ecs-
svc/5191625155316533644) deployment completed."
            },
            {
                "id": "a10cd55d-4ba6-4cea-a655-5a5d32ada8a0",
                "createdAt": "2025-02-21T14:55:32.833000-06:00",
                "message": "(service my-http-service) has started 1 tasks: (task
 fb9c8df512684aec92f3c57dc3f22361)."
            },
        ],
        "createdAt": "2025-02-21T14:54:48.862000-06:00",
        "placementConstraints": [],
        "placementStrategy": [],
        "networkConfiguration": {
            "awsvpcConfiguration": {
                "subnets": [
                    "subnet-12344321"
                ],
                "securityGroups": [
                    "sg-12344321"
                ],
                "assignPublicIp": "ENABLED"
            }
        },
```

```
        "healthCheckGracePeriodSeconds": 0,
        "schedulingStrategy": "REPLICA",
        "deploymentController": {
            "type": "ECS"
        },
        "createdBy": "arn:aws:iam::123456789012:role/AIDACKCEVSQ6C2EXAMPLE",
        "enableECSManagedTags": true,
        "propagateTags": "NONE",
        "enableExecuteCommand": false,
        "availabilityZoneRebalancing": "ENABLED"
    }
}
```

자세한 내용은 Amazon ECS 개발자 안내서의 Amazon ECS에서 Amazon EBS 볼륨 사용을 참조하세요.

예시 4: Amazon EBS 볼륨을 더 이상 사용하지 않도록 서비스 업데이트

다음 `update-service` 예시에서는 Amazon EBS 볼륨을 더 이상 사용하지 않도록 `my-http-service` 서비스를 업데이트합니다. `configuredAtLaunch`를 `false`로 설정하여 작업 정의 개정을 지정해야 합니다.

```
aws ecs update-service \
    --cluster mycluster \
    --task-definition mytaskdef \
    --service my-http-service \
    --desired-count 2 \
    --volume-configurations "[]"
```

출력:

```
{
    "service": {
        "serviceArn": "arn:aws:ecs:us-west-2:123456789012:service/mycluster/my-http-
service",
        "serviceName": "my-http-service",
        "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/mycluster",
        "loadBalancers": [],
        "serviceRegistries": [],
        "status": "ACTIVE",
        "desiredCount": 2,
        "runningCount": 2,
        "pendingCount": 0,
```

```
        "launchType": "FARGATE",
        "platformVersion": "LATEST",
        "platformFamily": "Linux",
        "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-definition/
mytaskdef:3",
        "deploymentConfiguration": {
            "deploymentCircuitBreaker": {
                "enable": true,
                "rollback": true
            },
            "maximumPercent": 200,
            "minimumHealthyPercent": 100,
            "alarms": {
                "alarmNames": [],
                "rollback": false,
                "enable": false
            }
        },
        "deployments": [
            {
                "id": "ecs-svc/7522791612543716777",
                "status": "PRIMARY",
                "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-
definition/mytaskdef:3",
                "desiredCount": 0,
                "pendingCount": 0,
                "runningCount": 0,
                "failedTasks": 0,
                "createdAt": "2025-02-21T15:25:38.598000-06:00",
                "updatedAt": "2025-02-21T15:25:38.598000-06:00",
                    "launchType": "FARGATE",
                "platformVersion": "1.4.0",
                "platformFamily": "Linux",
                "networkConfiguration": {
                    "awsvpcConfiguration": {
                        "subnets": [
                            "subnet-12344321"
                        ],
                        "securityGroups": [
                            "sg-12344321"
                        ],
                        "assignPublicIp": "ENABLED"
                    }
                },
```

```
                "rolloutState": "IN_PROGRESS",
                "rolloutStateReason": "ECS deployment ecs-svc/7522791612543716777 in
 progress."
            },
            {
                "id": "ecs-svc/2420458347226626275",
                "status": "ACTIVE",
                "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-
definition/myoldtaskdef:1",
                "desiredCount": 2,
                "pendingCount": 0,
                "runningCount": 2,
                "failedTasks": 0,
                "createdAt": "2025-02-21T15:07:20.519000-06:00",
                "updatedAt": "2025-02-21T15:10:59.955000-06:00",
                "launchType": "FARGATE",
                "platformVersion": "1.4.0",
                "platformFamily": "Linux",
                "networkConfiguration": {
                    "awsvpcConfiguration": {
                        "subnets": [
                            "subnet-12344321"
                        ],
                        "securityGroups": [
                            "sg-12344321"
                        ],
                        "assignPublicIp": "ENABLED"
                    }
                },
                "rolloutState": "COMPLETED",
                "rolloutStateReason": "ECS deployment ecs-svc/2420458347226626275
 completed.",
                "volumeConfigurations": [
                    {
                        "name": "ebs-volume",
                        "managedEBSVolume": {
                            "volumeType": "gp3",
                            "sizeInGiB": 100,
                            "iops": 3000,
                            "throughput": 125,
                            "roleArn": "arn:aws:iam::123456789012:role/
ecsInfrastructureRole",
                            "filesystemType": "ext4"
                        }
```

```
                    }
                ]
            }
        ],
        "roleArn": "arn:aws:iam::123456789012:role/aws-service-role/
ecs.amazonaws.com/AWSServiceRoleForECS",
        "events": [
            {
                "id": "4f2c3ca1-7800-4048-ba57-bba210ada2ad",
                "createdAt": "2025-02-21T15:10:59.959000-06:00",
                "message": "(service my-http-service) has reached a steady state."
            },
            {
                "id": "4b36a593-2d40-4ed6-8be8-b9b699eb6198",
                "createdAt": "2025-02-21T15:10:59.958000-06:00",
                "message": "(service my-http-service) (deployment ecs-
svc/2420458347226626275) deployment completed."
            },
            {
                "id": "88380089-14e2-4ef0-8dbb-a33991683371",
                "createdAt": "2025-02-21T15:09:39.055000-06:00",
                "message": "(service my-http-service) has stopped 1 running tasks:
 (task fb9c8df512684aec92f3c57dc3f22361)."
            },
            {
                "id": "97d84243-d52f-4255-89bb-9311391c61f6",
                "createdAt": "2025-02-21T15:08:57.653000-06:00",
                "message": "(service my-http-service) has stopped 1 running tasks:
 (task 33eff090ad2c40539daa837e6503a9bc)."
            },
            {
                "id": "672ece6c-e2d0-4021-b5da-eefb14001687",
                "createdAt": "2025-02-21T15:08:15.631000-06:00",
                "message": "(service my-http-service) has started 1 tasks: (task
 996c02a66ff24f3190a4a8e0c841740f)."
            },
            {
                "id": "a3cf9bea-9be6-4175-ac28-4c68360986eb",
                "createdAt": "2025-02-21T15:07:36.931000-06:00",
                "message": "(service my-http-service) has started 1 tasks: (task
 d5d23c39f89e46cf9a647b9cc6572feb)."
            },
            {
                "id": "b5823113-c2c5-458e-9649-8c2ed38f23a5",
```

```
                "createdAt": "2025-02-21T14:57:22.508000-06:00",
                "message": "(service my-http-service) has reached a steady state."
            },
            {
                "id": "b05a48e8-da35-4074-80aa-37ceb3167357",
                "createdAt": "2025-02-21T14:57:22.507000-06:00",
                "message": "(service my-http-service) (deployment ecs-
svc/5191625155316533644) deployment completed."
            },
            {
                "id": "a10cd55d-4ba6-4cea-a655-5a5d32ada8a0",
                "createdAt": "2025-02-21T14:55:32.833000-06:00",
                "message": "(service my-http-service) has started 1 tasks: (task
 fb9c8df512684aec92f3c57dc3f22361)."
            },
            {
                "id": "42da91fa-e26d-42ef-88c3-bb5965c56b2f",
                "createdAt": "2025-02-21T14:55:02.703000-06:00",
                "message": "(service my-http-service) has started 1 tasks: (task
 33eff090ad2c40539daa837e6503a9bc)."
            }
        ],
        "createdAt": "2025-02-21T14:54:48.862000-06:00",
        "placementConstraints": [],
        "placementStrategy": [],
        "networkConfiguration": {
            "awsvpcConfiguration": {
                "subnets": [
                    "subnet-12344321"
                ],
                "securityGroups": [
                    "sg-12344321"
                ],
                "assignPublicIp": "ENABLED"
            }
        },
        "healthCheckGracePeriodSeconds": 0,
        "schedulingStrategy": "REPLICA",
        "deploymentController": {
            "type": "ECS"
        },
        "createdBy": "arn:aws:iam::123456789012:role/AIDACKCEVSQ6C2EXAMPLE",
        "enableECSManagedTags": true,
        "propagateTags": "NONE",
```

```
        "enableExecuteCommand": false,
        "availabilityZoneRebalancing": "ENABLED"
    }
}
```

자세한 내용은 Amazon ECS 개발자 안내서의 Amazon ECS에서 Amazon EBS 볼륨 사용을 참조하세요.

예시 5: 서비스에 대한 가용 영역 리밸런싱 켜기

다음 `update-service` 예제에서는 `my-http-service` 서비스에 대해 가용 영역 리밸런싱을 켭니다.

```
aws ecs update-service \
    --cluster MyCluster \
    --service my-http-service \
    --availability-zone-rebalancing ENABLED
```

출력:

```
{
    "service": {
        "serviceArn": "arn:aws:ecs:us-east-1:123456789012:service/MyCluster/my-http-
service",
        "serviceName": "my-http-service",
        "clusterArn": "arn:aws:ecs:us-east-1:123456789012:cluster/MyCluster",
        "loadBalancers": [],
        "serviceRegistries": [],
        "status": "ACTIVE",
        "desiredCount": 2,
        "runningCount": 1,
        "pendingCount": 0,
        "capacityProviderStrategy": [
            {
                "capacityProvider": "FARGATE",
                "weight": 1,
                "base": 0
            }
        ],
        "platformVersion": "LATEST",
        "platformFamily": "Linux",
        "taskDefinition": "arn:aws:ecs:us-east-1:123456789012:task-definition/
MyTaskDefinition",
```

```
        "deploymentConfiguration": {
            "deploymentCircuitBreaker": {
                "enable": true,
                "rollback": true
            },
            "maximumPercent": 200,
            "minimumHealthyPercent": 100,
            "alarms": {
                "alarmNames": [],
                "rollback": false,
                "enable": false
            }
        },
        "deployments": [
            {
                "id": "ecs-svc/1976744184940610707",
                "status": "PRIMARY",
                "taskkDefinition": "arn:aws:ecs:us-east-1:123456789012:task-
definition/MyTaskDefinition",
                "desiredCount": 1,
                "pendingCount": 0,
                "runningCount": 1,
                "failedTasks": 0,
                "createdAt": "2024-12-03T16:24:25.225000-05:00",
                "updatedAt": "2024-12-03T16:25:15.837000-05:00",
                "capacityProviderStrategy": [
                    {
                        "capacityProvider": "FARGATE",
                        "weight": 1,
                        "base": 0
                    }
                ],
                "platformVersion": "1.4.0",
                "platformFamily": "Linux",
                "networkConfiguration": {
                    "awsvpcConfiguration": {
                        "subnets": [
                            "subnet-0d0eab1bb38d5ca64",
                            "subnet-0db5010045995c2d5"
                        ],
                        "securityGroups": [
                            "sg-02556bf85a191f59a"
                        ],
                        "assignPublicIp": "ENABLED"
```

```
                    }
                },
                "rolloutState": "COMPLETED",
                "rolloutStateReason": "ECS deployment ecs-svc/1976744184940610707
 completed."
            }
        ],
        "roleArn": "arn:aws:iam::123456789012:role/aws-service-role/
ecs.amazonaws.com/AWSServiceRoleForECS",
        "events": [],
        "createdAt": "2024-12-03T16:24:25.225000-05:00",
        "placementConstraints": [],
        "placementStrategy": [],
        "networkConfiguration": {
            "awsvpcConfiguration": {
                "subnets": [
                    "subnet-0d0eab1bb38d5ca64",
                    "subnet-0db5010045995c2d5"
                ],
                "securityGroups": [
                    "sg-02556bf85a191f59a"
                ],
                "assignPublicIp": "ENABLED"
            }
        },
        "healthCheckGracePeriodSeconds": 0,
        "schedulingStrategy": "REPLICA",
        "deploymentController": {
            "type": "ECS"
        },
        "createdBy": "arn:aws:iam::123456789012:role/Admin",
        "enableECSManagedTags": true,
        "propagateTags": "NONE",
        "enableExecuteCommand": false,
        "availabilityZoneRebalancing": "ENABLED"
    }
}
```

자세한 내용을 알아보려면 Amazon ECS 개발자 안내서의 콘솔을 사용하여 Amazon ECS 서비스 업데이트 섹션을 참조하시기 바랍니다.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateService를 참조하세요.

`update-task-protection`

다음 코드 예시는 `update-task-protection`의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: ECS 작업에 대한 작업 보호 활성화

다음 `update-task-protection`에서는 배포 또는 서비스 AutoScaling에서 스케일 인 중에 ECS 작업이 종료되지 않도록 보호합니다. 작업 보호를 위한 사용자 지정 만료 기간을 1분에서 최대 2,880분(48시간)까지 지정할 수 있습니다. 만료 기간을 지정하지 않으면 작업 보호 기본 시간 활성화는 2시간입니다.

```
aws ecs update-task-protection \
    --cluster ECS-project-update-cluster \
    --tasks c43ed3b1331041f289316f958adb6a24 \
    --protection-enabled \
    --expires-in-minutes 300
```

출력:

```
{
"protectedTasks": [
    {
        "taskArn": "arn:aws:ecs:us-west-2:123456789012:task/
c43ed3b1331041f289316f958adb6a24",
        "protectionEnabled": true,
        "expirationDate": "2024-09-14T19:53:36.687000-05:00"
    }
],
"failures": []
}
```

예제 2: ECS 작업에 대한 작업 보호 비활성화

다음 `update-task-protection`에서는 배포 또는 서비스 AutoScaling에서 스케일 인으로부터 보호되는 작업을 비활성화합니다.

```
aws ecs update-task-protection \
    --cluster ECS-project-update-cluster \
    --tasks c43ed3b1331041f289316f958adb6a24 \
```

```
    --no-protection-enabled
```

출력:

```
{
    "protectedTasks": [
        {
            "taskArn": "arn:aws:ecs:us-west-2:123456789012:task/
c43ed3b1331041f289316f958adb6a24",
            "protectionEnabled": false
        }
    ],
    "failures": []
}
```

작업 보호에 대한 자세한 내용은 Amazon ECS 개발자 안내서의 스케일 인 이벤트로 인해 Amazon ECS 작업이 종료되지 않도록 보호를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateTaskProtection을 참조하세요.

update-task-set

다음 코드 예시는 update-task-set의 사용 방법을 보여 줍니다.

AWS CLI

작업 세트를 업데이트하려면

다음 update-task-set 예시에서는 작업 세트를 업데이트하여 규모를 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
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateTaskSet 섹션을 참조하세요.

AWS CLI를 사용한 Amazon EFS 예제

다음 코드 예제에서는 Amazon EFS에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여 줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`create-file-system`

다음 코드 예시는 `create-file-system`의 사용 방법을 보여 줍니다.

AWS CLI

암호화된 파일 시스템을 생성하는 방법

다음 `create-file-system` 예제에서는 기본 CMK를 사용하여 암호화된 파일 시스템을 생성합니다. 또한 태그 `Name=my-file-system`도 추가합니다.

```
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 File System 사용 설명서에서 Creating Amazon EFS file systems을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateFileSystem을 참조하세요.

`create-mount-target`

다음 코드 예시는 `create-mount-target`의 사용 방법을 보여 줍니다.

AWS CLI

탑재 대상을 생성하는 방법

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 Creating mount targets을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateMountTarget을 참조하세요.

delete-file-system

다음 코드 예시는 delete-file-system의 사용 방법을 보여 줍니다.

AWS CLI

파일 시스템을 삭제하는 방법

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 File System 사용 설명서의 Deleting an Amazon EFS file system를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteFileSystem을 참조하세요.

delete-mount-target

다음 코드 예시는 delete-mount-target의 사용 방법을 보여 줍니다.

AWS CLI

탑재 대상을 삭제하는 방법

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 Creating mount targets을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteMountTarget을 참조하세요.

describe-file-systems

다음 코드 예시는 describe-file-systems의 사용 방법을 보여 줍니다.

AWS CLI

파일 시스템을 설명하는 방법

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 File System 사용 설명서의 Managing Amazon EFS file systems를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeFileSystems를 참조하세요.

describe-mount-targets

다음 코드 예시는 describe-mount-targets의 사용 방법을 보여 줍니다.

AWS CLI

탑재 대상을 설명하는 방법

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 Creating mount targets을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeMountTargets를 참조하세요.

describe-tags

다음 코드 예시는 describe-tags의 사용 방법을 보여 줍니다.

AWS CLI

파일 시스템의 태그를 설명하는 방법

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 사용 설명서의 Managing file system tags를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeTags 섹션을 참조하세요.

list-tags-for-resource

다음 코드 예시는 list-tags-for-resource의 사용 방법을 보여 줍니다.

AWS CLI

리소스에 대한 태그를 검색하는 방법

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 사용 설명서의 Managing file system tags를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource를 참조하세요.

tag-resource

다음 코드 예시에서는 tag-resource의 사용 방법을 보여줍니다.

AWS CLI

리소스에 태그를 지정하려면

다음 `tag-resource` 예제에서는 지정된 파일 시스템에 `Department=Business Intelligence` 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 사용 설명서의 [Managing file system tags](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [TagResource](#)를 참조하세요.

`untag-resource`

다음 코드 예시에서는 `untag-resource`의 사용 방법을 보여줍니다.

AWS CLI

리소스에서 태그를 제거하는 방법

다음 `untag-resource` 예제에서는 지정된 파일 시스템에서 `Department` 태그 키가 있는 태그를 제거합니다.

```
aws efs untag-resource \
    --resource-id fs-c7a0456e \
    --tag-keys Department
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Elastic File System 사용 설명서의 [Managing file system tags](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [UntagResource](#)를 참조하세요.

AWS CLI를 사용한 Amazon EKS 예시

다음 코드 예시에서는 Amazon EKS에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

associate-access-policy

다음 코드 예시는 associate-access-policy의 사용 방법을 보여 줍니다.

AWS CLI

클러스터의 액세스 항목에 액세스 정책과 그 범위를 연결하는 방법

다음 associate-access-policy는 액세스 정책과 그 범위를 지정된 클러스터의 액세스 항목에 연결합니다.

```
aws eks associate-access-policy \
    --cluster-name eks-customer \
    --principal-arn arn:aws:iam::111122223333:role/Admin \
    --policy-arn arn:aws:eks::aws:cluster-access-policy/AmazonEKSEditPolicy \
    --access-scope type=namespace,namespaces=default
```

출력:

```
{
    "clusterName": "eks-customer",
    "principalArn": "arn:aws:iam::111122223333:role/Admin",
    "associatedAccessPolicy": {
        "policyArn": "arn:aws:eks::aws:cluster-access-policy/AmazonEKSEditPolicy",
        "accessScope": {
            "type": "namespace",
            "namespaces": [
                "default"
            ]
```

```
        },
        "associatedAt": "2025-05-24T15:59:51.981000-05:00",
        "modifiedAt": "2025-05-24T15:59:51.981000-05:00"
    }
}
```

자세한 내용은 Amazon EKS 사용 설명서의 액세스 정책을 액세스 항목과 연결을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AssociateAccessPolicy를 참조하세요.

associate-encryption-config

다음 코드 예시는 associate-encryption-config의 사용 방법을 보여 줍니다.

AWS CLI

암호화 구성을 기존 클러스터에 연결하려면

다음 associate-encryption-config 예시에서는 아직 암호화가 활성화되지 않은 기존 EKS 클러스터에서 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 활성화 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AssociateEncryptionConfig 섹션을 참조하세요.

associate-identity-provider-config

다음 코드 예시는 associate-identity-provider-config의 사용 방법을 보여 줍니다.

AWS CLI

ID 제공업체를 Amazon EKS 클러스터에 연결하려면

다음 associate-identity-provider-config 예시에서는 ID 제공업체를 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 ID 제공업체에서 클러스터에 대한 사용자 인증 - OIDC ID 제공업체 연결 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AssociateIdentityProviderConfig 섹션을 참조하세요.

`create-access-entry`

다음 코드 예시는 `create-access-entry`의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: EKS 클러스터에 대한 액세스 항목을 생성하는 방법

다음 `create-access-entry` 예제에서는 IAM 위탁자가 EKS 클러스터에 액세스할 수 있도록 허용하는 액세스 항목을 생성합니다.

```
aws eks create-access-entry \
    --cluster-name eks-customer \
    --principal-arn arn:aws:iam::111122223333:user/eks-user
```

출력:

```
{
    "accessEntry": {
        "clusterName": "eks-customer",
        "principalArn": "arn:aws:iam::111122223333:user/eks-user",
        "kubernetesGroups": [],
        "accessEntryArn": "arn:aws:eks:us-west-2:111122223333:access-entry/eks-
customer/user/111122223333/eks-user/a1b2c3d4-5678-90ab-cdef-a6506e3d36p0",
        "createdAt": "2025-04-14T22:45:48.097000-05:00",
        "modifiedAt": "2025-04-14T22:45:48.097000-05:00",
        "tags": {},
        "username": "arn:aws:iam::111122223333:user/eks-user",
        "type": "STANDARD"
    }
```

```
}
```

자세한 내용은 Amazon EKS 사용 설명서의 액세스 항목 생성을 참조하세요.

예제 2: 액세스 항목 유형을 지정하여 EKS 클러스터에 대한 액세스 항목을 생성하는 방법

다음 `create-access-entry` 예제에서는 EKS 클러스터에 `EC2_LINUX` 유형의 액세스 항목을 생성합니다. 기본적으로 `STANDARD` 유형의 액세스 항목이 생성됩니다. 기본값 외에 다른 액세스 항목 유형을 지정하는 경우 CLI에서 IAM 역할 ARN을 전달해야 합니다.

```
aws eks create-access-entry \
    --cluster-name eks-customer \
    --principal-arn arn:aws:iam::111122223333:role/admin-test-ip \
    --type EC2_LINUX
```

출력:

```
{
    "accessEntry": {
        "clusterName": "eks-customer",
        "principalArn": "arn:aws:iam::111122223333:role/admin-test-ip",
        "kubernetesGroups": [
            "system:nodes"
        ],
        "accessEntryArn": "arn:aws:eks:us-west-2:111122223333:access-entry/eks-
customer/role/111122223333/admin-test-ip/accb5418-f493-f390-3e6e-c3f19f725fcp",
        "createdAt": "2025-05-06T19:42:45.453000-05:00",
        "modifiedAt": "2025-05-06T19:42:45.453000-05:00",
        "tags": {},
        "username": "system:node:{{EC2PrivateDNSName}}",
        "type": "EC2_LINUX"
    }
}
```

자세한 내용은 Amazon EKS 사용 설명서의 액세스 항목 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateAccessEntry를 참조하세요.

create-addon

다음 코드 예시는 `create-addon`의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 각 EKS 클러스터 버전에 대한 기본 호환성 버전으로 Amazon EKS 추가 기능을 만들려면

다음 `create-addon` 예시 명령은 각 EKS 클러스터 버전에 대한 기본 호환 버전으로 Amazon EKS 추가 기능을 만듭니다.

```
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 EKS 사용 설명서의 [Amazon EKS 추가 기능 관리 - 추가 기능 생성](https://docs.aws.amazon.com/eks/latest/userguide/managing-add-ons.html#creating-an-add-on) 섹션을 참조하세요.

예시 2: 특정 추가 기능 버전으로 Amazon EKS 추가 기능을 만들려면

다음 `create-addon` 예시 명령은 특정 추가 기능 버전으로 Amazon EKS 추가 기능을 만듭니다.

```
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 EKS 사용 설명서의 Amazon EKS 추가 기능 관리 - 추가 기능 생성 섹션을 참조하세요.

예시 3: 사용자 지정 구성 값으로 Amazon EKS 추가 기능을 만들고 충돌 세부 정보를 해결하려면

다음 `create-addon` 예시 명령은 사용자 지정 구성 값으로 Amazon EKS 추가 기능을 만들고 세부 정보 충돌을 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 EKS 사용 설명서의 Amazon EKS 추가 기능 관리 - 추가 기능 생성 섹션을 참조하세요.

예시 4: 사용자 지정 JSON 구성 값 파일을 사용하여 Amazon EKS 추가 기능을 만들려면

다음 `create-addon` 예시 명령은 사용자 지정 구성 값으로 Amazon EKS 추가 기능을 만들고 세부 정보 충돌을 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.json`의 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 EKS 사용 설명서의 Amazon EKS 추가 기능 관리 - 추가 기능 생성 섹션을 참조하세요.

예시 5: 사용자 지정 YAML 구성 값 파일을 사용하여 Amazon EKS 추가 기능을 만들려면

다음 `create-addon` 예시 명령은 사용자 지정 구성 값으로 Amazon EKS 추가 기능을 만들고 세부 정보 충돌을 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.yaml`의 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 EKS 사용 설명서의 Amazon EKS 추가 기능 관리 - 추가 기능 생성 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [CreateAddon](#) 섹션을 참조하세요.

create-cluster

다음 코드 예시는 create-cluster의 사용 방법을 보여 줍니다.

AWS CLI

새로운 클러스터를 생성하는 방법

이 예시 명령은 기본 리전에 이름이 prod인 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 `example`인 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [CreateCluster](#)를 참조하세요.

`create-fargate-profile`

다음 코드 예시는 `create-fargate-profile`의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 네임스페이스가 있는 선택기에 대한 EKS Fargate 프로파일을 만들려면

다음 `create-fargate-profile` 예시에서는 네임스페이스가 있는 선택기에 대한 EKS Fargate 프로파일을 만듭니다.

```
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 프로파일 - Fargate 프로파일 생성 섹션을 참조하세요.

예시 2: 네임스페이스와 레이블이 있는 선택기에 대한 EKS Fargate 프로파일을 만들려면

다음 `create-fargate-profile` 예시에서는 네임스페이스와 레이블이 있는 선택기에 대한 EKS Fargate 프로파일을 만듭니다.

```
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 프로파일 - Fargate 프로파일 생성 섹션을 참조하세요.

예시 3: 포드를 시작할 서브넷 ID와 함께 네임스페이스와 레이블이 있는 선택기의 EKS Fargate 프로파일을 만들려면

다음 `create-fargate-profile` 예시에서는 네임스페이스와 레이블, 포드를 실행할 서브넷의 ID가 있는 선택기에 대한 EKS Fargate 프로파일을 만듭니다.

```
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 프로파일 - Fargate 프로파일 생성 섹션을 참조하세요.

예시 4: 포드를 시작할 서브넷의 ID와 함께 여러 네임스페이스 및 레이블이 있는 선택기의 EKS Fargate 프로파일을 만들려면

다음 `create-fargate-profile` 예시에서는 포드를 실행할 서브넷의 ID와 함께 여러 네임스페이스 및 레이블이 있는 선택기에 대한 EKS Fargate 프로파일을 만듭니다.

```
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 프로파일 - Fargate 프로파일 생성](#) 섹션을 참조하세요.

예시 5: 포드를 시작할 서브넷의 ID와 함께 네임스페이스 및 레이블의 와일드카드 선택기를 사용하여 EKS Fargate 프로파일을 만들려면

다음 `create-fargate-profile` 예시에서는 포드를 실행할 서브넷의 ID와 함께 여러 네임스페이스 및 레이블이 있는 선택기에 대한 EKS Fargate 프로파일을 만듭니다.

```
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 프로파일 - Fargate 프로파일 생성 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateFargateProfile 섹션을 참조하세요.

`create-nodegroup`

다음 코드 예시는 `create-nodegroup`의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: Amazon EKS 클러스터에 대한 관리형 노드 그룹을 만들려면

다음 `create-nodegroup` 예시에서는 Amazon EKS 클러스터의 관리형 노드 그룹을 만듭니다.

```
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 생성 섹션을 참조하세요.

예시 2: 사용자 지정 instance-types 및 disk-size를 사용하여 Amazon EKS 클러스터에 대한 관리형 노드 그룹을 만들려면

다음 `create-nodegroup` 예시에서는 사용자 지정 instance-types 및 disk-size를 사용하여 Amazon EKS 클러스터에 대한 관리형 노드 그룹을 만듭니다.

```
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 생성 섹션을 참조하세요.

예시 3: 사용자 지정 instance-types, disk-size, ami-type, capacity-type, update-config, labels, taints 및 tags를 사용하여 Amazon EKS 클러스터에 대한 관리형 노드 그룹을 만들려면

다음 `create-nodegroup` 예시에서는 사용자 지정 instance-types, disk-size, ami-type, capacity-type, update-config, labels, taints 및 tags를 사용하여 Amazon EKS 클러스터에 대한 관리형 노드 그룹을 만듭니다.

```
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 생성 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateNodegroup 섹션을 참조하세요.

create-pod-identity-association

다음 코드 예시는 create-pod-identity-association의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: EKS 클러스터에서 EKS Pod Identity 연결을 생성하는 방법

다음 create-pod-identity-association 예제에서는 EKS 클러스터의 서비스 계정과 IAM 역할 간에 EKS Pod Identity 연결을 생성합니다.

```
aws eks create-pod-identity-association \
    --cluster-name eks-customer \
    --namespace default \
    --service-account default \
    --role-arn arn:aws:iam::111122223333:role/my-role
```

출력:

```
{
    "association": {
        "clusterName": "eks-customer",
        "namespace": "default",
        "serviceAccount": "default",
        "roleArn": "arn:aws:iam::111122223333:role/my-role",
        "associationArn": "arn:aws:eks:us-
west-2:111122223333:podidentityassociation/eks-customer/a-8mvwvh57cu74mgcst",
        "associationId": "a-8mvwvh57cu74mgcst",
        "tags": {},
        "createdAt": "2025-05-24T19:40:13.961000-05:00",
        "modifiedAt": "2025-05-24T19:40:13.961000-05:00"
    }
```

```
}
```

자세한 내용은 Amazon EKS 사용 설명서의 EKS Pod Identity가 포드에 AWS 서비스에 대한 액세스 권한을 부여하는 방법 알아보기를 참조하세요.

예제 2: 태그를 사용하여 EKS 클러스터에서 EKS Pod Identity 연결을 생성하는 방법

다음 `create-pod-identity-association`은 태그를 사용하여 EKS 클러스터의 IAM 역할과 서비스 계정 간에 EKS Pod Identity 연결을 생성합니다.

```
aws eks create-pod-identity-association \
    --cluster-name eks-customer \
    --namespace default \
    --service-account default \
    --role-arn arn:aws:iam::111122223333:role/my-role \
    --tags Key1=value1,Key2=value2
```

출력:

```
{
    "association": {
        "clusterName": "eks-customer",
        "namespace": "default",
        "serviceAccount": "default",
        "roleArn": "arn:aws:iam::111122223333:role/my-role",
        "associationArn": "arn:aws:eks:us-
west-2:111122223333:podidentityassociation/eks-customer/a-9njjin9gfghecgoda",
        "associationId": "a-9njjin9gfghecgoda",
        "tags": {
            "Key2": "value2",
            "Key1": "value1"
        },
        "createdAt": "2025-05-24T19:52:14.135000-05:00",
        "modifiedAt": "2025-05-24T19:52:14.135000-05:00"
    }
}
```

자세한 내용은 Amazon EKS 사용 설명서의 EKS Pod Identity가 포드에 AWS 서비스에 대한 액세스 권한을 부여하는 방법 알아보기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreatePodIdentityAssociation을 참조하세요.

delete-access-entry

다음 코드 예시는 delete-access-entry의 사용 방법을 보여 줍니다.

AWS CLI

클러스터와 연결된 액세스 항목을 삭제하는 방법

다음 delete-access-entry는 EKS 클러스터 eks-customer와 연결된 액세스 항목을 삭제합니다.

```
aws eks delete-access-entry \
    --cluster-name eks-customer \
    --principal-arn arn:aws:iam::111122223333:role/Admin
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon EKS 사용 설명서의 액세스 항목 삭제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteAccessEntry를 참조하세요.

delete-addon

다음 코드 예시는 delete-addon의 사용 방법을 보여 줍니다.

AWS CLI

예시 1. Amazon EKS 추가 기능을 삭제하지만 EKS 클러스터에서 추가 기능 소프트웨어를 유지하려면

다음 delete-addon 예시 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 EKS의 Amazon EKS 추가 기능 관리 - 추가 기능 삭제 섹션을 참조하세요.

예시 2. Amazon EKS 추가 기능을 삭제하고 EKS 클러스터에서 추가 기능 소프트웨어도 삭제하려면

다음 `delete-addon` 예시 명령은 Amazon EKS 추가 기능을 삭제하고 EKS 클러스터에서 추가 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 EKS의 Amazon EKS 추가 기능 관리 - 추가 기능 삭제 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteAddon 섹션을 참조하세요.

delete-cluster

다음 코드 예시는 delete-cluster의 사용 방법을 보여 줍니다.

AWS CLI

Amazon EKS 클러스터 컨트롤 플레인을 삭제하려면

다음 delete-cluster 예시에서는 Amazon EKS 클러스터 컨트롤 플레인을 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 EKS 사용 설명서의 Amazon EKS 클러스터 삭제 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteCluster를 참조하세요.

delete-fargate-profile

다음 코드 예시는 delete-fargate-profile의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 네임스페이스가 있는 선택기에 대한 EKS Fargate 프로파일을 만들려면

다음 delete-fargate-profile 예시에서는 네임스페이스가 있는 선택기에 대한 EKS Fargate 프로파일을 만듭니다.

```
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 삭제 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteFargateProfile 섹션을 참조하세요.

delete-nodegroup

다음 코드 예시는 `delete-nodegroup`의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: Amazon EKS 클러스터의 관리형 노드 그룹을 삭제하려면

다음 `delete-nodegroup` 예시에서는 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"
        }
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteNodegroup 섹션을 참조하세요.

delete-pod-identity-association

다음 코드 예시는 delete-pod-identity-association의 사용 방법을 보여 줍니다.

AWS CLI

EKS Pod Identity 연결을 삭제하는 방법

다음 delete-pod-identity-association 예제에서는 EKS 클러스터 a-9njjin9gfghecgocd에서 연결 ID가 eks-customer인 EKS Pod Identity 연결을 삭제합니다.

```
aws eks delete-pod-identity-association \
    --cluster-name eks-customer \
    --association-id a-9njjin9gfghecgocd
```

출력:

```
{
    "association": {
        "clusterName": "eks-customer",
        "namespace": "default",
        "serviceAccount": "default",
        "roleArn": "arn:aws:iam::111122223333:role/s3-role",
        "associationArn": "arn:aws:eks:us-
west-2:111122223333:podidentityassociation/eks-customer/a-9njjin9gfghecgocd",
        "associationId": "a-9njjin9gfghecgocd",
        "tags": {
            "Key2": "value2",
            "Key1": "value1"
        },
        "createdAt": "2025-05-24T19:52:14.135000-05:00",
```

```
        "modifiedAt": "2025-05-25T21:10:56.923000-05:00"
    }
}
```

자세한 내용은 Amazon EKS 사용 설명서의 EKS Pod Identity가 포드에 AWS 서비스에 대한 액세스 권한을 부여하는 방법 알아보기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeletePodIdentityAssociation을 참조하세요.

deregister-cluster

다음 코드 예시는 deregister-cluster의 사용 방법을 보여 줍니다.

AWS CLI

연결된 클러스터를 등록 취소하여 Amazon EKS 컨트롤 플레인에서 제거하려면

다음 deregister-cluster 예시에서는 연결된 클러스터를 등록 취소하여 Amazon EKS 컨트롤 플레인에서 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 취소 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeregisterCluster 섹션을 참조하세요.

describe-access-entry

다음 코드 예시는 describe-access-entry의 사용 방법을 보여 줍니다.

AWS CLI

EKS 클러스터의 액세스 항목을 설명하는 방법

다음 describe-access-entry 예제에서는 EKS 클러스터의 액세스 항목을 설명합니다.

```
aws eks describe-access-entry \
    --cluster-name eks-customer \
    --principal-arn arn:aws:iam::111122223333:user/eks-admin-user
```

출력:

```
{
    "accessEntry": {
        "clusterName": "eks-customer",
        "principalArn": "arn:aws:iam::111122223333:user/eks-admin-user",
        "kubernetesGroups": [],
        "accessEntryArn": "arn:aws:eks:us-west-2:111122223333:access-entry/eks-
customer/user/111122223333/eks-admin-user/0acb1bc6-cb0a-ede6-11ae-a6506e3d36p0",
        "createdAt": "2025-04-14T22:45:48.097000-05:00",
        "modifiedAt": "2025-04-14T22:45:48.097000-05:00",
        "tags": {},
        "username": "arn:aws:iam::111122223333:user/eks-admin-user",
        "type": "STANDARD"
    }
}
```

자세한 내용은 Amazon EKS 사용 설명서의 IAM 사용자에게 EKS 액세스 항목을 사용하여 Kubernetes에 대한 액세스 권한 부여를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeAccessEntry를 참조하세요.

describe-addon-configuration

다음 코드 예시는 describe-addon-configuration의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: Amazon vpc-cni 추가 기능을 만들거나 업데이트할 때 사용할 수 있는 구성 옵션

다음 `describe-addon-configuration` 예시에서는 vpc-cni 추가 기능에 대해 추가 기능이 생성되거나 업데이트될 때 사용 가능한 모든 구성 스키마를 해당 버전과 함께 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\"}"
}
```

예시 2: Amazon coredns 추가 기능을 만들거나 업데이트할 때 사용할 수 있는 구성 옵션

다음 describe-addon-configuration 예시에서는 coredns 추가 기능에 대한 추가 기능이 생성되거나 업데이트될 때 사용 가능한 모든 구성 스키마를 해당 버전과 함께 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 EKS의 Amazon EKS 클러스터에 대한 kubeconfig 파일 생성 또는 업데이트 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeAddonConfiguration 섹션을 참조하세요.

describe-addon-versions

다음 코드 예시는 describe-addon-versions의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: EKS 클러스터에 사용 가능한 모든 추가 기능을 나열하려면

다음 describe-addon-versions 예시에서는 사용 가능한 모든 AWS 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 EKS 사용 설명서의 [Amazon EKS 추가 기능 관리 - 추가 기능 생성](#) 섹션을 참조하세요.

예시 2: EKS용으로 지원되는 지정된 Kubernetes 버전에 사용 가능한 모든 추가 기능을 나열하려면

다음 `describe-addon-versions` 예시에서는 EKS용으로 지원되는 지정된 Kubernetes 버전에 사용 가능한 모든 추가 기능을 나열합니다.

```
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 EKS 사용 설명서의 Amazon EKS 추가 기능 관리 - 추가 기능 생성 섹션을 참조하세요.

예시 3: EKS용으로 지원되는 지정된 Kubernetes 버전에 사용 가능한 모든 vpc-cni 추가 기능 버전을 나열하려면

다음 `describe-addon-versions` 예시에서는 EKS용으로 지원되는 지정된 Kubernetes 버전에 대해 사용 가능한 모든 vpc-cni 추가 기능 버전을 나열합니다.

```
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 EKS 사용 설명서의 Amazon EKS 추가 기능 관리 - 추가 기능 생성 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeAddonVersions 섹션을 참조하세요.

describe-addon

다음 코드 예시는 describe-addon의 사용 방법을 보여 줍니다.

AWS CLI

Amazon EKS 클러스터에서 EKS 추가 기능을 능동적으로 실행하는 방법 설명

다음 describe-addon 예시에서는 Amazon EKS 클러스터에서 EKS 추가 기능을 능동적으로 실행합니다.

```
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'"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeAddon 섹션을 참조하세요.

describe-cluster

다음 코드 예시는 `describe-cluster`의 사용 방법을 보여 줍니다.

AWS CLI

Amazon EKS 클러스터에서 EKS 추가 기능을 능동적으로 실행하는 방법 설명

다음 `describe-cluster` 예시에서는 Amazon EKS 클러스터에서 EKS 추가 기능을 능동적으로 실행합니다.

```
aws eks describe-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"
        }
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeCluster를 참조하세요.

describe-fargate-profile

다음 코드 예시는 describe-fargate-profile의 사용 방법을 보여 줍니다.

AWS CLI

Fargate 프로파일 설명

다음 describe-fargate-profile 예시에서는 Fargate 프로파일을 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"
        }
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeFargateProfile 섹션을 참조하세요.

describe-identity-provider-config

다음 코드 예시는 describe-identity-provider-config의 사용 방법을 보여 줍니다.

AWS CLI

Amazon EKS 클러스터와 연결된 ID 제공업체 구성 설명

다음 `describe-identity-provider-config` 예시에서는 Amazon EKS 클러스터에 연결된 ID 제공업체 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 ID 제공업체에서 클러스터에 대한 사용자 인증](#) 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeIdentityProviderConfig](#) 섹션을 참조하세요.

`describe-insight`

다음 코드 예시는 `describe-insight`의 사용 방법을 보여 줍니다.

AWS CLI

ID를 사용하여 EKS 클러스터에 대한 인사이트 세부 정보를 가져오는 방법

다음 `describe-insight` 예제에서는 클러스터 이름과 인사이트 ID를 사용하여 지정된 인사이트에 대한 세부 정보를 반환합니다.

```
aws eks describe-insight \
    --cluster-name eks-customer \
    --id 38ea7a64-a14f-4e0e-95c7-8dbcab3c3623
```

출력:

```
{
    "insight": {
        "id": "38ea7a64-a14f-4e0e-95c7-8dbcab3c3623",
        "name": "Kubelet version skew",
        "category": "UPGRADE_READINESS",
        "kubernetesVersion": "1.33",
        "lastRefreshTime": "2025-05-24T11:22:50-05:00",
        "lastTransitionTime": "2025-05-24T11:22:50-05:00",
        "description": "Checks for kubelet versions of worker nodes in the cluster
 to see if upgrade would cause noncompliance with supported Kubernetes kubelet
 version skew policy.",
        "insightStatus": {
            "status": "PASSING",
            "reason": "Node kubelet versions match the cluster control plane
 version."
        },
        "recommendation": "Upgrade your worker nodes to match the Kubernetes version
 of your cluster control plane.",
        "additionalInfo": {
            "Kubelet version skew policy": "https://kubernetes.io/releases/version-
skew-policy/#kubelet",
            "Updating a managed node group": "https://docs.aws.amazon.com/eks/
latest/userguide/update-managed-node-group.html"
        },
        "resources": [],
        "categorySpecificSummary": {
```

```
            "deprecationDetails": []
        }
    }
}
```

자세한 내용은 Amazon EKS 사용 설명서의 클러스터 인사이트 보기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeInsight를 참조하세요.

describe-nodegroup

다음 코드 예시는 describe-nodegroup의 사용 방법을 보여 줍니다.

AWS CLI

Amazon EKS 클러스터에 대한 관리형 노드 그룹 설명

다음 describe-nodegroup 예시에서는 Amazon EKS 클러스터의 관리형 노드 그룹을 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": {}
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeNodegroup](https://docs.aws.amazon.com/goto/aws-cli/eks-2017-11-01/DescribeNodegroup) 섹션을 참조하세요.

describe-pod-identity-association

다음 코드 예시는 `describe-pod-identity-association`의 사용 방법을 보여 줍니다.

AWS CLI

Pod Identity 연결에 대한 세부 정보를 제공하는 방법

다음 `describe-pod-identity-association` 예제에서는 EKS 클러스터의 Pod Identity 연결을 설명합니다.

```
aws eks describe-pod-identity-association \
    --cluster-name eks-customer \
    --association-id a-9njjin9gfghecgocd
```

출력:

```
{
    "association": {
        "clusterName": "eks-customer",
        "namespace": "default",
        "serviceAccount": "default",
        "roleArn": "arn:aws:iam::111122223333:role/my-role",
        "associationArn": "arn:aws:eks:us-
west-2:111122223333:podidentityassociation/eks-customer/a-9njjin9gfghecgocd",
        "associationId": "a-9njjin9gfghecgocd",
        "tags": {
            "Key2": "value2",
            "Key1": "value1"
        },
        "createdAt": "2025-05-24T19:52:14.135000-05:00",
        "modifiedAt": "2025-05-24T19:52:14.135000-05:00"
    }
}
```

자세한 내용은 Amazon EKS 사용 설명서의 EKS Pod Identity가 포드에 AWS 서비스에 대한 액세스 권한을 부여하는 방법 알아보기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribePodIdentityAssociation을 참조하세요.

describe-update

다음 코드 예시는 describe-update의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 클러스터 업데이트를 설명하려면

다음 describe-update 예시에서는 이름이 지정된 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 EKS 사용 설명서의 [Amazon EKS 클러스터 Kubernetes 버전 업데이트](#) 섹션을 참조하세요.

예시 2: 클러스터 업데이트를 설명하려면

다음 `describe-update` 예시에서는 이름이 지정된 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 EKS 사용 설명서의 Amazon EKS 클러스터 Kubernetes 버전 업데이트 섹션을 참조하세요.

예시 3: 클러스터 업데이트를 설명하려면

다음 `describe-update` 예시에서는 이름이 지정된 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 EKS 사용 설명서의 Amazon EKS 클러스터 Kubernetes 버전 업데이트 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeUpdate 섹션을 참조하세요.

disassociate-access-policy

다음 코드 예시는 disassociate-access-policy의 사용 방법을 보여 줍니다.

AWS CLI

액세스 항목에서 액세스 정책 연결을 해제하는 방법

다음 disassociate-access-policy는 액세스 항목과 연결된 액세스 정책을 제거합니다.

```
aws eks disassociate-access-policy \
    --cluster-name eks-customer \
    --principal-arn arn:aws:iam::111122223333:role/Admin \
    --policy-arn arn:aws:eks::aws:cluster-access-policy/AmazonEKSEditPolicy
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon EKS 사용 설명서의 액세스 정책을 액세스 항목과 연결을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisassociateAccessPolicy를 참조하세요.

disassociate-identity-provider-config

다음 코드 예시는 disassociate-identity-provider-config의 사용 방법을 보여 줍니다.

AWS CLI

Amazon EKS 클러스터에서 ID 제공업체 연결 해제

다음 disassociate-identity-provider-config 예시에서는 ID 제공업체를 Amazon EKS 클러스터에서 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 ID 제공업체에서 클러스터에 대한 사용자 인증 - 클러스터에서 OIDC ID 제공업체 연결 해제 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisassociateIdentityProviderConfig 섹션을 참조하세요.

get-token

다음 코드 예시는 get-token의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 'my-eks-cluster'라는 Amazon EKS 클러스터의 인증 토큰 가져오기

다음 get-token 예시에서는 my-eks-cluster라는 Amazon EKS 클러스터에 대한 인증 토큰을 가져옵니다.

```
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..."
    }
}
```

예시 2: 토큰에 서명할 때 자격 증명에 대해 이 roleARN을 수임하여 'my-eks-cluster'라는 Amazon EKS 클러스터의 인증 토큰 가져오기

다음 `get-token` 예시에서는 토큰에 서명할 때 자격 증명에 대해 이 roleARN을 수임하여 my-eks-cluster라는 Amazon EKS 클러스터에 대한 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..."
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetToken 섹션을 참조하세요.

`list-access-entries`

다음 코드 예시는 `list-access-entries`의 사용 방법을 보여 줍니다.

AWS CLI

EKS 클러스터의 액세스 항목을 나열하는 방법

다음 `list-access-entries`는 EKS 클러스터 `eks-customer`와 연결된 액세스 항목 목록을 반환합니다.

```
aws eks list-access-entries \
```

```
    --cluster-name eks-customer
```

출력:

```
{
    "accessEntries": [
        "arn:aws:iam::111122223333:role/Admin",
        "arn:aws:iam::111122223333:role/admin-test-ip",
        "arn:aws:iam::111122223333:role/assume-worker-node-role",
        "arn:aws:iam::111122223333:user/eks-admin-user"
    ]
}
```

자세한 내용은 Amazon EKS 사용 설명서의 IAM 사용자에게 EKS 액세스 항목을 사용하여 Kubernetes에 대한 액세스 권한 부여를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListAccessEntries를 참조하세요.

list-access-policies

다음 코드 예시는 `list-access-policies`의 사용 방법을 보여 줍니다.

AWS CLI

사용 가능한 모든 액세스 정책을 나열하는 방법

이 `list-access-policies` 예제에서는 사용 가능한 모든 액세스 정책의 목록을 반환합니다.

```
aws eks list-access-policies
```

출력:

```
{
    "accessPolicies": [
        {
            "name": "AmazonEKSAdminPolicy",
            "arn": "arn:aws:eks::aws:cluster-access-policy/AmazonEKSAdminPolicy"
        },
        {
            "name": "AmazonEKSAdminViewPolicy",
            "arn": "arn:aws:eks::aws:cluster-access-policy/AmazonEKSAdminViewPolicy"
        },
```

```
        {
            "name": "AmazonEKSAutoNodePolicy",
            "arn": "arn:aws:eks::aws:cluster-access-policy/AmazonEKSAutoNodePolicy"
        },
        {
            "name": "AmazonEKSBlockStorageClusterPolicy",
            "arn": "arn:aws:eks::aws:cluster-access-policy/
AmazonEKSBlockStorageClusterPolicy"
        },
        {
            "name": "AmazonEKSBlockStoragePolicy",
            "arn": "arn:aws:eks::aws:cluster-access-policy/
AmazonEKSBlockStoragePolicy"
        },
        {
            "name": "AmazonEKSClusterAdminPolicy",
            "arn": "arn:aws:eks::aws:cluster-access-policy/
AmazonEKSClusterAdminPolicy"
        },
        {
            "name": "AmazonEKSComputeClusterPolicy",
            "arn": "arn:aws:eks::aws:cluster-access-policy/
AmazonEKSComputeClusterPolicy"
        },
        {
            "name": "AmazonEKSComputePolicy",
            "arn": "arn:aws:eks::aws:cluster-access-policy/AmazonEKSComputePolicy"
        },
        {
            "name": "AmazonEKSEditPolicy",
            "arn": "arn:aws:eks::aws:cluster-access-policy/AmazonEKSEditPolicy"
        },
        {
            "name": "AmazonEKSHybridPolicy",
            "arn": "arn:aws:eks::aws:cluster-access-policy/AmazonEKSHybridPolicy"
        },
        {
            "name": "AmazonEKSLoadBalancingClusterPolicy",
            "arn": "arn:aws:eks::aws:cluster-access-policy/
AmazonEKSLoadBalancingClusterPolicy"
        },
        {
            "name": "AmazonEKSLoadBalancingPolicy",
```

```
            "arn": "arn:aws:eks::aws:cluster-access-policy/
AmazonEKSLoadBalancingPolicy"
        },
        {
            "name": "AmazonEKSNetworkingClusterPolicy",
            "arn": "arn:aws:eks::aws:cluster-access-policy/
AmazonEKSNetworkingClusterPolicy"
        },
        {
            "name": "AmazonEKSNetworkingPolicy",
            "arn": "arn:aws:eks::aws:cluster-access-policy/
AmazonEKSNetworkingPolicy"
        },
        {
            "name": "AmazonEKSViewPolicy",
            "arn": "arn:aws:eks::aws:cluster-access-policy/AmazonEKSViewPolicy"
        },
        {
            "name": "AmazonEMRJobPolicy",
            "arn": "arn:aws:eks::aws:cluster-access-policy/AmazonEMRJobPolicy"
        },
        {
            "name": "AmazonSagemakerHyperpodClusterPolicy",
            "arn": "arn:aws:eks::aws:cluster-access-policy/
AmazonSagemakerHyperpodClusterPolicy"
        },
        {
            "name": "AmazonSagemakerHyperpodControllerPolicy",
            "arn": "arn:aws:eks::aws:cluster-access-policy/
AmazonSagemakerHyperpodControllerPolicy"
        },
        {
            "name": "AmazonSagemakerHyperpodSystemNamespacePolicy",
            "arn": "arn:aws:eks::aws:cluster-access-policy/
AmazonSagemakerHyperpodSystemNamespacePolicy"
        }
    ]
}
```

자세한 내용은 Amazon EKS 사용 설명서의 액세스 정책을 액세스 항목과 연결을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListAccessPolicies 섹션을 참조하세요.

list-addons

다음 코드 예시는 `list-addons`의 사용 방법을 보여 줍니다.

AWS CLI

'my-eks-cluster'라는 Amazon EKS 클러스터에 설치된 모든 추가 기능 나열

다음 `list-addons` 예시에서는 my-eks-cluster라는 Amazon EKS 클러스터에 설치된 모든 추가 기능을 나열합니다.

```
aws eks list-addons \
    --cluster-name my-eks-cluster
```

출력:

```
{
    "addons": [
        "kube-proxy",
        "vpc-cni"
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListAddons 섹션을 참조하세요.

list-associated-access-policies

다음 코드 예시는 `list-associated-access-policies`의 사용 방법을 보여 줍니다.

AWS CLI

액세스 항목과 연결된 액세스 정책을 나열하는 방법

다음 `list-associated-access-policies` 예제에서는 EKS 클러스터의 액세스 항목과 연결된 액세스 정책 목록을 반환합니다.

```
aws eks list-associated-access-policies \
    --cluster-name eks-customer \
    --principal-arn arn:aws:iam::111122223333:role/Admin
```

출력:

```
{
    "associatedAccessPolicies": [
        {
            "policyArn": "arn:aws:eks::aws:cluster-access-policy/
AmazonEKSAdminPolicy",
            "accessScope": {
                "type": "cluster",
                "namespaces": []
            },
            "associatedAt": "2025-05-24T17:26:22.935000-05:00",
            "modifiedAt": "2025-05-24T17:26:22.935000-05:00"
        }
    ],
    "clusterName": "eks-customer",
    "principalArn": "arn:aws:iam::111122223333:role/Admin"
}
```

자세한 내용은 Amazon EKS 사용 설명서의 IAM 사용자에게 EKS 액세스 항목을 사용하여 Kubernetes에 대한 액세스 권한 부여를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListAssociatedAccessPolicies를 참조하세요.

`list-clusters`

다음 코드 예시는 `list-clusters`의 사용 방법을 보여 줍니다.

AWS CLI

'my-eks-cluster'라는 Amazon EKS 클러스터에 설치된 모든 추가 기능을 나열하려면

다음 `list-clusters` 예시에서는 my-eks-cluster라는 Amazon EKS 클러스터에 설치된 모든 추가 기능을 나열합니다.

```
aws eks list-clusters
```

출력:

```
{
    "clusters": [
        "prod",
        "qa",
        "stage",
```

```
        "my-eks-cluster"
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListClusters를 참조하세요.

list-fargate-profiles

다음 코드 예시는 list-fargate-profiles의 사용 방법을 보여 줍니다.

AWS CLI

'my-eks-cluster'라는 Amazon EKS 클러스터의 모든 Fargate 프로파일을 나열하려면

다음 list-fargate-profiles 예시에서는 my-eks-cluster라는 Amazon EKS 클러스터의 모든 Fargate 프로파일을 나열합니다.

```
aws eks list-fargate-profiles \
    --cluster-name my-eks-cluster
```

출력:

```
{
    "fargateProfileNames": [
        "my-fargate-profile"
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListFargateProfiles 섹션을 참조하세요.

list-identity-provider-configs

다음 코드 예시는 list-identity-provider-configs의 사용 방법을 보여 줍니다.

AWS CLI

Amazon EKS 클러스터에 연결된 ID 제공업체 나열

다음 list-identity-provider-configs 예시에서는 Amazon EKS 클러스터에 연결된 ID 제공업체를 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 ID 제공업체에서 클러스터에 대한 사용자 인증 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListIdentityProviderConfigs 섹션을 참조하세요.

list-insights

다음 코드 예시는 `list-insights`의 사용 방법을 보여 줍니다.

AWS CLI

지정된 클러스터에 대한 모든 인사이트를 나열하는 방법

다음 `list-insights` 예제에서는 지정된 클러스터에 대해 확인된 모든 인사이트의 목록을 반환합니다.

```
aws eks list-insights \
--cluster-name eks-customer
```

출력:

```
{
    "insights": [
        {
            "id": "38ea7a64-a14f-4e0e-95c7-8dbcab3c3616",
            "name": "Kubelet version skew",
            "category": "UPGRADE_READINESS",
            "kubernetesVersion": "1.33",
            "lastRefreshTime": "2025-05-24T11:22:50-05:00",
```

```
            "lastTransitionTime": "2025-05-24T11:22:50-05:00",
            "description": "Checks for kubelet versions of worker nodes in the
 cluster to see if upgrade would cause noncompliance with supported Kubernetes
 kubelet version skew policy.",
            "insightStatus": {
                "status": "PASSING",
                "reason": "Node kubelet versions match the cluster control plane
 version."
            }
        },
        {
            "id": "9cd91472-f99c-45a9-b7d7-54d4900dee23",
            "name": "EKS add-on version compatibility",
            "category": "UPGRADE_READINESS",
            "kubernetesVersion": "1.33",
            "lastRefreshTime": "2025-05-24T11:22:59-05:00",
            "lastTransitionTime": "2025-05-24T11:22:50-05:00",
            "description": "Checks version of installed EKS add-ons to ensure they
 are compatible with the next version of Kubernetes. ",
            "insightStatus": {
                "status": "PASSING",
                "reason": "All installed EKS add-on versions are compatible with
 next Kubernetes version."
            }
        },
        {
            "id": "0deb269d-b1e1-458c-a2b4-7a57f940c875",
            "name": "Cluster health issues",
            "category": "UPGRADE_READINESS",
            "kubernetesVersion": "1.33",
            "lastRefreshTime": "2025-05-24T11:22:59-05:00",
            "lastTransitionTime": "2025-05-24T11:22:50-05:00",
            "description": "Checks for any cluster health issues that prevent
 successful upgrade to the next Kubernetes version on EKS.",
            "insightStatus": {
                "status": "PASSING",
                "reason": "No cluster health issues detected."
            }
        },
        {
            "id": "214fa274-344f-420b-812a-5049ce72c9ww",
            "name": "kube-proxy version skew",
            "category": "UPGRADE_READINESS",
            "kubernetesVersion": "1.33",
```

```
            "lastRefreshTime": "2025-05-24T11:22:50-05:00",
            "lastTransitionTime": "2025-05-24T11:22:50-05:00",
            "description": "Checks version of kube-proxy in cluster to see if
 upgrade would cause noncompliance with supported Kubernetes kube-proxy version skew
 policy.",
            "insightStatus": {
                "status": "PASSING",
                "reason": "kube-proxy versions match the cluster control plane
 version."
            }
        }
    ]
}
```

자세한 내용은 Amazon EKS 사용 설명서의 [클러스터 인사이트 보기](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ListInsights](#)를 참조하세요.

list-nodegroups

다음 코드 예시는 `list-nodegroups`의 사용 방법을 보여 줍니다.

AWS CLI

Amazon EKS 클러스터의 모든 노드 그룹 나열

다음 `list-nodegroups` 예시에서는 Amazon EKS 클러스터에의 모든 노드 그룹을 나열합니다.

```
aws eks list-nodegroups \
    --cluster-name my-eks-cluster
```

출력:

```
{
    "nodegroups": [
        "my-eks-managed-node-group",
        "my-eks-nodegroup"
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [ListNodegroups](#) 섹션을 참조하세요.

list-pod-identity-associations

다음 코드 예시는 list-pod-identity-associations의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: EKS 클러스터의 Pod Identity 연결을 나열하는 방법

다음 list-pod-identity-associations는 모든 네임스페이스 및 서비스 계정에서 EKS 클러스터 eks-customer와 연결된 Pod Identity 연결 목록을 반환합니다.

```
aws eks list-pod-identity-associations \
    --cluster-name eks-customer
```

출력:

```
{
    "associations": [
        {
            "clusterName": "eks-customer",
            "namespace": "default",
            "serviceAccount": "default",
            "associationArn": "arn:aws:eks:us-
west-2:111122223333:podidentityassociation/eks-customer/a-9njjin9gfghecgocd",
            "associationId": "a-9njjin9gfghecgocd"
        },
        {
            "clusterName": "eks-customer",
            "namespace": "kube-system",
            "serviceAccount": "eks-customer",
            "associationArn": "arn:aws:eks:us-
west-2:111122223333:podidentityassociation/eks-customer/a-dvtacahdvjn01ffbc",
            "associationId": "a-dvtacahdvjn01ffbc"
        },
        {
            "clusterName": "eks-customer",
            "namespace": "kube-system",
            "serviceAccount": "coredns",
            "associationArn": "arn:aws:eks:us-
west-2:111122223333:podidentityassociation/eks-customer/a-yrpsdroc4ei7k6xps",
            "associationId": "a-yrpsdroc4ei7k6xps"
        }
    ]
```

```
}
```

자세한 내용은 Amazon EKS 사용 설명서의 EKS Pod Identity가 포드에 AWS 서비스에 대한 액세스 권한을 부여하는 방법 알아보기를 참조하세요.

예제 2: 네임스페이스 및 서비스 계정을 기반으로 EKS 클러스터의 Pod Identity 연결을 나열하는 방법

다음 `list-pod-identity-associations`는 네임스페이스 및 서비스 계정을 기반으로 EKS 클러스터의 Pod Identity 연결 목록을 반환합니다.

```
aws eks list-pod-identity-associations \
    --cluster-name eks-customer \
    --namespace kube-system \
    --service-account eks-customer
```

출력:

```
{
    "associations": [
        {
            "clusterName": "eks-customer",
            "namespace": "kube-system",
            "serviceAccount": "eks-customer",
            "associationArn": "arn:aws:eks:us-
west-2:111122223333:podidentityassociation/eks-customer/a-dvtacahdvjn01ffbc",
            "associationId": "a-dvtacahdvjn01ffbc"
        }
    ]
}
```

자세한 내용은 Amazon EKS 사용 설명서의 EKS Pod Identity가 포드에 AWS 서비스에 대한 액세스 권한을 부여하는 방법 알아보기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListPodIdentityAssociations를 참조하세요.

`list-tags-for-resource`

다음 코드 예시는 `list-tags-for-resource`의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: Amazon EKS 클러스터 ARN의 모든 태그를 나열하려면

다음 `list-tags-for-resource` 예시에서는 Amazon EKS 클러스터 ARN에 대한 모든 태그를 나열합니다.

```
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"
    }
}
```

예시 2: Amazon EKS 노드 그룹 ARN의 모든 태그를 나열하려면

다음 `list-tags-for-resource` 예시에서는 Amazon EKS 노드 그룹 ARN의 모든 태그를 나열합니다.

```
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"
    }
}
```

예시 3: Amazon EKS Fargate 프로파일 ARN의 모든 태그를 나열하려면

다음 `list-tags-for-resource` 예시에서는 Amazon EKS Fargate 프로파일 ARN의 모든 태그를 나열합니다.

```
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"
    }
}
```

예시 4: Amazon EKS 추가 기능 ARN의 모든 태그를 나열하려면

다음 `list-tags-for-resource` 예시에서는 Amazon EKS 추가 기능 ARN의 모든 태그를 나열합니다.

```
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"
    }
}
```

예시 5: Amazon EKS OIDC ID 제공업체 ARN의 모든 태그를 나열하려면

다음 `list-tags-for-resource` 예시에서는 Amazon EKS OIDC ID 제공업체 ARN의 모든 태그를 나열합니다.

```
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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource 섹션을 참조하세요.

`list-update`

다음 코드 예시는 `list-update`의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: Amazon EKS 클러스터 이름과 연결된 업데이트를 나열하려면

다음 `list-updates` 예시에서는 Amazon EKS 클러스터 이름의 모든 업데이트 ID를 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"
    ]
}
```

예시 2: Amazon EKS 노드 그룹의 모든 업데이트 ID를 나열하려면

다음 `list-updates` 예시에서는 Amazon EKS 노드 그룹의 모든 업데이트 ID를 나열합니다.

```
aws eks list-updates \
    --name my-eks-cluster \
    --nodegroup-name my-eks-managed-node-group
```

출력:

```
{
    "updateIds": [
        "8c6c1bef-61fe-42ac-a242-89412387b8e7"
    ]
}
```

예시 3: Amazon EKS 추가 기능의 모든 업데이트 ID를 나열하려면

다음 `list-updates` 예시에서는 Amazon EKS 추가 기능의 모든 업데이트 ID를 나열합니다.

```
aws eks list-updates \
```

```
    --name my-eks-cluster \
    --addon-name vpc-cni
```

출력:

```
{
    "updateIds": [
        "9cdba8d4-79fb-3c83-afe8-00b508d33268"
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListUpdate 섹션을 참조하세요.

list-updates

다음 코드 예시는 list-updates의 사용 방법을 보여 줍니다.

AWS CLI

클러스터에 대한 업데이트를 나열하려면

이 예시 명령은 기본 리전에 example 클러스터의 현재 업데이트를 나열합니다.

명령:

```
aws eks list-updates --name example
```

출력:

```
{
    "updateIds": [
        "10bddb13-a71b-425a-b0a6-71cd03e59161"
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListUpdates 섹션을 참조하세요.

register-cluster

다음 코드 예시는 register-cluster의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: Amazon EKS에 외부 EKS_ANYWHERE Kubernetes 클러스터 등록

다음 `register-cluster` 예시에서는 Amazon EKS에 외부 EKS_ANYWHERE Kubernetes 클러스터를 등록합니다.

```
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 생성 섹션을 참조하세요.

예시 2: Amazon EKS에 외부 Kubernetes 클러스터 등록

다음 `register-cluster` 예시에서는 Amazon EKS에 외부 EKS_ANYWHERE Kubernetes 클러스터를 등록합니다.

```
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 생성 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RegisterCluster 섹션을 참조하세요.

tag-resource

다음 코드 예시는 tag-resource의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: Amazon EKS 클러스터에 지정된 태그를 추가하려면

다음 tag-resource 예시에서는 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.

예시 2: Amazon EKS 노드 그룹에 지정된 태그를 추가하려면

다음 tag-resource 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

untag-resource

다음 코드 예시는 `untag-resource`의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: Amazon EKS 클러스터에서 지정된 태그를 삭제하려면

다음 `untag-resource` 예시에서는 지정된 태그를 Amazon EKS 클러스터에서 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.

예시 2: Amazon EKS 노드 그룹에서 지정된 태그를 삭제하려면

다음 `untag-resource` 예시에서는 지정된 태그를 Amazon EKS 노드 그룹에서 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.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

update-access-entry

다음 코드 예시는 `update-access-entry`의 사용 방법을 보여 줍니다.

AWS CLI

EKS 클러스터에 대한 액세스 항목을 업데이트하는 방법

다음 `update-access-entry`는 Kubernetes 그룹 `tester`를 추가하여 EKS 클러스터의 액세스 항목을 업데이트합니다.

```
aws eks update-access-entry \
    --cluster-name eks-customer \
    --principal-arn arn:aws:iam::111122223333:role/Admin \
    --kubernetes-groups tester
```

출력:

```
{
    "accessEntry": {
        "clusterName": "eks-customer",
        "principalArn": "arn:aws:iam::111122223333:role/Admin",
        "kubernetesGroups": [
            "tester"
        ],
        "accessEntryArn": "arn:aws:eks:us-west-2:111122223333:access-entry/eks-
customer/role/111122223333/Admin/d2cb8183-d6ec-b82a-d967-eca21902a4b4",
        "createdAt": "2025-05-24T11:02:04.432000-05:00",
        "modifiedAt": "2025-05-24T17:08:01.608000-05:00",
        "tags": {},
        "username": "arn:aws:sts::111122223333:assumed-role/Admin/{{SessionName}}",
        "type": "STANDARD"
    }
}
```

자세한 내용은 Amazon EKS 사용 설명서의 [액세스 항목 업데이트](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [UpdateAccessEntry](#)를 참조하세요.

`update-addon`

다음 코드 예시는 `update-addon`의 사용 방법을 보여 줍니다.

AWS CLI

예시 1. 서비스 계정 역할 ARN으로 Amazon EKS 추가 기능을 업데이트하려면

다음 update-addon 예시 명령은 서비스 계정 역할 ARN으로 Amazon EKS 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
```

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 EKS 사용 설명서의 Amazon EKS 추가 기능 관리 - 추가 기능 업데이트 섹션을 참조하세요.

예시 2. 특정 추가 기능 버전으로 Amazon EKS 추가 기능을 업데이트하려면

다음 update-addon 예시 명령은 특정 추가 기능 버전으로 Amazon EKS 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 EKS 사용 설명서의 [Amazon EKS 추가 기능 관리 - 추가 기능 업데이트] 섹션을 참조하세요.

예시 3. 사용자 지정 구성 값으로 Amazon EKS 추가 기능을 업데이트하고 충돌 세부 정보를 해결하려면

다음 `update-addon` 예시 명령은 Amazon EKS 추가 기능을 사용자 지정 구성 값으로 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 EKS 사용 설명서의 [Amazon EKS 추가 기능 관리 - 추가 기능 업데이트] 섹션을 참조하세요.

예시 4. 사용자 지정 JSON 구성 값 파일로 Amazon EKS 추가 기능을 업데이트하려면

다음 `update-addon` 예시 명령은 Amazon EKS 추가 기능을 사용자 지정 JSON 구성 값으로 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.json`의 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 EKS 사용 설명서의 Amazon EKS 추가 기능 관리 - 추가 기능 업데이트 섹션을 참조하세요.

예시 5. 사용자 지정 YAML 구성 값 파일로 Amazon EKS 추가 기능을 업데이트하려면

다음 update-addon 예시 명령은 Amazon EKS 추가 기능을 사용자 지정 YAML 구성 값으로 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 EKS 사용 설명서의 Amazon EKS 추가 기능 관리 - 추가 기능 업데이트 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateAddon 섹션을 참조하세요.

`update-cluster-config`

다음 코드 예시는 `update-cluster-config`의 사용 방법을 보여 줍니다.

AWS CLI

클러스터 엔드포인트 액세스를 업데이트하려면

이 예시 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 예시 명령은 example 클러스터에 대한 모든 클러스터 컨트롤 플레인 로깅 유형을 활성화합니다.

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": []
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateClusterConfig 섹션을 참조하세요.

update-cluster-version

다음 코드 예시는 `update-cluster-version`의 사용 방법을 보여 줍니다.

AWS CLI

'my-eks-cluster'라는 Amazon EKS 클러스터를 지정된 Kubernetes 버전으로 업데이트하려면

다음 `update-cluster-version` 예시에서는 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 EKS 사용 설명서의 Amazon EKS 클러스터 Kubernetes 버전 업데이트 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateClusterVersion 섹션을 참조하세요.

update-kubeconfig

다음 코드 예시는 update-kubeconfig의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 'my-eks-cluster'라는 Amazon EKS 클러스터에 연결할 수 있도록 kubeconfig를 만들거나 업데이트하여 kubectl 구성

다음 update-kubeconfig 예시에서는 my-eks-cluster라는 Amazon EKS 클러스터에 연결할 수 있도록 kubeconfig를 만들거나 업데이트하여 kubectl을 구성합니다.

```
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 EKS 사용 설명서의 [Amazon EKS 클러스터에 대한 kubeconfig 파일 생성 또는 업데이트](#) 섹션을 참조하세요.

예시 2: 'my-eks-cluster'라는 Amazon EKS 클러스터에 연결할 수 있도록 kubeconfig(클러스터 인증을 위한 역할을 수임하는 role-arn 옵션 포함)를 만들거나 업데이트하여 kubectl을 구성

다음 `update-kubeconfig` 예시에서는 my-eks-cluster라는 Amazon EKS 클러스터에 연결할 수 있도록 kubeconfig(클러스터 인증을 위한 역할을 수임하는 role-arn 옵션 포함)를 만들거나 업데이트하여 kubectl을 구성합니다.

```
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 EKS 사용 설명서의 [Amazon EKS 클러스터에 대한 kubeconfig 파일 생성 또는 업데이트](#) 섹션을 참조하세요.

예시 3: `my-eks-cluster`라는 Amazon EKS 클러스터에 연결할 수 있도록 kubeconfig(사용자 지정 클러스터 별칭 및 user-alias와 함께 클러스터 인증을 위한 역할을 수임하는 role-arn 옵션 포함)를 만들거나 업데이트하여 kubectl 구성

다음 `update-kubeconfig` 예시에서는 my-eks-cluster라는 Amazon EKS 클러스터에 연결할 수 있도록 kubeconfig(사용자 지정 클러스터 별칭 및 user-alias와 함께 클러스터 인증을 위한 역할을 수임하는 role-arn 옵션 포함)를 만들거나 업데이트하여 kubectl을 구성합니다.

```
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 EKS 사용 설명서의 [Amazon EKS 클러스터에 대한 kubeconfig 파일 생성 또는 업데이트](#) 섹션을 참조하세요.

예시 4: 검토를 위해 kubeconfig 파일 항목을 출력하고 'my-eks-cluster'라는 Amazon EKS 클러스터에 연결할 수 있도록 kubectl 구성

다음 `update-kubeconfig` 예시에서는 my-eks-cluster라는 Amazon EKS 클러스터에 연결할 수 있도록 kubeconfig(사용자 지정 클러스터 별칭 및 user-alias와 함께 클러스터 인증을 위한 역할을 수임하는 role-arn 옵션 포함)를 만들거나 업데이트하여 kubectl을 구성합니다.

```
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 EKS 사용 설명서의 Amazon EKS 클러스터에 대한 kubeconfig 파일 생성 또는 업데이트 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateKubeconfig 섹션을 참조하세요.

update-nodegroup-config

다음 코드 예시는 update-nodegroup-config의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: Amazon EKS 클러스터의 EKS 워커 노드에 새 레이블과 테인트를 추가하도록 관리형 노드 그룹 업데이트

다음 update-nodegroup-config 예시에서는 관리형 노드 그룹을 업데이트하여 Amazon EKS 클러스터의 EKS 워커 노드에 새 레이블과 테인트를 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 업데이트](#) 섹션을 참조하세요.

예시 2: Amazon EKS 클러스터의 EKS 워커 노드에 대한 레이블 및 테인트를 제거하도록 관리형 노드 그룹 업데이트

다음 `update-nodegroup-config` 예시에서는 Amazon EKS 클러스터의 EKS 워커 노드에 대한 레이블과 테인트를 제거하도록 관리형 노드 그룹을 업데이트합니다.

```
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 업데이트 섹션을 참조하세요.

예시 3: Amazon EKS 클러스터의 EKS 워커 노드에 대한 레이블 및 테인트를 제거 및 추가하도록 관리형 노드 그룹 업데이트

다음 `update-nodegroup-config` 예시에서는 Amazon EKS 클러스터의 EKS 워커 노드에 대한 레이블과 테인트를 제거하고 추가하도록 관리형 노드 그룹을 업데이트합니다.

```
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 업데이트 섹션을 참조하세요.

예시 4: Amazon EKS 클러스터의 EKS 워커 노드에 대한 scaling-config 및 update-config를 업데이트하도록 관리형 노드 그룹 업데이트

다음 `update-nodegroup-config` 예시에서는 Amazon EKS 클러스터의 EKS 워커 노드에 대한 scaling-config 및 update-config를 업데이트하도록 관리형 노드 그룹을 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 업데이트 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateNodegroupConfig 섹션을 참조하세요.

`update-nodegroup-version`

다음 코드 예시는 `update-nodegroup-version`의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: Amazon EKS 관리형 노드 그룹의 Kubernetes 버전 또는 AMI 버전 업데이트

다음 `update-nodegroup-version` 예시에서는 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 업데이트 섹션을 참조하세요.

예시 2: Amazon EKS 관리형 노드 그룹의 Kubernetes 버전 또는 AMI 버전 업데이트

다음 `update-nodegroup-version` 예시에서는 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>` 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateNodegroupVersion 섹션을 참조하세요.

`update-pod-identity-association`

다음 코드 예시는 `update-pod-identity-association`의 사용 방법을 보여 줍니다.

AWS CLI

EKS Pod Identity 연결을 업데이트하는 방법

다음 `update-pod-identity-association` 예제에서는 연결 ID `a-9njjin9gfghecgocd`에 대해 연결된 IAM 역할을 `arn:aws:iam::111122223333:role/my-role`에서 `arn:aws:iam::111122223333:role/s3-role`로 변경하여 EKS Pod Identity 연결을 업데이트합니다. 이 API는 연결된 IAM 역할만 업데이트할 수 있습니다.

```
aws eks update-pod-identity-association \
    --cluster-name eks-customer \
```

```
    --association-id a-9njjin9gfghecgocd \
    --role-arn arn:aws:iam::111122223333:role/s3-role
```

출력:

```
{
    "association": {
        "clusterName": "eks-customer",
        "namespace": "default",
        "serviceAccount": "default",
        "roleArn": "arn:aws:iam::111122223333:role/s3-role",
        "associationArn": "arn:aws:eks:us-
west-2:111122223333:podidentityassociation/eks-customer/a-9njjin9gfghecgocd",
        "associationId": "a-9njjin9gfghecgocd",
        "tags": {
            "Key2": "value2",
            "Key1": "value1"
        },
        "createdAt": "2025-05-24T19:52:14.135000-05:00",
        "modifiedAt": "2025-05-25T21:01:53.120000-05:00"
    }
}
```

자세한 내용은 Amazon EKS 사용 설명서의 EKS Pod Identity가 포드에 AWS 서비스에 대한 액세스 권한을 부여하는 방법 알아보기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdatePodIdentityAssociation을 참조하세요.

AWS CLI를 사용한 Elastic Beanstalk 예제

다음 코드 예제는 Elastic Beanstalk와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

작업

`abort-environment-update`

다음 코드 예시는 `abort-environment-update`의 사용 방법을 보여 줍니다.

AWS CLI

배포를 중단하려면

다음 명령은 `my-env` 환경에 대해 실행 중인 애플리케이션 버전 배포를 중단합니다.

```
aws elasticbeanstalk abort-environment-update --environment-name my-env
```

- API 세부 정보는 AWS CLI 명령 참조의 AbortEnvironmentUpdate를 참조하세요.

`check-dns-availability`

다음 코드 예시는 `check-dns-availability`의 사용 방법을 보여 줍니다.

AWS CLI

CNAME의 가용성을 확인하려면

다음 명령은 하위 도메인 `my-cname.elasticbeanstalk.com`의 가용성을 확인합니다.

```
aws elasticbeanstalk check-dns-availability --cname-prefix my-cname
```

출력:

```
{
    "Available": true,
    "FullyQualifiedCNAME": "my-cname.elasticbeanstalk.com"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CheckDnsAvailability를 참조하세요.

`create-application-version`

다음 코드 예시는 `create-application-version`의 사용 방법을 보여 줍니다.

AWS CLI

새 애플리케이션 버전을 만들려면

다음 명령은 'MyApp'이라는 애플리케이션의 새 버전인 'v1'을 만듭니다.

```
aws elasticbeanstalk create-application-version --application-name MyApp --
version-label v1 --description MyAppv1 --source-bundle S3Bucket="amzn-s3-demo-
bucket",S3Key="sample.war" --auto-create-application
```

애플리케이션이 아직 없는 경우 auto-create-application 옵션으로 인해 애플리케이션이 자동으로 만들어집니다. 소스 번들은 "amzn-s3-demo-bucket"이라는 이름의 s3 버킷에 저장된 .war 파일로, Apache Tomcat 샘플 애플리케이션이 포함되어 있습니다.

출력:

```
{
  "ApplicationVersion": {
      "ApplicationName": "MyApp",
      "VersionLabel": "v1",
      "Description": "MyAppv1",
      "DateCreated": "2015-02-03T23:01:25.412Z",
      "DateUpdated": "2015-02-03T23:01:25.412Z",
      "SourceBundle": {
          "S3Bucket": "amzn-s3-demo-bucket",
          "S3Key": "sample.war"
      }
  }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateApplicationVersion을 참조하세요.

`create-application`

다음 코드 예시는 `create-application`의 사용 방법을 보여 줍니다.

AWS CLI

새 애플리케이션을 만들려면

다음 명령은 'MyApp'이라는 새 애플리케이션을 생성합니다.

```
aws elasticbeanstalk create-application --application-name MyApp --description "my
 application"
```

create-application 명령은 애플리케이션의 이름과 설명만 구성합니다. 애플리케이션의 소스 코드를 업로드하려면 create-application-version을 사용하여 애플리케이션의 초기 버전을 만듭니다. 또한, create-application-version에는 애플리케이션과 애플리케이션 버전을 한 번에 만들 수 있는 auto-create-application 옵션도 있습니다.

출력:

```
{
  "Application": {
      "ApplicationName": "MyApp",
      "ConfigurationTemplates": [],
      "DateUpdated": "2015-02-12T18:32:21.181Z",
      "Description": "my application",
      "DateCreated": "2015-02-12T18:32:21.181Z"
  }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateApplication 섹션을 참조하세요.

create-configuration-template

다음 코드 예시는 create-configuration-template의 사용 방법을 보여 줍니다.

AWS CLI

구성 템플릿을 만들려면

다음 명령은 ID가 e-rpqsewtp2j인 환경에 적용된 설정으로 my-app-v1 구성 템플릿을 생성합니다.

```
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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [CreateConfigurationTemplate](https://docs.aws.amazon.com/goto/aws-cli/elasticbeanstalk-2010-12-01/CreateConfigurationTemplate)을 참조하세요.

`create-environment`

다음 코드 예시는 `create-environment`의 사용 방법을 보여 줍니다.

AWS CLI

애플리케이션을 위한 새 환경을 만들려면

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"
}
```

v1은 이전에 create-application-version으로 업로드한 애플리케이션 버전의 레이블입니다.

환경 구성 옵션을 정의할 JSON 파일을 지정하려면

다음 `create-environment` 명령은 솔루션 스택 또는 구성 템플릿에서 가져온 값을 재정의하는 데 `myoptions.json` JSON 파일을 사용하도록 지정합니다.

```
aws elasticbeanstalk create-environment --environment-name sample-env --application-name sampleapp --option-settings file://myoptions.json
```

`myoptions.json`은 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 안내서에서 Option Values를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateEnvironment를 참조하세요.

`create-storage-location`

다음 코드 예시는 `create-storage-location`의 사용 방법을 보여 줍니다.

AWS CLI

스토리지 위치를 만들려면

다음 명령은 Amazon S3에 스토리지 위치를 만듭니다.

```
aws elasticbeanstalk create-storage-location
```

출력:

```
{
    "S3Bucket": "elasticbeanstalk-us-west-2-0123456789012"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [CreateStorageLocation](#)을 참조하세요.

`delete-application-version`

다음 코드 예시는 `delete-application-version`의 사용 방법을 보여 줍니다.

AWS CLI

애플리케이션 버전을 삭제하려면

다음 명령은 `my-app` 애플리케이션의 `22a0-stage-150819_182129` 애플리케이션 버전을 삭제합니다.

```
aws elasticbeanstalk delete-application-version --version-label 22a0-
stage-150819_182129 --application-name my-app
```

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteApplicationVersion](#)을 참조하세요.

`delete-application`

다음 코드 예시는 `delete-application`의 사용 방법을 보여 줍니다.

AWS CLI

애플리케이션 삭제

다음 명령은 `my-app` 애플리케이션을 삭제합니다.

```
aws elasticbeanstalk delete-application --application-name my-app
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteApplication을 참조하세요.

delete-configuration-template

다음 코드 예시는 delete-configuration-template의 사용 방법을 보여 줍니다.

AWS CLI

구성 템플릿을 삭제하려면

다음 명령은 my-app 애플리케이션에 대한 my-template 구성 템플릿을 삭제합니다.

```
aws elasticbeanstalk delete-configuration-template --template-name my-template --
application-name my-app
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteConfigurationTemplate을 참조하세요.

delete-environment-configuration

다음 코드 예시는 delete-environment-configuration의 사용 방법을 보여 줍니다.

AWS CLI

초안 구성을 삭제하려면

다음 명령은 my-env 환경에 대한 초안 구성을 삭제합니다.

```
aws elasticbeanstalk delete-environment-configuration --environment-name my-env --
application-name my-app
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteEnvironmentConfiguration을 참조하세요.

describe-application-versions

다음 코드 예시는 describe-application-versions의 사용 방법을 보여 줍니다.

AWS CLI

애플리케이션 버전 정보를 보려면

다음 명령은 v2 애플리케이션 버전에 대한 정보를 검색합니다.

```
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"
          }
      }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeApplicationVersions를 참조하세요.

describe-applications

다음 코드 예시는 describe-applications의 사용 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeApplications](https://docs.aws.amazon.com/goto/aws-cli/elasticbeanstalk-2010-12-01/DescribeApplications)를 참조하세요.

describe-configuration-options

다음 코드 예시는 describe-configuration-options의 사용 방법을 보여 줍니다.

AWS CLI

환경에 대한 구성 옵션을 보려면

다음 명령은 my-env 환경에 사용 가능한 모든 구성 옵션에 대한 설명을 검색합니다.

```
aws elasticbeanstalk describe-configuration-options --environment-name my-env --
application-name my-app
```

출력(간략한 버전):

```
{
    "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 안내서에서 Option Values를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeConfigurationOptions](https://docs.aws.amazon.com/cli/latest/reference/elasticbeanstalk/describe-configuration-options.html)를 참조하세요.

describe-configuration-settings

다음 코드 예시는 describe-configuration-settings의 사용 방법을 보여 줍니다.

AWS CLI

환경에 대한 구성 설정을 보려면

다음 명령은 my-env 환경에 대한 구성 설정을 검색합니다.

```
aws elasticbeanstalk describe-configuration-settings --environment-name my-env --
application-name my-app
```

출력(간략한 버전):

```
{
    "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 안내서에서 Option Values를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeConfigurationSettings를 참조하세요.

describe-environment-health

다음 코드 예시는 describe-environment-health의 사용 방법을 보여 줍니다.

AWS CLI

환경 상태를 보려면

다음 명령은 my-env 환경에 대한 전체 상태 정보를 검색합니다.

```
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 안내서에서 Enhanced Health Reporting and Monitoring을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeEnvironmentHealth를 참조하세요.

`describe-environment-resources`

다음 코드 예시는 `describe-environment-resources`의 사용 방법을 보여 줍니다.

AWS CLI

환경의 AWS 리소스에 대한 정보를 보려면

다음 명령은 `my-env` 환경의 리소스에 대한 정보를 검색합니다.

```
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"
            }
        ]
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeEnvironmentResources를 참조하세요.

`describe-environments`

다음 코드 예시는 `describe-environments`의 사용 방법을 보여 줍니다.

AWS CLI

환경에 대한 정보를 보려면

다음 명령은 `my-env` 환경에 대한 정보를 검색합니다.

```
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"
            }
        ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeEnvironments를 참조하세요.

describe-events

다음 코드 예시는 describe-events의 사용 방법을 보여 줍니다.

AWS CLI

환경에 대한 이벤트를 보려면

다음 명령은 my-env 환경에 대한 이벤트를 검색합니다.

```
aws elasticbeanstalk describe-events --environment-name my-env
```

출력(간략한 버전):

```
{
    "Events": [
        {
            "ApplicationName": "my-app",
            "EnvironmentName": "my-env",
            "Message": "Environment health has transitioned from Info to Ok.",
            "EventDate": "2015-08-20T07:06:53.535Z",
            "Severity": "INFO"
        },
        {
            "ApplicationName": "my-app",
            "EnvironmentName": "my-env",
            "Severity": "INFO",
            "RequestId": "b7f3960b-4709-11e5-ba1e-07e16200da41",
            "Message": "Environment update completed successfully.",
```

```
            "EventDate": "2015-08-20T07:06:02.049Z"
        },
        ...
        {
            "ApplicationName": "my-app",
            "EnvironmentName": "my-env",
            "Severity": "INFO",
            "RequestId": "ca8dfbf6-41ef-11e5-988b-651aa638f46b",
            "Message": "Using elasticbeanstalk-us-west-2-012445113685 as Amazon S3
 storage bucket for environment data.",
            "EventDate": "2015-08-13T19:16:27.561Z"
        },
        {
            "ApplicationName": "my-app",
            "EnvironmentName": "my-env",
            "Severity": "INFO",
            "RequestId": "cdfba8f6-41ef-11e5-988b-65638f41aa6b",
            "Message": "createEnvironment is starting.",
            "EventDate": "2015-08-13T19:16:26.581Z"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeEvents를 참조하세요.

describe-instances-health

다음 코드 예시는 describe-instances-health의 사용 방법을 보여 줍니다.

AWS CLI

환경 상태를 보려면

다음 명령은 my-env 환경의 인스턴스에 대한 상태 정보를 검색합니다.

```
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 안내서에서 Enhanced Health Reporting and Monitoring을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeInstancesHealth를 참조하세요.

`list-available-solution-stacks`

다음 코드 예시는 `list-available-solution-stacks`의 사용 방법을 보여 줍니다.

AWS CLI

솔루션 스택을 보려면

다음 명령은 현재 사용 가능한 모든 플랫폼 구성과 과거에 사용했던 모든 플랫폼 구성에 대한 솔루션 스택을 나열합니다.

```
aws elasticbeanstalk list-available-solution-stacks
```

출력(간략한 버전):

```
{
    "SolutionStacks": [
        "64bit Amazon Linux 2015.03 v2.0.0 running Node.js",
        "64bit Amazon Linux 2015.03 v2.0.0 running PHP 5.6",
        "64bit Amazon Linux 2015.03 v2.0.0 running PHP 5.5",
        "64bit Amazon Linux 2015.03 v2.0.0 running PHP 5.4",
        "64bit Amazon Linux 2015.03 v2.0.0 running Python 3.4",
        "64bit Amazon Linux 2015.03 v2.0.0 running Python 2.7",
        "64bit Amazon Linux 2015.03 v2.0.0 running Python",
        "64bit Amazon Linux 2015.03 v2.0.0 running Ruby 2.2 (Puma)",
        "64bit Amazon Linux 2015.03 v2.0.0 running Ruby 2.2 (Passenger Standalone)",
        "64bit Amazon Linux 2015.03 v2.0.0 running Ruby 2.1 (Puma)",
        "64bit Amazon Linux 2015.03 v2.0.0 running Ruby 2.1 (Passenger Standalone)",
        "64bit Amazon Linux 2015.03 v2.0.0 running Ruby 2.0 (Puma)",
        "64bit Amazon Linux 2015.03 v2.0.0 running Ruby 2.0 (Passenger Standalone)",
        "64bit Amazon Linux 2015.03 v2.0.0 running Ruby 1.9.3",
        "64bit Amazon Linux 2015.03 v2.0.0 running Tomcat 8 Java 8",
        "64bit Amazon Linux 2015.03 v2.0.0 running Tomcat 7 Java 7",
        "64bit Amazon Linux 2015.03 v2.0.0 running Tomcat 7 Java 6",
```

```
        "64bit Windows Server Core 2012 R2 running IIS 8.5",
        "64bit Windows Server 2012 R2 running IIS 8.5",
        "64bit Windows Server 2012 running IIS 8",
        "64bit Windows Server 2008 R2 running IIS 7.5",
        "64bit Amazon Linux 2015.03 v2.0.0 running Docker 1.6.2",
        "64bit Amazon Linux 2015.03 v2.0.0 running Multi-container Docker 1.6.2
 (Generic)",
        "64bit Debian jessie v2.0.0 running GlassFish 4.1 Java 8 (Preconfigured -
 Docker)",
        "64bit Debian jessie v2.0.0 running GlassFish 4.0 Java 7 (Preconfigured -
 Docker)",
        "64bit Debian jessie v2.0.0 running Go 1.4 (Preconfigured - Docker)",
        "64bit Debian jessie v2.0.0 running Go 1.3 (Preconfigured - Docker)",
        "64bit Debian jessie v2.0.0 running Python 3.4 (Preconfigured - Docker)",
    ],
    "SolutionStackDetails": [
        {
            "PermittedFileTypes": [
                "zip"
            ],
            "SolutionStackName": "64bit Amazon Linux 2015.03 v2.0.0 running Node.js"
        },
        ...
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListAvailableSolutionStacks를 참조하세요.

rebuild-environment

다음 코드 예시는 rebuild-environment의 사용 방법을 보여 줍니다.

AWS CLI

환경을 재구축하려면

다음 명령은 my-env 환경에서 리소스를 종료하고 다시 생성합니다.

```
aws elasticbeanstalk rebuild-environment --environment-name my-env
```

- API 세부 정보는 AWS CLI 명령 참조의 RebuildEnvironment를 참조하세요.

`request-environment-info`

다음 코드 예시는 `request-environment-info`의 사용 방법을 보여 줍니다.

AWS CLI

테일 로그를 요청하려면

다음 명령은 `my-env` 환경에서 로그를 요청합니다.

```
aws elasticbeanstalk request-environment-info --environment-name my-env --info-type tail
```

로그를 요청한 후 retrieve-environment-info로 해당 위치를 검색합니다.

- API 세부 정보는 AWS CLI 명령 참조의 [RequestEnvironmentInfo](https://docs.aws.amazon.com/goto/aws-cli/elasticbeanstalk-2010-12-01/RequestEnvironmentInfo)를 참조하세요.

`restart-app-server`

다음 코드 예시는 `restart-app-server`의 사용 방법을 보여 줍니다.

AWS CLI

애플리케이션 서버를 다시 시작하려면

다음 명령은 `my-env` 환경의 모든 인스턴스에서 애플리케이션 서버를 다시 시작합니다.

```
aws elasticbeanstalk restart-app-server --environment-name my-env
```

- API 세부 정보는 AWS CLI 명령 참조의 RestartAppServer를 참조하세요.

`retrieve-environment-info`

다음 코드 예시는 `retrieve-environment-info`의 사용 방법을 보여 줍니다.

AWS CLI

테일 로그를 검색하려면

다음 명령은 `my-env` 환경에서 로그에 대한 링크를 검색합니다.

```
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 request-environment-info로 로그를 요청해야 합니다.

- API 세부 정보는 AWS CLI 명령 참조의 RetrieveEnvironmentInfo를 참조하세요.

`swap-environment-cnames`

다음 코드 예시는 `swap-environment-cnames`의 사용 방법을 보여 줍니다.

AWS CLI

환경 CNAMES를 바꾸려면

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
```

- API 세부 정보는 AWS CLI 명령 참조의 SwapEnvironmentCnames를 참조하세요.

`terminate-environment`

다음 코드 예시는 `terminate-environment`의 사용 방법을 보여 줍니다.

AWS CLI

환경을 종료하려면

다음 명령은 `my-env` Elastic Beanstalk 환경을 종료합니다.

```
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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [TerminateEnvironment](https://awscli.amazonaws.com/v2/documentation/api/latest/reference/elasticbeanstalk/terminate-environment.html)를 참조하세요.

`update-application-version`

다음 코드 예시는 `update-application-version`의 사용 방법을 보여 줍니다.

AWS CLI

애플리케이션 버전의 설명을 변경하려면

다음 명령은 `22a0-stage-150819_185942` 애플리케이션 버전에 대한 설명을 업데이트합니다.

```
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"
        }
    }
}
```


- API 세부 정보는 AWS CLI 명령 참조의 UpdateApplicationVersion을 참조하세요.

update-application

다음 코드 예시는 update-application의 사용 방법을 보여 줍니다.

AWS CLI

애플리케이션의 설명을 변경하려면

다음 명령은 my-app 애플리케이션의 설명을 업데이트합니다.

```
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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateApplication을 참조하세요.

`update-configuration-template`

다음 코드 예시는 `update-configuration-template`의 사용 방법을 보여 줍니다.

AWS CLI

구성 템플릿을 업데이트하려면

다음 명령은 저장된 구성 템플릿 `my-template`에서 구성된 CloudWatch 사용자 지정 상태 지표 구성 `ConfigDocument`를 제거합니다.

```
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 안내서에서 Option Values를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateConfigurationTemplate을 참조하세요.

`update-environment`

다음 코드 예시는 `update-environment`의 사용 방법을 보여 줍니다.

AWS CLI

환경을 새 버전으로 업데이트하려면

다음 명령은 'my-env'라는 환경을 환경이 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-env' 환경의 'PARAM1' 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.json은 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 안내서에서 Option Values를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateEnvironment를 참조하세요.

validate-configuration-settings

다음 코드 예시는 `validate-configuration-settings`의 사용 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`은 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 안내서에서 Option Values를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ValidateConfigurationSettings를 참조하세요.

AWS CLI를 사용한 Elastic Load Balancing - 버전 1 예제

다음 코드 예제에서는 Elastic Load Balancing - 버전 1과 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

add-tags

다음 코드 예시는 add-tags의 사용 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"
```

- API 세부 정보는 AWS CLI 명령 참조의 AddTags를 참조하세요.

apply-security-groups-to-load-balancer

다음 코드 예시는 `apply-security-groups-to-load-balancer`의 사용 방법을 보여 줍니다.

AWS CLI

보안 그룹을 VPC의 로드 밸런서와 연결하려면

이 예제에서는 VPC의 지정된 로드 밸런서에 보안 그룹을 연결합니다.

명령:

```
aws elb apply-security-groups-to-load-balancer --load-balancer-name my-load-balancer
 --security-groups sg-fc448899
```

출력:

```
{
  "SecurityGroups": [
      "sg-fc448899"
  ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ApplySecurityGroupsToLoadBalancer를 참조하세요.

attach-load-balancer-to-subnets

다음 코드 예시는 `attach-load-balancer-to-subnets`의 사용 방법을 보여 줍니다.

AWS CLI

로드 밸런서에 서브넷을 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"
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [AttachLoadBalancerToSubnets](https://docs.aws.amazon.com/goto/aws-cli/elasticloadbalancing-2012-06-01/AttachLoadBalancerToSubnets)를 참조하세요.

`configure-health-check`

다음 코드 예시는 `configure-health-check`의 사용 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
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [ConfigureHealthCheck](https://docs.aws.amazon.com/goto/aws-cli/elasticloadbalancing-2012-06-01/ConfigureHealthCheck)를 참조하세요.

`create-app-cookie-stickiness-policy`

다음 코드 예시는 `create-app-cookie-stickiness-policy`의 사용 방법을 보여 줍니다.

AWS CLI

HTTPS 로드 밸런서에 대한 스티키 정책을 생성하려면

이 예제에서는 애플리케이션 생성 쿠키의 스티키 세션 수명을 따르는 스티키 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
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateAppCookieStickinessPolicy를 참조하세요.

`create-lb-cookie-stickiness-policy`

다음 코드 예시는 `create-lb-cookie-stickiness-policy`의 사용 방법을 보여 줍니다.

AWS CLI

HTTPS 로드 밸런서에 대한 기간 기반 스티키 정책을 생성하려면

이 예제에서는 지정된 만료 기간으로 제어되는 스티키 세션 수명이 있는 스티키 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
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateLbCookieStickinessPolicy를 참조하세요.

`create-load-balancer-listeners`

다음 코드 예시는 `create-load-balancer-listeners`의 사용 방법을 보여 줍니다.

AWS CLI

로드 밸런서의 HTTP 리스너를 만들려면

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 밸런서의 HTTPS 리스너를 만들려면

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"
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateLoadBalancerListeners를 참조하세요.

`create-load-balancer-policy`

다음 코드 예시는 `create-load-balancer-policy`의 사용 방법을 보여 줍니다.

AWS CLI

로드 밸런서에서 프록시 프로토콜을 활성화하는 정책을 만들려면

이 예제에서는 지정된 로드 밸런서에 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 만들려면

이 예제에서는 권장 보안 정책을 사용하여 지정된 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=Reference-Security-
Policy,AttributeValue=ELBSecurityPolicy-2015-03
```

사용자 지정 보안 정책을 사용하여 SSL 협상 정책을 만들려면

이 예제에서는 프로토콜과 암호를 활성화하여 사용자 지정 보안 정책을 통해 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 만들려면

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 만들려면

이 예제에서는 퍼블릭 키 정책을 사용하여 백엔드 인스턴스에서 인증할 수 있는 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
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateLoadBalancerPolicy를 참조하세요.

create-load-balancer

다음 코드 예시는 create-load-balancer의 사용 방법을 보여 줍니다.

AWS CLI

HTTP 로드 밸런서를 만들려면

이 예제에서는 VPC에 HTTP 리스너가 있는 로드 밸런서를 생성합니다.

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-Classic에 HTTP 리스너가 있는 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 만들려면

이 예제에서는 VPC에 HTTPS 리스너가 있는 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" --subnets subnet-15aaab61 --security-groups sg-a61988c3
```

출력:

```
{
    "DNSName": "my-load-balancer-1234567890.us-west-2.elb.amazonaws.com"
}
```

이 예제에서는 EC2-Classic에 HTTPS 리스너가 있는 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 만들려면

이 예제에서는 VPC에 HTTP 리스너가 있는 내부 로드 밸런서를 생성합니다.

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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateLoadBalancer를 참조하세요.

delete-load-balancer-listeners

다음 코드 예시는 delete-load-balancer-listeners의 사용 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
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteLoadBalancerListeners를 참조하세요.

delete-load-balancer-policy

다음 코드 예시는 delete-load-balancer-policy의 사용 방법을 보여 줍니다.

AWS CLI

로드 밸런서에서 정책을 삭제하려면

이 예제에서는 지정된 로드 밸런서에서 지정된 정책을 삭제합니다. 이 정책은 어떤 리스너에서도 활성화되어서는 안 됩니다.

명령:

```
aws elb delete-load-balancer-policy --load-balancer-name my-load-balancer --policy-name my-duration-cookie-policy
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteLoadBalancerPolicy를 참조하세요.

delete-load-balancer

다음 코드 예시는 delete-load-balancer의 사용 방법을 보여 줍니다.

AWS CLI

로드 밸런서 삭제

이 예제에서는 지정된 로드 밸런서를 삭제합니다.

명령:

```
aws elb delete-load-balancer --load-balancer-name my-load-balancer
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteLoadBalancer를 참조하세요.

deregister-instances-from-load-balancer

다음 코드 예시는 deregister-instances-from-load-balancer의 사용 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeregisterInstancesFromLoadBalancer를 참조하세요.

describe-account-limits

다음 코드 예시는 describe-account-limits의 사용 방법을 보여 줍니다.

AWS CLI

Classic Load Balancer의 제한을 설명하려면

다음 `describe-account-limits` 예제에서는 AWS 계정의 Classic Load Balancer 제한에 대한 세부 정보를 표시합니다.

```
aws elb describe-account-limits
```

출력:

```
{
    "Limits": [
        {
            "Name": "classic-load-balancers",
            "Max": "20"
        },
        {
            "Name": "classic-listeners",
            "Max": "100"
        },
        {
            "Name": "classic-registered-instances",
            "Max": "1000"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeAccountLimits](#) 섹션을 참조하세요.

`describe-instance-health`

다음 코드 예시는 `describe-instance-health`의 사용 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 응답 예제입니다.

출력:

```
{
  "InstanceStates": [
      {
          "InstanceId": "i-7299c809",
          "ReasonCode": "ELB",
          "State": "OutOfService",
          "Description": "Instance registration is still in progress."
    }
```

```
    ]
}
```

다음은 비정상 인스턴스에 대한 응답 예제입니다.

출력:

```
{
    "InstanceStates": [
        {
            "InstanceId": "i-7299c809",
            "ReasonCode": "Instance",
            "State": "OutOfService",
            "Description": "Instance has failed at least the UnhealthyThreshold number
 of health checks consecutively."
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeInstanceHealth를 참조하세요.

describe-load-balancer-attributes

다음 코드 예시는 describe-load-balancer-attributes의 사용 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
      }
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeLoadBalancerAttributes를 참조하세요.

describe-load-balancer-policies

다음 코드 예시는 describe-load-balancer-policies의 사용 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"
    }
  ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeLoadBalancerPolicies를 참조하세요.

describe-load-balancer-policy-types

다음 코드 예시는 describe-load-balancer-policy-types의 사용 방법을 보여 줍니다.

AWS CLI

Elastic Load Balancing에서 정의한 로드 밸런서 정책 유형을 설명하려면

이 예제에서는 로드 밸런서에 대한 정책 구성을 만드는 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."
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeLoadBalancerPolicyTypes를 참조하세요.

describe-load-balancers

다음 코드 예시는 `describe-load-balancers`의 사용 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 예제는 VPC의 HTTPS 로드 밸런서에 대한 응답입니다.

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"
        }
      }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeLoadBalancers를 참조하세요.

describe-tags

다음 코드 예시는 `describe-tags`의 사용 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"
```

```
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeTags 섹션을 참조하세요.

detach-load-balancer-from-subnets

다음 코드 예시는 detach-load-balancer-from-subnets의 사용 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"
   ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DetachLoadBalancerFromSubnets를 참조하세요.

disable-availability-zones-for-load-balancer

다음 코드 예시는 disable-availability-zones-for-load-balancer의 사용 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"
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DisableAvailabilityZonesForLoadBalancer를 참조하세요.

`enable-availability-zones-for-load-balancer`

다음 코드 예시는 `enable-availability-zones-for-load-balancer`의 사용 방법을 보여 줍니다.

AWS CLI

로드 밸런서의 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"
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 EnableAvailabilityZonesForLoadBalancer를 참조하세요.

modify-load-balancer-attributes

다음 코드 예시는 modify-load-balancer-attributes의 사용 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ModifyLoadBalancerAttributes를 참조하세요.

`register-instances-with-load-balancer`

다음 코드 예시는 `register-instances-with-load-balancer`의 사용 방법을 보여 줍니다.

AWS CLI

로드 밸런서를 사용하여 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"
       }
   ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 RegisterInstancesWithLoadBalancer를 참조하세요.

`remove-tags`

다음 코드 예시는 `remove-tags`의 사용 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
```

- API 세부 정보는 AWS CLI 명령 참조의 RemoveTags를 참조하세요.

set-load-balancer-listener-ssl-certificate

다음 코드 예시는 set-load-balancer-listener-ssl-certificate의 사용 방법을 보여 줍니다.

AWS CLI

HTTPS 로드 밸런서의 SSL 인증서를 업데이트하려면

이 예제에서는 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
```

- API 세부 정보는 AWS CLI 명령 참조의 SetLoadBalancerListenerSslCertificate를 참조하세요.

set-load-balancer-policies-for-backend-server

다음 코드 예시는 set-load-balancer-policies-for-backend-server의 사용 방법을 보여 줍니다.

AWS CLI

백엔드 인스턴스의 포트와 연결된 정책을 교체하려면

이 예제에서는 현재 지정된 포트에 연결된 정책을 교체합니다.

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.

- API 세부 정보는 AWS CLI 명령 참조의 SetLoadBalancerPoliciesForBackendServer를 참조하세요.

`set-load-balancer-policies-of-listener`

다음 코드 예시는 `set-load-balancer-policies-of-listener`의 사용 방법을 보여 줍니다.

AWS CLI

리스너와 연결된 정책을 교체하려면

이 예제에서는 현재 지정된 리스너와 연결된 정책을 교체합니다.

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.

- API 세부 정보는 AWS CLI 명령 참조의 SetLoadBalancerPoliciesOfListener를 참조하세요.

AWS CLI를 사용한 Elastic Load Balancing - 버전 2 예시

다음 코드 예시에서는 Elastic Load Balancing - 버전 2에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`add-listener-certificates`

다음 코드 예시에서는 `add-listener-certificates` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

보안 리스너에 인증서를 추가하려면

이 예시에서는 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
      }
  ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 AddListenerCertificates 섹션을 참조하세요.

add-tags

다음 코드 예시는 add-tags의 사용 방법을 보여 줍니다.

AWS CLI

로드 밸런서에 태그를 추가하려면

다음 add-tags 예시에서는 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"
```

- API 세부 정보는 AWS CLI 명령 참조의 AddTags 섹션을 참조하세요.

create-listener

다음 코드 예시는 create-listener의 사용 방법을 보여 줍니다.

AWS CLI

예 1: HTTP 리스너를 생성하는 방법

다음 create-listener 예시에서는 요청을 지정된 대상 그룹으로 전달하는 지정된 Application Load Balancer의 HTTP 리스너를 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 Balancers 사용 설명서의 자습서: AWS CLI를 Application Load Balancer 생성을 참조하세요.

예 2: HTTPS 리스너를 생성하는 방법

다음 `create-listener` 예시에서는 요청을 지정된 대상 그룹으로 전달하는 지정된 Application Load Balancer의 HTTPS 리스너를 생성합니다. HTTPS 리스너에 대해 SSL 인증서를 지정해야 합니다. AWS Certificate Manager(ACM)를 사용하여 인증서를 생성하고 관리할 수 있습니다. 또는 SSL/TLS 도구를 사용해 인증서를 생성하고, 인증 기관(CA)에서 서명한 인증서를 가져오고, AWS Identity and Access Management(IAM)으로 인증서를 업로드할 수 있습니다.

```
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.

예 1: TCP 리스너를 생성하는 방법

다음 `create-listener` 예시에서는 요청을 지정된 대상 그룹으로 전달하는 지정된 Network Load Balancer의 TCP 리스너를 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 Balancers 사용 설명서의 자습서: AWS CLI를 Network Load Balancer 생성을 참조하세요.

예 4: TLS 리스너를 생성하는 방법

다음 `create-listener` 예시에서는 요청을 지정된 대상 그룹으로 전달하는 지정된 Network Load Balancer의 TLS 리스너를 생성합니다. TLS 리스너에 대해 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 정보는 Network Load Balancer 사용 설명서의 Network Load Balancer를 위한 TLS 리스너를 참조하세요.

예 4: UDP 리스너를 생성하는 방법

다음 `create-listener` 예시에서는 요청을 지정된 대상 그룹으로 전달하는 지정된 Network Load Balancer의 UDP 리스너를 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 Balancers 사용 설명서의 자습서: AWS CLI를 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 Balancers 사용 설명서의 AWS CLI를 사용하여 Gateway Load Balancer 시작하기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [CreateListener](#)를 참조하세요.

`create-load-balancer`

다음 코드 예시는 `create-load-balancer`의 사용 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 Balancers 사용 설명서의 [자습서: AWS CLI를 Application Load Balancer 생성]을 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 Balancers 사용 설명서의 자습서: AWS CLI를 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 Balancers 사용 설명서의 자습서: AWS CLI를 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 Balancers 사용 설명서의 AWS CLI를 사용하여 Gateway Load Balancer 시작하기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateLoadBalancer를 참조하세요.

create-rule

다음 코드 예시는 create-rule의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 경로 조건과 전달 작업을 사용하여 규칙을 만들려면

다음 create-rule 예시에서는 URL에 지정된 패턴이 포함된 경우 요청을 지정된 대상 그룹으로 전달하는 규칙을 만듭니다.

```
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/*"]
        }
    }
]
```

예시 2: 호스트 조건과 고정 응답을 사용하여 규칙을 만들려면

다음 create-rule 예시에서는 호스트 헤더의 호스트 이름이 지정된 호스트 이름과 일치하는 경우 고정 응답을 제공하는 규칙을 만듭니다.

```
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"
        }
    }
]
```

예시 3: 소스 IP 주소 조건, 인증 작업 및 전달 작업을 사용하여 규칙을 만들려면

다음 `create-rule` 예시에서는 소스 IP 주소가 지정된 IP 주소와 일치하는 경우 사용자를 인증하고 인증에 성공하면 지정된 대상 그룹으로 요청을 전달하는 규칙을 만듭니다.

```
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.json`의 콘텐츠

```
[
    {
        "Field": "source-ip",
        "SourceIpConfig": {
            "Values": ["192.0.2.0/24", "198.51.100.10/32"]
        }
    }
]
```

`actions-authenticate.json`의 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": "123456789012345678901234567890",
                "SessionCookieName": "my-cookie",
                "SessionTimeout": 3600,
                "Scope": "email",
                "AuthenticationRequestExtraParams": {
                    "display": "page",
                    "prompt": "login"
                },
                "OnUnauthenticatedRequest": "deny"
            },
            "Order": 1
        },
        {
            "Type": "forward",
            "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
east-1:880185128111:targetgroup/cli-test/642a97ecb0e0f26b",
            "Order": 2
        }
]
```

- API 세부 정보는 AWS CLI 명령 참조의 [CreateRule](#) 섹션을 참조하세요.

`create-target-group`

다음 코드 예시는 `create-target-group`의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: Application Load Balancer 대상 그룹을 만들려면

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.

예시 2: 트래픽을 Application Load Balancer에서 Lambda 함수로 라우팅할 대상 그룹을 만들려면

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 비활성화 됩니다.

```
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 대상으로](#)를 참조하세요.

예시 3: Network Load Balancer 대상 그룹을 만들려면

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 Application Load Balancer 사용 설명서의 [Create a target group](#) 섹션을 참조하세요.

예시 4: Network Load Balancer에서 Application Load Balancer로 트래픽을 라우팅하는 대상 그룹을 만들려면

다음 `create-target-group` 예시에서는 Application Load Balancer를 대상으로 등록(대상 유형은 `alb`임)하는 Network Load Balancer 대상 그룹을 만듭니다.

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 Application Load Balancer 사용 설명서의 Create a target group with an Application Load Balancer as the target 섹션을 참조하세요.

예시 5: Gateway Load Balancer 대상 그룹을 만들려면

다음 `create-target-group` 예시에서는 대상이 인스턴스이고 대상 그룹 프로토콜이 GENEVE인 Gateway Load Balancer 대상 그룹을 만듭니다.

```
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 Create a target group <https://docs.aws.amazon.com/elasticloadbalancing/latest/gateway/create-target-group.html>`__ 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateTargetGroup을 참조하세요.

delete-listener

다음 코드 예시는 delete-listener의 사용 방법을 보여 줍니다.

AWS CLI

리스너를 삭제하려면

다음 delete-listener 예시에서는 지정된 리스너를 삭제합니다.

```
aws elbv2 delete-listener \
    --listener-arn arn:aws:elasticloadbalancing:ua-west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteListener 섹션을 참조하세요.

delete-load-balancer

다음 코드 예시는 delete-load-balancer의 사용 방법을 보여 줍니다.

AWS CLI

로드 밸런서 삭제

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
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteLoadBalancer를 참조하세요.

delete-rule

다음 코드 예시는 `delete-rule`의 사용 방법을 보여 줍니다.

AWS CLI

규칙 삭제

다음 `delete-rule` 예시에서는 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
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteRule을 참조하세요.

delete-target-group

다음 코드 예시는 `delete-target-group`의 사용 방법을 보여 줍니다.

AWS CLI

대상 그룹 삭제

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 사용 설명서의 Delete a load balancer 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteTargetGroup을 참조하세요.

deregister-targets

다음 코드 예시는 deregister-targets의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 대상 그룹에서 대상 등록을 취소하려면

다음 deregister-targets 예시에서는 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
```

예시 2: 포트 재정의를 사용하여 등록된 대상을 등록 취소하려면

다음 deregister-targets 예시에서는 포트 재정의로 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
```

- API 세부 정보는 AWS CLI 명령 참조의 DeregisterTargets 섹션을 참조하세요.

describe-account-limits

다음 코드 예시는 describe-account-limits의 사용 방법을 보여 줍니다.

AWS CLI

Elastic Load Balancing 제한을 설명하려면

다음 describe-account-limits 예시에서는 현재 리전 내 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 정보는 AWS 일반 참조의 할당량을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeAccountLimits 섹션을 참조하세요.

describe-listener-certificates

다음 코드 예시는 describe-listener-certificates의 사용 방법을 보여 줍니다.

AWS CLI

보안 리스너에 대한 인증서를 설명하려면

이 예시에서는 지정된 보안 리스너에 대한 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
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeListenerCertificates 섹션을 참조하세요.

describe-listeners

다음 코드 예시는 describe-listeners의 사용 방법을 보여 줍니다.

AWS CLI

리스너를 설명하려면

이 예시에서는 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 예시에서는 지정된 로드 밸런서의 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeListeners 섹션을 참조하세요.

describe-load-balancer-attributes

다음 코드 예시는 describe-load-balancer-attributes의 사용 방법을 보여 줍니다.

AWS CLI

로드 밸런서 속성을 설명하려면

다음 describe-load-balancer-attributes 예시에서는 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 예시 출력은 Application Load Balancer의 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 예시 출력에는 Network Load Balancer에 대한 속성이 포함되어 있습니다.

```
{
    "Attributes": [
        {
            "Value": "false",
            "Key": "access_logs.s3.enabled"
        },
        {
            "Value": "",
            "Key": "access_logs.s3.bucket"
        },
        {
            "Value": "",
            "Key": "access_logs.s3.prefix"
        },
        {
            "Value": "false",
            "Key": "deletion_protection.enabled"
        },
        {
```

```
            "Value": "false",
            "Key": "load_balancing.cross_zone.enabled"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeLoadBalancerAttributes 섹션을 참조하세요.

describe-load-balancers

다음 코드 예시는 `describe-load-balancers`의 사용 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
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeLoadBalancers를 참조하세요.

describe-rules

다음 코드 예시는 describe-rules의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 규칙을 설명하려면

다음 describe-rules 예시에서는 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
```

예시 2: 리스너에 대한 규칙을 설명하려면

다음 describe-rules 예시에서는 지정된 리스너에 대한 규칙에 대한 세부 정보를 표시합니다. 출력에는 기본 규칙과 사용자가 추가한 다른 규칙이 포함됩니다.

```
aws elbv2 describe-rules \
    --listener-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeRules 섹션을 참조하세요.

describe-ssl-policies

다음 코드 예시는 describe-ssl-policies의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 로드 밸런서 유형별로 SSL 협상에 사용되는 정책을 나열하려면

다음 describe-ssl-policies 예시에서는 Application Load Balancer와의 SSL 협상에 사용할 수 있는 정책의 이름을 표시합니다. 이 예시에서는 --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"
]
```

예시 2: 특정 프로토콜을 지원하는 정책을 나열하려면

다음 describe-ssl-policies 예시에서는 TLS 1.3 프로토콜을 지원하는 정책의 이름을 표시합니다. 이 예시에서는 --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"
]
```

예시 3: 정책의 암호를 표시하려면

다음 describe-ssl-policies 예시에서는 지정된 정책에 대한 암호의 이름을 표시합니다. 이 예시에서는 --query 파라미터를 사용하여 암호 이름만 표시합니다. 목록의 첫 번째 암호는 우선 순위 1을 가지며 나머지 암호는 우선 순위가 정해져 있습니다.

```
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 정보는 Application Load Balancer 사용 설명서의 Security policies 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeSslPolicies 섹션을 참조하세요.

`describe-tags`

다음 코드 예시는 `describe-tags`의 사용 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"
                }
            ]
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeTags 섹션을 참조하세요.

describe-target-group-attributes

다음 코드 예시는 `describe-target-group-attributes`의 사용 방법을 보여 줍니다.

AWS CLI

대상 그룹 속성을 설명하려면

다음 `describe-target-group-attributes` 예시에서는 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 또는 HTTPS이고 대상 유형이 `instance` 또는 `ip`인 경우 출력에는 속성이 포함됩니다.

```
{
    "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"
            }
        ]
}
```

프로토콜이 HTTP 또는 HTTPS이고 대상 유형이 `lambda`인 경우 다음 출력에는 속성이 포함됩니다.

```
{
    "Attributes": [
        {
            "Value": "false",
            "Key": "lambda.multi_value_headers.enabled"
        }
    ]
}
```

다음 출력에는 프로토콜이 TCP, TLS, UDP 또는 TCP_UDP인 경우 속성이 포함됩니다.

```
{
    "Attributes": [
        {
            "Value": "false",
            "Key": "proxy_protocol_v2.enabled"
        },
        {
            "Value": "300",
            "Key": "deregistration_delay.timeout_seconds"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeTargetGroupAttributes 섹션을 참조하세요.

describe-target-groups

다음 코드 예시는 describe-target-groups의 사용 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 예시에서는 --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 Application Load Balancer 사용 설명서의 Target groups 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeTargetGroups를 참조하세요.

describe-target-health

다음 코드 예시는 describe-target-health의 사용 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"
            }
        }
```

```
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeTargetHealth를 참조하세요.

modify-listener

다음 코드 예시는 modify-listener의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 기본 작업을 전달 작업으로 변경하려면

다음 modify-listener 예시에서는 기본 작업을 지정된 리스너에 대한 forward 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=forward,TargetGroupArn=arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-new-targets/2453ed029918f21f
```

출력:

```
{
    "Listeners": [
        {
            "ListenerArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/
f2f7dc8efc522ab2",
            "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188",
            "Protocol": "HTTP",
            "Port": 80,
            "DefaultActions": [
                {
                    "Type": "forward",
                    "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-new-targets/2453ed029918f21f"
                }
            ]
        }
```

```
    ]
}
```

예시 2: 기본 작업을 리디렉션 작업으로 변경하려면

다음 `modify-listener` 예시에서는 기본 작업을 지정된 리스너에 대한 `redirect` 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,
 RedirectConfig='{Protocol=HTTPS,StatusCode=HTTP_302}'
```

출력:

```
{
    "Listeners": [
        {
            "ListenerArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/
f2f7dc8efc522ab2",
            "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188",
            "Protocol": "HTTP",
            "Port": 80,
            "DefaultActions": [
                {
                    "Type": "redirect",
                    "RedirectConfig": {
                        "Protocol": "HTTPS",
                        "Port": "#{port}",
                        "Host": "#{host}",
                        "Path": "/#{path}",
                        "Query": "#{query}",
                        "StatusCode": "HTTP_302",
                    }
                }
            ]
        }
    ]
}
```

예시 3: 서버 인증서를 변경하려면

다음 `modify-listener` 예시에서는 지정된 HTTPS 리스너에 대한 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": [
        {
            "ListenerArn": "arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/0467ef3c8400ae65",
            "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188",
            "Protocol": "HTTPS",
            "Port": 443,
            "DefaultActions": [
                {
                    "Type": "forward",
                    "TargetGroupArn": "arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067"
                }
            ],
            "SslPolicy": "ELBSecurityPolicy-2015-05",
            "Certificates": [
                {
                    "CertificateArn": "arn:aws:iam::123456789012:server-certificate/my-new-server-cert"
                }
            ],
        }
    ]
}
```

자세한 내용은 Application Load Balancer 사용 설명서의 [리스너 규칙](#)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ModifyListener](#) 섹션을 참조하세요.

modify-load-balancer-attributes

다음 코드 예시는 modify-load-balancer-attributes의 사용 방법을 보여 줍니다.

AWS CLI

삭제 보호를 활성화하려면

이 예시에서는 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 예시에서는 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 예시에서는 지정된 로드 밸런서에 대한 액세스 로그를 활성화합니다. S3 버킷은 로드 밸런서와 동일한 리전에 존재해야 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"
      }
  ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ModifyLoadBalancerAttributes를 참조하세요.

`modify-rule`

다음 코드 예시는 `modify-rule`의 사용 방법을 보여 줍니다.

AWS CLI

규칙을 수정하려면

다음 `modify-rule` 예시에서는 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"
                }
            ]
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ModifyRule 섹션을 참조하세요.

`modify-target-group-attributes`

다음 코드 예시는 `modify-target-group-attributes`의 사용 방법을 보여 줍니다.

AWS CLI

등록 취소 지연 제한 시간을 수정하려면

이 예시에서는 지정된 대상 그룹에 대해 등록 취소 지연 시간 제한을 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"
      }
  ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [ModifyTargetGroupAttributes](#) 섹션을 참조하세요.

`modify-target-group`

다음 코드 예시는 `modify-target-group`의 사용 방법을 보여 줍니다.

AWS CLI

대상 그룹에 대한 상태 확인 구성을 수정하려면

다음 `modify-target-group` 예시에서는 지정된 대상 그룹에 대한 대상의 상태를 평가하는 데 사용되는 상태 확인의 구성을 변경합니다. CLI가 쉼표를 구문 분석하는 방식 때문에 `--matcher` 옵션의 범위를 큰따옴표 대신 작은따옴표로 묶어야 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 Application Load Balancer 사용 설명서의 Target groups 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyTargetGroup 섹션을 참조하세요.

register-targets

다음 코드 예시는 register-targets의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 인스턴스 ID별로 대상 그룹에 대상을 등록하려면

다음 register-targets 예시에서는 지정된 인스턴스를 대상 그룹에 등록합니다. 대상 그룹에는 대상 유형이 instance여야 합니다.

```
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
```

예시 2: 포트 재정의를 사용하여 대상 그룹에 대상을 등록하려면

다음 register-targets 예시에서는 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
```

예시 3: IP 주소별로 대상 그룹에 대상을 등록하려면

다음 register-targets 예시에서는 지정된 IP 주소를 대상 그룹에 등록합니다. 대상 그룹에는 대상 유형이 ip여야 합니다.

```
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
```

예시 4: Lambda 함수를 대상으로 등록하려면

다음 `register-targets` 예시에서는 지정된 IP 주소를 대상 그룹에 등록합니다. 대상 그룹에는 대상 유형이 `lambda`여야 합니다. Elastic Load Balancing에 Lambda 함수를 간접적으로 호출할 권한이 있어야 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
```

- API 세부 정보는 AWS CLI 명령 참조의 RegisterTargets 섹션을 참조하세요.

`remove-listener-certificates`

다음 코드 예시는 `remove-listener-certificates`의 사용 방법을 보여 줍니다.

AWS CLI

보안 리스너에서 인증서를 제거하려면

이 예시에서는 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
```

- API 세부 정보는 AWS CLI 명령 참조의 RemoveListenerCertificates 섹션을 참조하세요.

`remove-tags`

다음 코드 예시는 `remove-tags`의 사용 방법을 보여 줍니다.

AWS CLI

로드 밸런서에서 태그를 제거하려면

다음 remove-tags 예시에서는 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
```

- API 세부 정보는 AWS CLI 명령 참조의 RemoveTags 섹션을 참조하세요.

set-ip-address-type

다음 코드 예시는 set-ip-address-type의 사용 방법을 보여 줍니다.

AWS CLI

로드 밸런서의 주소 유형을 설정하려면

이 예시에서는 지정된 로드 밸런서의 주소 유형을 dualstack으로 설정합니다. 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 SetIpAddressType 섹션을 참조하세요.

set-rule-priorities

다음 코드 예시는 set-rule-priorities의 사용 방법을 보여 줍니다.

AWS CLI

규칙 우선 순위를 설정하려면

이 예시에서는 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"
              }
          ]
      }
  ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 SetRulePriorities 섹션을 참조하세요.

set-security-groups

다음 코드 예시는 set-security-groups의 사용 방법을 보여 줍니다.

AWS CLI

보안 그룹을 로드 밸런서와 연결하려면

이 예시에서는 지정된 보안 그룹을 지정된 로드 밸런서와 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"
  ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 SetSecurityGroups 섹션을 참조하세요.

set-subnets

다음 코드 예시는 set-subnets의 사용 방법을 보여 줍니다.

AWS CLI

로드 밸런서에 대해 가용 영역을 활성화하려면

이 예시에서는 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"
      }
  ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 SetSubnets 섹션을 참조하세요.

AWS CLI를 사용한 Elastic Transcoder 예제

다음 코드 예제에서는 Elastic Transcoder에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여 줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`cancel-job`

다음 코드 예시는 `cancel-job`의 사용 방법을 보여 줍니다.

AWS CLI

ElasticTranscoder 작업을 취소하는 방법

이렇게 하면 ElasticTranscoder에 대해 지정된 작업이 취소됩니다.

명령:

```
aws elastictranscoder cancel-job --id 3333333333333-abcde3
```

- API 세부 정보는 AWS CLI 명령 참조의 CancelJob을 참조하세요.

create-job

다음 코드 예시는 `create-job`의 사용 방법을 보여 줍니다.

AWS CLI

ElasticTranscoder 작업을 생성하는 방법

다음 `create-job` 예제에서는 ElasticTranscoder에 대한 작업을 생성합니다.

```
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.json`의 콘텐츠:

```
[{
    "Key":"ETS_example_file.mp4",
    "FrameRate":"auto",
    "Resolution":"auto",
    "AspectRatio":"auto",
    "Interlaced":"auto",
    "Container":"mp4"
}]
```

outputs.json의 콘텐츠:

```
[
    {
        "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"
        }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateJob을 참조하세요.

create-pipeline

다음 코드 예시는 `create-pipeline`의 사용 방법을 보여 줍니다.

AWS CLI

ElasticTranscoder의 파이프라인을 생성하는 방법

다음 `create-pipeline` 예제에서는 ElasticTranscoder에 대한 파이프라인을 생성합니다.

```
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"
        }
    ]
```

```
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreatePipeline을 참조하세요.

create-preset

다음 코드 예시는 create-preset의 사용 방법을 보여 줍니다.

AWS CLI

ElasticTranscoder에 대한 사전 설정을 생성하려면

다음 create-preset 예제에서는 ElasticTranscoder에 대한 사전 설정을 생성합니다.

```
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.json의 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.json`의 콘텐츠:

```
{
    "Codec":"AAC",
    "CodecOptions":{
        "Profile":"AAC-LC"
    },
    "SampleRate":"44100",
    "BitRate":"96",
    "Channels":"2"
}
```

`thumbnails.json`의 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": ""
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreatePreset을 참조하세요.

delete-pipeline

다음 코드 예시는 delete-pipeline의 사용 방법을 보여 줍니다.

AWS CLI

지정된 ElasticTranscoder 파이프라인을 삭제하는 방법

이렇게 하면 지정된 ElasticTranscoder 파이프라인이 삭제됩니다.

명령:

```
aws elastictranscoder delete-pipeline --id 1111111111111-abcde1
```

출력:

```
{
```

```
    "Success":"true"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeletePipeline을 참조하세요.

delete-preset

다음 코드 예시는 delete-preset의 사용 방법을 보여 줍니다.

AWS CLI

지정된 ElasticTranscoder 사전 설정을 삭제하는 방법

이렇게 하면 지정된 ElasticTranscoder 사전 설정이 삭제됩니다.

명령:

```
aws elastictranscoder delete-preset --id 5555555555555-abcde5
```

- API 세부 정보는 AWS CLI 명령 참조의 DeletePreset을 참조하세요.

list-jobs-by-pipeline

다음 코드 예시는 list-jobs-by-pipeline의 사용 방법을 보여 줍니다.

AWS CLI

지정된 파이프라인에서 ElasticTranscoder 작업 목록을 검색하는 방법

이 예제에서는 지정된 파이프라인에서 ElasticTranscoder 작업 목록을 검색합니다.

명령:

```
aws elastictranscoder list-jobs-by-pipeline --pipeline-id 1111111111111-abcde1
```

출력:

```
{
    "Jobs": []
```

```
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListJobsByPipeline을 참조하세요.

list-jobs-by-status

다음 코드 예시는 `list-jobs-by-status`의 사용 방법을 보여 줍니다.

AWS CLI

상태가 Complete인 ElasticTranscoder 작업 목록을 검색하는 방법

이 예제에서는 상태가 Complete인 ElasticTranscoder 작업 목록을 검색합니다.

명령:

```
aws elastictranscoder list-jobs-by-status --status Complete
```

출력:

```
{
   "Jobs": []
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListJobsByStatus를 참조하세요.

list-pipelines

다음 코드 예시는 `list-pipelines`의 사용 방법을 보여 줍니다.

AWS CLI

ElasticTranscoder 파이프라인 목록을 검색하는 방법

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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListPipelines를 참조하세요.

list-presets

다음 코드 예시는 `list-presets`의 사용 방법을 보여 줍니다.

AWS CLI

ElasticTranscoder 사전 설정 목록을 검색하는 방법

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=="
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListPresets를 참조하세요.

read-job

다음 코드 예시는 `read-job`의 사용 방법을 보여 줍니다.

AWS CLI

ElasticTranscoder 작업을 검색하는 방법

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"
    }
}
```


- API 세부 정보는 AWS CLI 명령 참조의 [ReadJob](#)을 참조하세요.

read-pipeline

다음 코드 예시는 read-pipeline의 사용 방법을 보여 줍니다.

AWS CLI

ElasticTranscoder 파이프라인을 검색하는 방법

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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ReadPipeline를 참조하세요.

read-preset

다음 코드 예시는 read-preset의 사용 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ReadPreset을 참조하세요.

update-pipeline-notifications

다음 코드 예시는 update-pipeline-notifications의 사용 방법을 보여 줍니다.

AWS CLI

ElasticTranscoder 파이프라인의 알림을 업데이트하는 방법

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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdatePipelineNotifications를 참조하세요.

`update-pipeline-status`

다음 코드 예시는 `update-pipeline-status`의 사용 방법을 보여 줍니다.

AWS CLI

ElasticTranscoder 파이프라인의 상태를 업데이트하는 방법

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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdatePipelineStatus를 참조하세요.

update-pipeline

다음 코드 예시는 update-pipeline의 사용 방법을 보여 줍니다.

AWS CLI

ElasticTranscoder 파이프라인을 업데이트하는 방법

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.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": "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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdatePipeline을 참조하세요.

AWS CLI를 사용한 ElastiCache 예시

다음 코드 예시는 ElastiCache와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

add-tags-to-resource

다음 코드 예시에서는 add-tags-to-resource 코드를 사용하는 방법을 보여줍니다.

AWS CLI

리소스에 태그 추가

다음 add-tags-to-resource 예시에서는 클러스터 또는 스냅샷 리소스에 최대 10개의 태그 키와 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 사용자 안내서의 비용 할당 태그를 사용하여 비용 모니터링을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AddTagsToResource를 참조하세요.

authorize-cache-security-group-ingress

다음 코드 예시에서는 authorize-cache-security-group-ingress의 사용 방법을 보여줍니다.

AWS CLI

캐시 보안 그룹에 수신 승인

다음 authorize-cache-security-group-ingress 예시에서는 캐시 보안 그룹에 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"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Elasticache 사용자 안내서의 Amazon ElastiCache의 셀프 서비스 업데이트를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AuthorizeCacheSecurityGroupIngress를 참조하세요.

batch-apply-update-action

다음 코드 예시에서는 batch-apply-update-action의 사용 방법을 보여줍니다.

AWS CLI

서비스 업데이트 적용

다음 batch-apply-update-action 예시에서는 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 사용자 안내서의 Amazon ElastiCache의 셀프 서비스 업데이트를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 BatchApplyUpdateAction을 참조하세요.

batch-stop-update-action

다음 코드 예시에서는 `batch-stop-update-action`의 사용 방법을 보여줍니다.

AWS CLI

서비스 업데이트 중지

다음 `batch-stop-update-action` 예시에서는 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 사용자 안내서의 Amazon ElastiCache의 셀프 서비스 업데이트를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 BatchStopUpdateAction을 참조하세요.

copy-snapshot

다음 코드 예시에서는 copy-snapshot의 사용 방법을 보여줍니다.

AWS CLI

스냅샷 복사

다음 copy-snapshot 예시에서는 기존 스냅샷의 사본을 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 사용자 안내서의 백업 내보내기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CopySnapshot을 참조하세요.

`create-cache-cluster`

다음 코드 예시에서는 `create-cache-cluster`의 사용 방법을 보여줍니다.

AWS CLI

캐시 클러스터 생성

다음 `create-cache-cluster` 예시에서는 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 사용자 안내서의 클러스터 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateCacheCluster를 참조합니다.

`create-cache-parameter-group`

다음 코드 예시에서는 `create-cache-parameter-group`의 사용 방법을 보여줍니다.

AWS CLI

캐시 파라미터 그룹 생성

다음 `create-cache-parameter-group` 예시에서는 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 사용자 안내서의 [파라미터 그룹 생성]을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [CreateCacheParameterGroup]을 참조하세요.

`create-cache-subnet-group`

다음 코드 예시에서는 `create-cache-subnet-group`의 사용 방법을 보여줍니다.

AWS CLI

캐시 서브넷 그룹 생성

다음 `create-cache-subnet-group` 예시에서는 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 사용자 안내서의 [캐시 서브넷 그룹 생성]을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateCacheSubnetGroup을 참조하세요.

create-global-replication-group

다음 코드 예시에서는 create-global-replication-group의 사용 방법을 보여줍니다.

AWS CLI

글로벌 복제 그룹 생성

다음 create-global-replication-group 예시에서는 새 글로벌 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 사용자 안내서의 글로벌 데이터 저장소를 사용하여 AWS 리전 간 복제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateGlobalReplicationGroup을 참조하세요.

`create-replication-group`

다음 코드 예시에서는 `create-replication-group`의 사용 방법을 보여줍니다.

AWS CLI

복제 그룹 생성

다음 `create-replication-group` 예시에서는 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 사용자 안내서의 Redis 복제 그룹 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateReplicationGroup을 참조하세요.

`create-snapshot`

다음 코드 예시에서는 `create-snapshot`의 사용 방법을 보여줍니다.

AWS CLI

스냅샷 생성

다음 `create-snapshot` 예시에서는 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 사용자 안내서의 ElastiCache for Redis 백업 및 복원을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateSnapshot을 참조하세요.

`create-user-group`

다음 코드 예시에서는 `create-user-group`의 사용 방법을 보여줍니다.

AWS CLI

사용자 그룹 생성

다음 `create-user-group` 예시에서는 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 사용자 안내서의 역할 기반 액세스 제어(RBAC)를 사용하여 사용자 인증을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateUserGroup을 참조하세요.

create-user

다음 코드 예시에서는 create-user의 사용 방법을 보여줍니다.

AWS CLI

사용자 생성

다음 create-user 예시에서는 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 사용자 안내서의 역할 기반 액세스 제어(RBAC)를 사용하여 사용자 인증을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateUser를 참조하세요.

decrease-node-groups-in-global-replication-group

다음 코드 예시에서는 decrease-node-groups-in-global-replication-group의 사용 방법을 보여줍니다.

AWS CLI

글로벌 복제 그룹의 노드 그룹 수 감소

다음 decrease-node-groups-in-global-replication-group 예시에서는 Redis 엔진을 사용하여 노드 그룹 수를 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 사용자 안내서의 글로벌 데이터 저장소를 사용하여 AWS 리전 간 복제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DecreaseNodeGroupsInGlobalReplicationGroup을 참조하세요.

decrease-replica-count

다음 코드 예시에서는 decrease-replica-count의 사용 방법을 보여줍니다.

AWS CLI

복제본 수 감소

다음 `decrease-replica-count` 예시에서는 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 사용자 안내서의 복제본 수 변경을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DecreaseReplicaCount를 참조하세요.

delete-cache-cluster

다음 코드 예시에서는 `delete-cache-cluster`의 사용 방법을 보여줍니다.

AWS CLI

캐시 클러스터 삭제

다음 `delete-cache-cluster` 예시에서는 이전에 프로비저닝된 지정된 클러스터를 삭제합니다. 명령은 연결된 모든 캐시 노드, 노드 엔드포인트 및 클러스터 자체를 삭제합니다. 이 작업에서 성공적인 응답을 받으면 Amazon ElastiCache는 즉시 클러스터 삭제를 시작합니다. 이 작업은 취소하거나 되돌릴 수 없습니다.

이 작업은 다음에는 유효하지 않습니다.

Redis(클러스터 모드 활성화됨) 클러스터, 복제 그룹의 마지막 읽기 전용 복제본인 클러스터, 다중 AZ 모드가 활성화된 노드 그룹(샤드), Redis(클러스터 모드 활성화됨) 복제 그룹의 클러스터, 사용 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 사용자 안내서의 클러스터 삭제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteCacheCluster를 참조합니다.

delete-cache-parameter-group

다음 코드 예시에서는 delete-cache-parameter-group의 사용 방법을 보여줍니다.

AWS CLI

캐시 파라미터 그룹 삭제

다음 delete-cache-parameter-group 예시에서는 지정된 캐시 파라미터 그룹을 삭제합니다. 캐시 파라미터 그룹이 캐시 클러스터에 연결된 경우 해당 그룹은 삭제할 수 없습니다.

```
aws elasticache delete-cache-parameter-group \
    --cache-parameter-group-name myparamgroup
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Elasticache 사용자 안내서의 파라미터 그룹 삭제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteCacheParameterGroup을 참조하세요.

delete-cache-subnet-group

다음 코드 예시에서는 delete-cache-subnet-group의 사용 방법을 보여줍니다.

AWS CLI

캐시 서브넷 그룹 삭제

다음 delete-cache-subnet-group 예시에서는 지정된 캐시 서브넷 그룹을 삭제합니다. 캐시 서브넷 그룹이 클러스터에 연결된 경우, 해당 그룹은 삭제할 수 없습니다.

```
aws elasticache delete-cache-subnet-group \
    --cache-subnet-group-name "mygroup"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Elasticache 사용자 안내서의 서브넷 그룹 삭제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteCacheSubnetGroup을 참조하세요.

delete-global-replication-group

다음 코드 예시에서는 delete-global-replication-group의 사용 방법을 보여줍니다.

AWS CLI

글로벌 복제 그룹 삭제

다음 delete-global-replication-group 예시에서는 새 글로벌 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 사용자 안내서의 글로벌 데이터 저장소를 사용하여 AWS 리전 간 복제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteGlobalReplicationGroup을 참조하세요.

`delete-replication-group`

다음 코드 예시에서는 `delete-replication-group`의 사용 방법을 보여줍니다.

AWS CLI

복제 그룹 삭제

다음 `delete-replication-group` 예시에서는 기존 복제 그룹을 삭제합니다. 기본적으로 이 작업은 기본 및 모든 읽기 전용 복제본을 포함한 전체 복제 그룹을 삭제합니다. 복제 그룹에 기본 복제본이 하나만 있는 경우 RetainPrimaryCluster=true를 설정하여 기본 복제본을 유지하면서 읽기 전용 복제본만 선택적으로 삭제할 수 있습니다.

이 작업에서 성공적인 응답을 받으면 Amazon ElastiCache는 즉시 선택된 리소스 삭제를 시작합니다. 이 작업은 취소하거나 되돌릴 수 없습니다. 이 작업은 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
     }
 }
```

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteReplicationGroup](#)을 참조하세요.

`delete-snapshot`

다음 코드 예시에서는 `delete-snapshot`의 사용 방법을 보여줍니다.

AWS CLI

스냅샷 삭제

다음 `delete-snapshot` 예시에서는 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 사용자 안내서의 ElastiCache for Redis 백업 및 복원을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteSnapshot을 참조하세요.

delete-user-group

다음 코드 예시에서는 delete-user-group의 사용 방법을 보여줍니다.

AWS CLI

사용자 그룹 삭제

다음 delete-user-group 예시에서는 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 사용자 안내서의 역할 기반 액세스 제어(RBAC)를 사용하여 사용자 인증을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteUserGroup을 참조하세요.

delete-user

다음 코드 예시에서는 delete-user의 사용 방법을 보여줍니다.

AWS CLI

사용자 삭제

다음 delete-user 예시에서는 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 사용자 안내서의 역할 기반 액세스 제어(RBAC)를 사용하여 사용자 인증을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteUser를 참조하세요.

`describe-cache-clusters`

다음 코드 예시에서는 `describe-cache-clusters`의 사용 방법을 보여줍니다.

AWS CLI

캐시 클러스터 설명

다음 `describe-cache-clusters` 예시에서는 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 사용자 안내서의 [클러스터 관리]를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeCacheClusters]를 참조하세요.

describe-cache-engine-versions

다음 코드 예시에서는 describe-cache-engine-versions의 사용 방법을 보여줍니다.

AWS CLI

캐시 엔진 버전 설명

다음 describe-cache-engine-versions 예시에서는 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeCacheEngineVersions를 참조하세요.

`describe-cache-parameter-groups`

다음 코드 예시에서는 `describe-cache-parameter-groups`의 사용 방법을 보여줍니다.

AWS CLI

캐시 파라미터 그룹 설명

다음 `describe-cache-parameter-groups` 예시에서는 캐시 파라미터 그룹의 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 사용자 안내서의 파라미터 그룹을 사용하여 엔진 파라미터 구성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeCacheParameterGroups를 참조하세요.

`describe-cache-parameters`

다음 코드 예시에서는 `describe-cache-parameters`의 사용 방법을 보여줍니다.

AWS CLI

캐시 파라미터 설명

다음 'describe-cache-parameters' 예시에서는 지정된 캐시 파라미터 그룹의 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 사용자 안내서의 파라미터 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeCacheParameters를 참조하세요.

describe-cache-subnet-groups

다음 코드 예시에서는 describe-cache-subnet-groups의 사용 방법을 보여줍니다.

AWS CLI

캐시 서브넷 그룹 설명

다음 describe-cache-subnet-groups 예시에서는 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 사용자 안내서의 서브넷 및 서브넷 그룹 또는 ElastiCache for Memcached 사용자 안내서의 서브넷 및 서브넷 그룹을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeCacheSubnetGroups를 참조하세요.

describe-engine-default-parameters

다음 코드 예시에서는 describe-engine-default-parameters의 사용 방법을 보여줍니다.

AWS CLI

엔진 기본 파라미터 설명

다음 describe-engine-default-parameters 예시에서는 지정된 캐시 엔진의 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"
```

```
            }
        ]
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeEngineDefaultParameters](#)를 참조하세요.

describe-events

다음 코드 예시에서는 `describe-events`의 사용 방법을 보여줍니다.

AWS CLI

복제 그룹의 이벤트 설명

다음 `describe-events` 예시에서는 복제 그룹의 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 사용자 안내서의 [이벤트 모니터링](#)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeEvents](#)를 참조하세요.

`describe-global-replication-groups`

다음 코드 예시에서는 `describe-global-replication-groups`의 사용 방법을 보여줍니다.

AWS CLI

글로벌 복제 그룹 설명

다음 `describe-global-replication-groups` 예시에서는 글로벌 데이터 저장소의 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 사용자 안내서의 [글로벌 데이터 저장소를 사용하여 AWS 리전 간 복제](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeGlobalReplicationGroups](#)를 참조하세요.

`describe-replication-groups`

다음 코드 예시에서는 `describe-replication-groups`의 사용 방법을 보여줍니다.

AWS CLI

복제 그룹 세부 정보의 목록 반환

다음 `describe-replication-groups` 예시에서는 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 사용자 안내서의 클러스터 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeReplicationGroups를 참조하세요.

`describe-reserved-cache-nodes-offerings`

다음 코드 예시에서는 `describe-reserved-cache-nodes-offerings`의 사용 방법을 보여줍니다.

AWS CLI

예약 캐시 노드 오퍼링 설명

다음 `describe-reserved-cache-nodes-offerings` 예시에서는 예약 캐시 노드 옵션의 세부 정보를 반환합니다.

```
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 사용자 안내서의 [예약 노드 오퍼링의 정보 가져오기] 또는 Elasticache Memcached 사용자 안내서의 [예약 노드 오퍼링의 정보 가져오기]를 참조하세요.

• API 세부 정보는 AWS CLI 명령 참조의 [DescribeReservedCacheNodesOfferings]를 참조하세요.

`describe-reserved-cache-nodes`

다음 코드 예시에서는 `describe-reserved-cache-nodes`의 사용 방법을 보여줍니다.

AWS CLI

예약 캐시 노드 설명

다음 `describe-reserved-cache-nodes` 예시에서는 이 계정의 예약 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:xxxxxxxxxxxx52:reserved-instance:mynode"
        }
    ]
```

```
}
```

자세한 내용은 Elasticache 사용자 안내서의 예약 노드로 비용 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeReservedCacheNodes를 참조하세요.

describe-service-updates

다음 코드 예시에서는 describe-service-updates의 사용 방법을 보여줍니다.

AWS CLI

서비스 업데이트 설명

다음 describe-service-updates 예시에서는 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"
            }
        ]
 }
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeServiceUpdates를 참조하세요.

describe-snapshots

다음 코드 예시에서는 describe-snapshots의 사용 방법을 보여줍니다.

AWS CLI

스냅샷 설명

다음 'describe-snapshots' 예시에서는 클러스터 또는 복제 그룹 스냅샷의 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 사용자 안내서의 ElastiCache for Redis 백업 및 복원을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeSnapshots를 참조하세요.

describe-update-actions

다음 코드 예시에서는 describe-update-actions의 사용 방법을 보여줍니다.

AWS CLI

업데이트 작업 설명

다음 `describe-update-actions` 예시에서는 업데이트 작업의 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 사용자 안내서의 Amazon ElastiCache의 셀프 서비스 업데이트를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeUpdateActions를 참조하세요.

describe-user-groups

다음 코드 예시에서는 `describe-user-groups`의 사용 방법을 보여줍니다.

AWS CLI

사용자 그룹 설명

다음 `describe-user-groups` 예시에서는 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 사용자 안내서의 역할 기반 액세스 제어(RBAC)를 사용하여 사용자 인증을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeUserGroups를 참조하세요.

`describe-users`

다음 코드 예시에서는 `describe-users`의 사용 방법을 보여줍니다.

AWS CLI

사용자 설명

다음 `describe-users` 예시에서는 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 사용자 안내서의 역할 기반 액세스 제어(RBAC)를 사용하여 사용자 인증을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeUsers를 참조하세요.

disassociate-global-replication-group

다음 코드 예시에서는 `disassociate-global-replication-group`의 사용 방법을 보여줍니다.

AWS CLI

글로벌 복제 그룹에서 보조 클러스터 연결 해제

다음 `disassociate-global-replication-group` 예시에서는 글로벌 데이터 저장소에서 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 사용자 안내서의 글로벌 데이터 저장소를 사용하여 AWS 리전 간 복제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisassociateGlobalReplicationGroup을 참조하세요.

`increase-node-groups-in-global-replication-group`

다음 코드 예시에서는 `increase-node-groups-in-global-replication-group`의 사용 방법을 보여줍니다.

AWS CLI

글로벌 복제 그룹의 노드 그룹 수 증가

다음 `increase-node-groups-in-global-replication-group` 예시에서는 Redis 엔진을 사용하여 노드 그룹 수를 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 사용자 안내서의 글로벌 데이터 저장소를 사용하여 AWS 리전 간 복제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 IncreaseNodeGroupsInGlobalReplicationGroup을 참조하세요.

`increase-replica-count`

다음 코드 예시에서는 `increase-replica-count`의 사용 방법을 보여줍니다.

AWS CLI

복제본 수 증가

다음 `increase-replica-count` 예시에서는 두 가지 중 하나를 수행합니다. 하나는 Redis(클러스터 모드 비활성화됨) 복제 그룹의 복제본 수를 동적으로 늘리는 것입니다. 다른 하나는 Redis(클러스터 모드 활성화됨) 복제 그룹의 하나 이상의 노드 그룹(샤드)에서 복제본 노드 수를 동적으로 늘리는 것입니다. 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 사용자 안내서의 샤드에서 복제본 수 증가를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 IncreaseReplicaCount를 참조하세요.

`list-allowed-node-type-modifications`

다음 코드 예시에서는 `list-allowed-node-type-modifications`의 사용 방법을 보여줍니다.

AWS CLI

허용된 노드 수정 나열

다음 `list-allowed-node-type-modifications` 예시에서는 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 사용자 안내서의 ElastiCache for Redis 클러스터 스케일링을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListAllowedNodeTypeModifications를 참조하세요.

list-tags-for-resource

다음 코드 예시에서는 `list-tags-for-resource`의 사용 방법을 보여줍니다.

AWS CLI

리소스의 태그 나열

다음 `list-tags-for-resource` 예시에서는 리소스의 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 사용자 안내서의 AWS CLI를 사용하여 태그 나열을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource를 참조하세요.

modify-cache-cluster

다음 코드 예시에서는 modify-cache-cluster의 사용 방법을 보여줍니다.

AWS CLI

캐시 클러스터 수정

다음 modify-cache-cluster 예시에서는 지정된 클러스터의 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 사용자 안내서의 ElastiCache 클러스터 수정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyCacheCluster를 참조하세요.

`modify-cache-parameter-group`

다음 코드 예시에서는 `modify-cache-parameter-group`의 사용 방법을 보여줍니다.

AWS CLI

캐시 파라미터 그룹 수정

다음 `modify-cache-parameter-group` 예시에서는 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 사용자 안내서의 파라미터 그룹 수정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyCacheParameterGroup을 참조하세요.

modify-cache-subnet-group

다음 코드 예시에서는 modify-cache-subnet-group의 사용 방법을 보여줍니다.

AWS CLI

캐시 서브넷 그룹 수정

다음 modify-cache-subnet-group 예시에서는 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 사용자 안내서의 서브넷 그룹 수정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyCacheSubnetGroup을 참조하세요.

modify-global-replication-group

다음 코드 예시에서는 modify-global-replication-group의 사용 방법을 보여줍니다.

AWS CLI

글로벌 복제 그룹 수정

다음 modify-global-replication-group은 Redis 엔진을 사용하여 자동 장애 조치를 비활성화하는 글로벌 복제 그룹의 속성을 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 사용자 안내서의 글로벌 데이터 저장소를 사용하여 AWS 리전 간 복제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyGlobalReplicationGroup을 참조하세요.

modify-replication-group-shard-configuration

다음 코드 예시에서는 modify-replication-group-shard-configuration의 사용 방법을 보여줍니다.

AWS CLI

복제 그룹 샤드 구성 수정

다음 modify-replication-group-shard-configuration은 Redis 엔진을 사용하여 노드 그룹 수를 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 사용자 안내서의 ElastiCache for Redis 클러스터 스케일링을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyReplicationGroupShardConfiguration을 참조하세요.

modify-replication-group

다음 코드 예시에서는 `modify-replication-group`의 사용 방법을 보여줍니다.

AWS CLI

복제 그룹 수정

다음 `modify-replication-group`은 Redis 엔진을 사용하여 다중 AZ를 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 사용자 안내서의 복제 그룹 수정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyReplicationGroup을 참조하세요.

modify-user-group

다음 코드 예시에서는 modify-user-group의 사용 방법을 보여줍니다.

AWS CLI

사용자 그룹 수정

다음 modify-user-group 예시에서는 사용자 그룹에 사용자를 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 사용자 안내서의 역할 기반 액세스 제어(RBAC)를 사용하여 사용자 인증을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyUserGroup을 참조하세요.

modify-user

다음 코드 예시에서는 modify-user의 사용 방법을 보여줍니다.

AWS CLI

사용자 수정

다음 modify-user 예시에서는 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 사용자 안내서의 역할 기반 액세스 제어(RBAC)를 사용하여 사용자 인증을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyUser를 참조하세요.

`purchase-reserved-cache-nodes-offering`

다음 코드 예시에서는 `purchase-reserved-cache-nodes-offering`의 사용 방법을 보여줍니다.

AWS CLI

예약 캐시 노드 오퍼링 구매

다음 `purchase-reserved-cache-nodes-offering` 예시에서는 예약 캐시 노드 오퍼링을 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 사용자 안내서의 [예약 노드 오퍼링의 정보 가져오기](#) 또는 Elasticache Memcached 사용자 안내서의 [예약 노드 오퍼링의 정보 가져오기](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [PurchaseReservedCacheNodesOffering](#)을 참조하세요.

reboot-cache-cluster

다음 코드 예시에서는 `reboot-cache-cluster`의 사용 방법을 보여줍니다.

AWS CLI

캐시 클러스터 재부팅

다음 `reboot-cache-cluster` 예시에서는 프로비저닝된 클러스터 내에서 캐시 노드의 일부 또는 전부를 재부팅합니다. 이 작업은 수정된 캐시 파라미터 그룹을 클러스터에 적용합니다. 재부팅 작업은 최대한 빨리 수행되고 이에 따라 클러스터가 일시적으로 중지됩니다. 재부팅 중에 클러스터 상태는 `REBOOTING`으로 설정됩니다.

```
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 사용자 안내서의 클러스터 재부팅<https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/Clusters.Rebooting.html>을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RebootCacheCluster를 참조하세요.

reset-cache-parameter-group

다음 코드 예시에서는 `reset-cache-parameter-group`의 사용 방법을 보여줍니다.

AWS CLI

캐시 파라미터 그룹 재설정

다음 `reset-cache-parameter-group` 예시에서는 캐시 파라미터 그룹의 파라미터를 엔진 또는 시스템 기본값으로 수정합니다. 파라미터 이름 목록을 제출하여 특정 파라미터를 재설정할 수 있습니다. 캐시 파라미터 그룹 전체를 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ResetCacheParameterGroup을 참조하세요.

start-migration

다음 코드 예시에서는 `start-migration`의 사용 방법을 보여줍니다.

AWS CLI

마이그레이션 시작

다음 `start-migration` 예시에서는 Redis 엔진을 사용하여 Amazon EC2의 자체 호스팅 Redis에서 Amazon ElastiCache로 데이터를 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 사용자 안내서의 ElastiCache로 온라인 마이그레이션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartMigration을 참조하세요.

test-failover

다음 코드 예시에서는 `test-failover`의 사용 방법을 보여줍니다.

AWS CLI

노드 그룹의 장애 조치 테스트

다음 `test-failover` 예시에서는 복제 그룹(콘솔의 클러스터)의 지정된 노드 그룹(콘솔의 샤드)에 대한 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
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 TestFailover를 참조하세요.

AWS CLI를 사용한 MediaStore 예시

다음 코드 예시에서는 MediaStore와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`create-container`

다음 코드 예시에서는 `create-container`의 사용 방법을 보여줍니다.

AWS CLI

컨테이너 생성

다음 create-container 예시에서는 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 AWS Elemental MediaStore 사용자 안내서의 컨테이너 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateContainer를 참조하세요.

delete-container-policy

다음 코드 예시에서는 delete-container-policy의 사용 방법을 보여줍니다.

AWS CLI

컨테이너 정책 삭제

다음 delete-container-policy 예시에서는 지정된 컨테이너에 할당된 정책을 삭제합니다. 정책이 삭제되면 AWS Elemental MediaStore가 컨테이너에 기본 정책을 자동으로 할당합니다.

```
aws mediastore delete-container-policy \
    --container-name LiveEvents
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Elemental MediaStore API 참조의 DeleteContainerPolicy를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteContainerPolicy를 참조하세요.

delete-container

다음 코드 예시에서는 `delete-container`의 사용 방법을 보여줍니다.

AWS CLI

컨테이너 삭제

다음 `delete-container` 예시에서는 지정된 컨테이너를 삭제합니다. 객체가 없는 컨테이너만 삭제할 수 있습니다.

```
aws mediastore delete-container \
    --container-name=ExampleLiveDemo
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Elemental MediaStore 사용자 안내서의 컨테이너 삭제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteContainer를 참조하세요.

delete-cors-policy

다음 코드 예시에서는 `delete-cors-policy`의 사용 방법을 보여줍니다.

AWS CLI

CORS 정책 삭제

다음 `delete-cors-policy` 예시에서는 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 Elemental MediaStore 사용자 안내서의 CORS 정책 삭제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteCorsPolicy를 참조하세요.

delete-lifecycle-policy

다음 코드 예시에서는 delete-lifecycle-policy의 사용 방법을 보여줍니다.

AWS CLI

객체 수명 주기 정책 삭제

다음 delete-lifecycle-policy 예시에서는 지정된 컨테이너에 연결된 객체 수명 주기 정책을 삭제합니다. 변경이 적용되려면 최대 20분이 걸립니다.

```
aws mediastore delete-lifecycle-policy \
    --container-name LiveEvents
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Elemental MediaStore 사용자 안내서의 객체 수명 주기 정책 삭제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteLifecyclePolicy를 참조하세요.

describe-container

다음 코드 예시에서는 describe-container의 사용 방법을 보여줍니다.

AWS CLI

컨테이너 세부 정보 보기

다음 describe-container 예시에서는 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 Elemental MediaStore 사용자 안내서의 컨테이너 세부 정보 보기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeContainer를 참조하세요.

describe-object

다음 코드 예시에서는 describe-object의 사용 방법을 보여줍니다.

AWS CLI

특정 컨테이너에 있는 객체 및 폴더의 목록 보기

다음 describe-object 예시에서는 특정 컨테이너에 저장된 항목(객체 및 폴더)을 표시합니다.

```
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 Elemental MediaStore 사용자 안내서의 객체 세부 정보 보기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeObject를 참조하세요.

`get-container-policy`

다음 코드 예시에서는 `get-container-policy`의 사용 방법을 보여줍니다.

AWS CLI

컨테이너 정책 보기

다음 `get-container-policy` 예시에서는 지정된 컨테이너의 리소스 기반 정책을 표시합니다.

```
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 AWS Elemental MediaStore 사용자 안내서의 컨테이너 정책 보기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetContainerPolicy를 참조하세요.

get-cors-policy

다음 코드 예시에서는 get-cors-policy의 사용 방법을 보여줍니다.

AWS CLI

CORS 정책 보기

다음 get-cors-policy 예시에서는 지정된 컨테이너에 할당된 크로스 오리진 리소스 공유 (CORS) 정책을 표시합니다.

```
aws mediastore get-cors-policy \
    --container-name ExampleContainer \
    --region us-west-2
```

출력:

```
{
    "CorsPolicy": [
        {
            "AllowedMethods": [
                "GET",
                "HEAD"
            ],
            "MaxAgeSeconds": 3000,
            "AllowedOrigins": [
                ""
            ],
            "AllowedHeaders": [
                ""
            ]
        }
    ]
}
```

자세한 내용은 AWS Elemental MediaStore 사용자 안내서의 CORS 정책 보기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetCorsPolicy를 참조하세요.

get-lifecycle-policy

다음 코드 예시에서는 get-lifecycle-policy의 사용 방법을 보여줍니다.

AWS CLI

객체 수명 주기 정책 보기

다음 `get-lifecycle-policy` 예시에서는 지정된 컨테이너에 연결된 객체 수명 주기 정책을 표시합니다.

```
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 AWS Elemental MediaStore 사용자 안내서의 [객체 수명 주기 정책 보기](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetLifecyclePolicy를 참조하세요.

get-object

다음 코드 예시에서는 get-object의 사용 방법을 보여줍니다.

AWS CLI

객체 다운로드

다음 get-object 예시에서는 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 일부 다운로드

다음 get-object 예시에서는 지정된 엔드포인트에 객체의 일부를 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 Elemental MediaStore 사용자 안내서의 객체 다운로드를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetObject를 참조하세요.

list-containers

다음 코드 예시에서는 `list-containers`의 사용 방법을 보여줍니다.

AWS CLI

컨테이너 목록 보기

다음 `list-containers` 예시에서는 계정에 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 Elemental MediaStore 사용자 안내서의 컨테이너 목록 보기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListContainers를 참조하세요.

list-items

다음 코드 예시에서는 `list-items`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 특정 컨테이너에 있는 객체 및 폴더의 목록 보기

다음 `list-items` 예시에서는 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"
        }
    ]
}
```

예시 2: 특정 폴더에 있는 객체 및 폴더의 목록 보기

다음 `list-items` 예시에서는 특정 폴더에 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"
        }
    ]
}
```

자세한 내용은 AWS Elemental MediaStore 사용자 안내서의 [객체 목록 보기]를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ListItems]를 참조하세요.

`list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`의 사용 방법을 보여줍니다.

AWS CLI

컨테이너의 태그 나열

다음 `list-tags-for-resource` 예시에서는 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 AWS Elemental MediaStore API 참조의 ListTagsForResource를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource를 참조하세요.

put-container-policy

다음 코드 예시에서는 put-container-policy의 사용 방법을 보여줍니다.

AWS CLI

컨테이너 정책 편집

다음 put-container-policy 예시에서는 지정된 컨테이너에 다른 정책을 할당합니다. 이 예시에서는 업데이트된 정책이 LiveEventsContainerPolicy.json이라는 파일에 정의되어 있습니다.

```
aws mediastore put-container-policy \
    --container-name LiveEvents \
    --policy file://LiveEventsContainerPolicy.json
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Elemental MediaStore 사용자 안내서의 컨테이너 정책 편집을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutContainerPolicy를 참조하세요.

put-cors-policy

다음 코드 예시에서는 put-cors-policy의 사용 방법을 보여줍니다.

AWS CLI

예시 1: CORS 정책 추가

다음 put-cors-policy 예시에서는 지정된 컨테이너에 크로스 오리진 리소스 공유(CORS) 정책을 추가합니다. CORS 정책의 내용은 corsPolicy.json이라는 파일에 있습니다.

```
aws mediastore put-cors-policy \
    --container-name ExampleContainer \
    --cors-policy file://corsPolicy.json
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Elemental MediaStore 사용자 안내서의 컨테이너에 CORS 정책 추가를 참조하세요.

예시 2: CORS 정책 편집

다음 put-cors-policy 예시에서는 지정된 컨테이너에 할당된 크로스 오리진 리소스 공유(CORS) 정책을 업데이트합니다. 업데이트된 CORS 정책의 내용은 corsPolicy2.json이라는 파일에 있습니다.

자세한 내용은 AWS Elemental MediaStore 사용자 안내서의 CORS 정책 편집을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutCorsPolicy를 참조하세요.

put-lifecycle-policy

다음 코드 예시에서는 put-lifecycle-policy의 사용 방법을 보여줍니다.

AWS CLI

객체 수명 주기 정책 생성

다음 put-lifecycle-policy 예시에서는 객체 수명 주기 정책을 지정된 컨테이너에 연결합니다. 이렇게 하면 서비스가 컨테이너에 객체를 저장할 기간을 지정할 수 있습니다. MediaStore는 LiveEventsLifecyclePolicy.json이라는 이름의 파일에 있는 정책에 표시된 대로 컨테이너의 객체가 만료 날짜에 도달하면 해당 객체를 삭제합니다.

```
aws mediastore put-lifecycle-policy \
    --container-name ExampleContainer \
    --lifecycle-policy file://ExampleLifecyclePolicy.json
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Elemental MediaStore 사용자 안내서의 컨테이너에 객체 수명 주기 정책 추가를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutLifecyclePolicy를 참조하세요.

put-object

다음 코드 예시에서는 put-object의 사용 방법을 보여줍니다.

AWS CLI

객체 업로드

다음 put-object 예시에서는 지정된 컨테이너에 객체를 업로드합니다. 객체가 컨테이너 내에 저장될 폴더 경로를 지정할 수 있습니다. 폴더가 이미 있을 경우 AWS Elemental MediaStore는 폴더에 객체를 저장합니다. 폴더가 없으면 폴더를 만든 후 그 폴더에 객체를 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 Elemental MediaStore 사용자 안내서의 객체 업로드를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutObject를 참조하세요.

start-access-logging

다음 코드 예시에서는 start-access-logging의 사용 방법을 보여줍니다.

AWS CLI

컨테이너에 대한 액세스 로깅 활성화

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 AWS Elemental MediaStore 사용자 안내서의 컨테이너에 대한 액세스 로깅 활성화를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartAccessLogging을 참조하세요.

stop-access-logging

다음 코드 예시에서는 stop-access-logging의 사용 방법을 보여줍니다.

AWS CLI

컨테이너에 대한 액세스 로깅 비활성화

다음 stop-access-logging 예시에서는 지정된 컨테이너에 대한 액세스 로깅을 비활성화합니다.

```
aws mediastore stop-access-logging \
    --container-name LiveEvents
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Elemental MediaStore 사용자 안내서의 컨테이너에 대한 액세스 로깅 비활성화를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StopAccessLogging을 참조하세요.

tag-resource

다음 코드 예시에서는 tag-resource의 사용 방법을 보여줍니다.

AWS CLI

컨테이너에 태그 추가

다음 tag-resource 예시에서는 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 AWS Elemental MediaStore API 참조의 TagResource를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

untag-resource

다음 코드 예시에서는 untag-resource의 사용 방법을 보여줍니다.

AWS CLI

컨테이너에서 태그 제거

다음 untag-resource 예시에서는 지정된 태그 키와 관련 값을 컨테이너에서 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 AWS Elemental MediaStore API 참조의 UntagResource를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

AWS CLI를 사용한 Amazon EMR 예시

다음 코드 예시에서는 Amazon EMR과 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

add-instance-fleet

다음 코드 예시에서는 add-instance-fleet 코드를 사용하는 방법을 보여줍니다.

AWS CLI

클러스터에 태스크 인스턴스 플릿 추가

이 예시에서는 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [AddInstanceFleet](https://docs.aws.amazon.com/goto/aws-cli/elasticmapreduce-2009-03-31/AddInstanceFleet) 섹션을 참조하세요.

add-steps

다음 코드 예시에서는 `add-steps` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

1: 클러스터에 사용자 지정 JAR 단계를 추가하는 방법

명령:

```
aws emr add-steps --cluster-id j-XXXXXXXX --steps
 Type=CUSTOM_JAR,Name=CustomJAR,ActionOnFailure=CONTINUE,Jar=s3://amzn-s3-demo-
bucket/mytest.jar,Args=arg1,arg2,arg3
 Type=CUSTOM_JAR,Name=CustomJAR,ActionOnFailure=CONTINUE,Jar=s3://amzn-s3-demo-
bucket/mytest.jar,MainClass=mymainclass,Args=arg1,arg2,arg3
```

필요한 파라미터:

```
Jar
```

선택적 파라미터:

```
Type, Name, ActionOnFailure, Args
```

출력:

```
{
    "StepIds":[
        "s-XXXXXXXX",
        "s-YYYYYYYY"
    ]
}
```

2. 클러스터에 스트리밍 단계를 추가하는 방법

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://amzn-s3-demo-bucket/wordcount/
output]
```

필요한 파라미터:

```
Type, Args
```

선택적 파라미터:

```
Name, ActionOnFailure
```

JSON과 동등(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://amzn-s3-demo-bucket/wordcount/output"],
    "ActionOnFailure": "CONTINUE",
    "Type": "STREAMING"
  }
]
```

참고: 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 추가하는 방법(JSON만 해당)

JSON(multiplefiles.json):

```
[
  {
     "Name": "JSON Streaming Step",
     "Type": "STREAMING",
     "ActionOnFailure": "CONTINUE",
     "Args": [
         "-files",
         "s3://amzn-s3-demo-bucket/mapper.py,s3://amzn-s3-demo-bucket/reducer.py",
         "-mapper",
         "mapper.py",
         "-reducer",
         "reducer.py",
         "-input",
         "s3://amzn-s3-demo-bucket/input",
         "-output",
         "s3://amzn-s3-demo-bucket/output"]
  }
]
```

명령:

```
aws emr add-steps --cluster-id j-XXXXXXXX  --steps file://./multiplefiles.json
```

필요한 파라미터:

```
Type, Args
```

선택적 파라미터:

```
Name, ActionOnFailure
```

출력:

```
{
    "StepIds":[
        "s-XXXXXXXX",
    ]
```

```
}
```

4. 클러스터에 Hive 단계 추가

명령:

```
aws emr add-steps --cluster-id j-XXXXXXXX --steps Type=HIVE,Name='Hive
 program',ActionOnFailure=CONTINUE,Args=[-f,s3://amzn-s3-demo-bucket/
myhivescript.q,-d,INPUT=s3://amzn-s3-demo-bucket/myhiveinput,-d,OUTPUT=s3://
amzn-s3-demo-bucket/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://amzn-s3-demo-bucket/hive-ads/output/2014-04-18/11-07-32,-d,LIBS=s3://
elasticmapreduce/samples/hive-ads/libs]
```

필요한 파라미터:

```
Type, Args
```

선택적 파라미터:

```
Name, ActionOnFailure
```

출력:

```
{
    "StepIds":[
        "s-XXXXXXXX",
        "s-YYYYYYYY"
    ]
}
```

5. 클러스터에 Pig 단계 추가

명령:

```
aws emr add-steps --cluster-id j-XXXXXXXX --steps Type=PIG,Name='Pig
 program',ActionOnFailure=CONTINUE,Args=[-f,s3://amzn-s3-demo-bucket/
mypigscript.pig,-p,INPUT=s3://amzn-s3-demo-bucket/mypiginput,-p,OUTPUT=s3://
amzn-s3-demo-bucket/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://amzn-s3-demo-bucket/pig-
apache/output,arg1,arg2]
```

필요한 파라미터:

```
Type, Args
```

선택적 파라미터:

```
Name, ActionOnFailure
```

출력:

```
{
    "StepIds":[
        "s-XXXXXXXX",
        "s-YYYYYYYY"
    ]
}
```

6. 클러스터에 Impala 단계를 추가하는 방법

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
```

필요한 파라미터:

```
Type, Args
```

선택적 파라미터:

```
Name, ActionOnFailure
```

출력:

```
{
```

```
    "StepIds":[
        "s-XXXXXXXX",
        "s-YYYYYYYY"
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 AddSteps 섹션을 참조하세요.

`add-tags`

다음 코드 예시에서는 `add-tags` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

1: 클러스터에 태그 추가

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 나열하는 방법

--Command:

```
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"
    }
]
```

- API 세부 정보는 AWS CLI 명령 참조의 AddTags 섹션을 참조하세요.

`create-cluster-examples`

다음 코드 예시에서는 `create-cluster-examples` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

다음 예시의 대부분은 Amazon EMR 서비스 역할과 Amazon EC2 인스턴스 프로파일을 지정했다고 가정합니다. 이렇게 하지 않은 경우 필요한 각 IAM 역할을 지정하거나 클러스터를 생성할 때 `--use-default-roles` 파라미터를 사용해야 합니다. IAM 역할 지정에 대한 자세한 내용은 Amazon EMR 관리 안내서의 AWS 서비스에 대한 Amazon EMR 권한에 대한 IAM 역할 구성을 참조하세요.

예시 1: 클러스터 생성

다음 `create-cluster` 예시에서는 클러스터를 생성합니다.

```
aws emr create-cluster \
    --release-label emr-5.14.0 \
    --instance-type m4.large \
    --instance-count 2
```

이 명령은 출력을 생성하지 않습니다.

예시 2: 기본 ServiceRole 및 InstanceProfile 역할을 사용하여 Amazon EMR 클러스터를 생성하는 방법

다음 `create-cluster` 예시에서는 `--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
```

예시 3: 인스턴스 플릿을 사용하는 Amazon EMR 클러스터 생성

다음 `create-cluster` 예시에서는 `--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
```

예시 4: 기본 역할로 클러스터를 생성하는 방법

다음 `create-cluster` 예시에서는 `--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
```

예시 5: 클러스터를 생성하고 설치할 애플리케이션을 지정하는 방법

다음 `create-cluster` 예시에서는 `--applications` 파라미터를 사용하여 Amazon EMR이 설치하는 애플리케이션을 지정합니다. 이 예시에서는 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
```

예시 6: Spark를 포함하는 클러스터를 생성하는 방법

다음 예시에서는 를 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
```

예시 7: 클러스터 인스턴스에 사용할 사용자 지정 AMI를 지정하는 방법

다음 `create-cluster` 예시에서는 ID가 `ami-a518e6df`인 Amazon Linux AMI를 기반으로 클러스터 인스턴스를 생성합니다.

```
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
```

예시 8: 애플리케이션 구성을 사용자 지정하는 방법

다음 예시에서는 `--configurations` 파라미터를 사용하여 Hadoop에 대한 애플리케이션 사용자 지정이 포함된 JSON 구성 파일을 지정합니다. 자세한 내용은 Amazon EMR 릴리스 안내서의 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 예시에서는 `configurations.json` 파일을 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 예시에서는 `configurations.json` 파일을 Amazon S3의 파일로 참조합니다.

```
aws emr create-cluster \
    --configurations https://s3.amazonaws.com/amzn-s3-demo-bucket/
configurations.json \
    --release-label emr-5.9.0 \
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
 \
    --auto-terminate
```

예시 9: 마스터, 코어 및 태스크 인스턴스 그룹을 사용하여 클러스터를 생성하는 방법

다음 `create-cluster` 예시에서는 `--instance-groups`를 사용하여 마스터, 코어 및 태스크 인스턴스 그룹에 사용할 EC2 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,
```

예시 10: 모든 단계를 완료한 후 클러스터를 종료하도록 지정하는 방법

다음 `create-cluster` 예시에서는 `--auto-terminate`를 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
```

예시 11: Amazon EC2 키 페어, 네트워크 구성 및 보안 그룹과 같은 클러스터 구성 세부 정보를 지정하는 방법

다음 `create-cluster` 예시에서는 이름이 인 Amazon EC2 키 페어`myKey`와 이름이 인 사용자 지정 인스턴스 프로파일을 사용하여 클러스터를 생성합니다`myProfile`. 키 페어는 클러스터 노드에 대한 SSH 연결을 승인하는 데 사용되며, 대부분 마스터 노드입니다. 자세한 내용은 Amazon EMR 관리 안내서의 Use an Amazon EC2 Key Pair for SSH Credentials를 참조하세요.

```
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 예시에서는 Amazon VPC 서브넷에서 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 예시에서는 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 예시에서는 클러스터를 생성하고 Amazon EMR 관리형 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 예시에서는 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 예시에서는 클러스터를 생성하고 EMR 관리형 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 예시는 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 예시에서는 로컬에 저장된 `ec2_attributes.json`라는 JSON 파일을 사용하여 보안 그룹 구성 파라미터를 지정합니다. 참고: 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.json`의 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"]
    }
]
```

예시 12: 디버깅을 활성화하고 로그 URI를 지정하는 방법

다음 `create-cluster` 예시에서는 `--enable-debugging` 파라미터를 사용하여 Amazon EMR 콘솔에서 디버깅 도구를 사용하여 로그 파일을 더 쉽게 볼 수 있습니다. `--log-uri` 파라미터에 `--enable-debugging`이 필요합니다.

```
aws emr create-cluster \
    --enable-debugging \
    --log-uri s3://amzn-s3-demo-bucket/myLog \
    --release-label emr-5.9.0 \
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
 \
    --auto-terminate
```

예시 13: 클러스터를 생성할 때 태그를 추가하는 방법

태그는 클러스터를 식별하고 관리하는 데 도움이 되는 키-값 페어입니다. 다음 `create-cluster` 예시에서는 `--tags` 파라미터를 사용하여 클러스터에 대한 세 개의 태그를 생성합니다. 하나는 키 이름이 name, 값이 `Shirley Rodriguez`인 태그와 키 이름이 age, 값이 29인 태그, 다른 하나는 키 이름이 department, 값이 `Analytics`인 태그를 생성합니다.

```
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 예시에서는 클러스터에 적용된 태그를 나열합니다.

```
aws emr describe-cluster \
    --cluster-id j-XXXXXXYY \
```

```
    --query Cluster.Tags
```

예시 14: 암호화 및 기타 보안 기능을 활성화하는 보안 구성을 사용하는 방법

다음 `create-cluster` 예시에서는 `--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
```

예시 15: 인스턴스 그룹에 대해 구성된 추가 EBS 스토리지 볼륨을 사용하여 클러스터를 생성하는 방법

EBS 볼륨을 추가로 지정할 때는 `VolumeType` 인수가 필요하고 `EbsBlockDeviceConfigs`가 지정된 경우 `SizeInGB`입니다.

다음 `create-cluster` 예시에서는 코어 인스턴스 그룹에서 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
 'InstanceGroupType=CORE,InstanceCount=2,InstanceType=d2.xlarge,EbsConfiguration={EbsOptimiz
{VolumeSpecification={VolumeType=io1,SizeInGB=100,Iops=100},VolumesPerInstance=4}]}'
 \
    --auto-terminate
```

다음 예시에서는 마스터 인스턴스 그룹에서 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
```

예시 16: 자동 조정 정책을 사용하여 클러스터를 생성하는 방법

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 추가하고 제거합니다. 자세한 내용은 Amazon EMR 관리 안내서의 Using Automatic Scaling in Amazon EMR<https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-automatic-scaling.html>`_을 참조하세요.

자동 조정 정책을 연결할 때 `--auto-scaling-role EMR_AutoScaling_DefaultRole`을 사용하여 자동 조정에 대한 기본 역할도 지정해야 합니다.

다음 `create-cluster` 예시에서는 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 예시에서는 JSON 파일, `instancegroupconfig.json`을 사용하여 클러스터의 모든 인스턴스 그룹의 구성을 지정합니다. 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.json`의 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}"
                                    }
                                ]
                            }
                        }
                    }
                ]
            }
        }
]
```

예시 17: 클러스터를 생성할 때 사용자 지정 JAR 단계 추가

다음 `create-cluster` 예시에서는 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 `Type=CUSTOM_JAR`를 사용하여 지정됩니다.

사용자 지정 JAR 단계에는 JAR의 경로와 파일 이름을 지정하는 `Jar=` 파라미터가 필요합니다. 선택적 파라미터는 `Type`, `Name`, `ActionOnFailure`, `Args`, `MainClass`입니다. 지정되지 않은 경우 JAR 파일이 매니페스트 파일의 `Main-Class`를 지정해야 합니다.

```
aws emr create-cluster \
    --steps Type=CUSTOM_JAR,Name=CustomJAR,ActionOnFailure=CONTINUE,Jar=s3://amzn-
s3-demo-bucket/
mytest.jar,Args=arg1,arg2,arg3 Type=CUSTOM_JAR,Name=CustomJAR,ActionOnFailure=CONTINUE,Jar=s
amzn-s3-demo-bucket/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
```

예시 18: 클러스터를 생성할 때 스트리밍 단계 추가

다음 `create-cluster` 예시에서는 모든 단계가 실행된 후 종료되는 스트리밍 단계를 클러스터에 추가합니다. 스트리밍 단계에는 파라미터 `Type` 및 `Args`가 필요합니다. 스트리밍 단계 선택적 파라미터는 `Name` 및 `ActionOnFailure`입니다.

다음 예시에서는 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://amzn-s3-demo-bucket/wordcount/output] \
    --release-label emr-5.3.1 \
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
 \
    --auto-terminate
```

다음 예시에서는 `multiplefiles.json`이라는 로컬로 저장된 JSON 구성 파일을 사용합니다. JSON 구성은 여러 파일을 지정합니다. 한 단계에서 여러 파일을 지정하려면 JSON 구성 파일을 사용하여 단계를 지정해야 합니다. JSON 인수에는 목록의 고유한 항목으로 옵션과 값이 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.json`의 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://amzn-s3-demo-bucket/wordcount/output"
        ],
        "ActionOnFailure": "CONTINUE",
        "Type": "STREAMING"
```

```
    }
]
```

예시 19: 클러스터 생성 시 Hive 단계 추가

다음 예시에서는 클러스터를 생성할 때 Hive 단계를 추가합니다. Hive 단계에는 파라미터 Type 및 Args가 필요합니다. Hive 단계의 선택적 파라미터는 Name 및 ActionOnFailure입니다.

```
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://amzn-s3-demo-bucket/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
```

예시 20: 클러스터를 생성할 때 피그 단계를 추가하는 방법

다음 예시에서는 클러스터를 생성할 때 Pig 단계를 추가합니다. Pig 단계 필수 파라미터는 Type 및 Args입니다. Pig 단계 선택적 파라미터는 Name 및 ActionOnFailure입니다.

```
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://amzn-s3-demo-bucket/pig-apache/output] \
    --applications Name=Pig \
    --release-label emr-5.3.1 \
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
```

예시 21: 부트스트랩 작업 추가

다음 create-cluster 예시에서는 Amazon S3에 저장된 스크립트로 정의된 두 부트스트랩 작업을 실행합니다.

```
aws emr create-cluster \
```

```
    --bootstrap-actions Path=s3://amzn-s3-demo-bucket/
myscript1,Name=BootstrapAction1,Args=[arg1,arg2] Path=s3://amzn-s3-demo-bucket/
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
```

예시 22: EMRFS 일관된 보기를 활성화하고 RetryCount 및 RetryPeriod 설정을 사용자 지정하는 방법

다음 `create-cluster` 예시에서는 EMRFS 일관성 있는 보기에 대한 재시도 횟수 및 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 예시에서는 `emrfsconfig.json`이라는 로컬로 저장된 JSON 구성 파일을 사용하여 이전 예시와 동일한 EMRFS 구성을 지정합니다.

```
aws emr create-cluster \
    --instance-type m4.large \
    --release-label emr-5.9.0 \
    --emrfs file://emrfsconfig.json
```

`emrfsconfig.json`의 콘텐츠:

```
{
    "Consistent": true,
    "RetryCount": 6,
    "RetryPeriod": 30
}
```

예시 23: Kerberos가 구성된 클러스터 생성

다음 `create-cluster` 예시에서는 Kerberos가 활성화된 보안 구성을 사용하여 클러스터를 생성하고 `--kerberos-attributes`를 사용하여 클러스터에 대한 Kerberos 파라미터를 설정합니다.

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 `kerberos_attributes.json`이라는 로컬로 저장된 JSON 파일을 참조합니다. 이 예시에서는 명령을 실행하는 디렉터리에 파일이 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` 예시에서는 `--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` 예시에서는 '--log-encryption-kms-key-id'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://amzn-s3-demo-bucket/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` 예시에서는 배치 SPREAD 전략을 사용하여 EC2 배치 그룹 내의 고가용성(HA) 클러스터에 마스터 노드를 배치하기 위해 "--placement-group-configs" 구성을 사용하는 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=3,InstanceType=m4.largeInstanceGroupType=CORE,
 \
    --placement-group-configs InstanceRole=MASTER
```

다음 `create-cluster` 예시에서는 '--auto-termination-policy' 구성을 사용하여 클러스터에 대한 자동 유휴 종료 임계값을 배치하는 Amazon EMR 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` 예시에서는 '-os-release-label'을 사용하여 클러스터 시작을 위한 Amazon Linux 릴리스를 정의하는 Amazon EMR 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
```

예시 24: EBS 루트 볼륨 속성 지정: EMR 릴리스 6.15.0 이상으로 생성된 클러스터 인스턴스의 크기, iops 및 처리량

다음 `create-cluster` 예시에서는 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
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateClusterExamples 섹션을 참조하세요.

`create-default-roles`

다음 코드 예시에서는 `create-default-roles` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

1: EC2에 대한 기본 IAM 역할을 생성하는 방법

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"
        }
    }
]
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateDefaultRoles 섹션을 참조하세요.

`create-security-configuration`

다음 코드 예시에서는 `create-security-configuration` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

1: 인증서 공급자의 경우 PEM으로 전송 중 암호화를 활성화하고, SSSE-S3S3로 저장 중 암호화를 활성화하고, 로컬 디스크 키 공급자의 경우 AWS-KMS로 보안 구성을 생성하는 방법

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"
}
```

JSON과 동등(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 생성하는 방법

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"
}
```

JSON과 동등(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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateSecurityConfiguration 섹션을 참조하세요.

`delete-security-configuration`

다음 코드 예시에서는 `delete-security-configuration` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

현재 리전에서 보안 구성을 삭제하는 방법

명령:

```
aws emr delete-security-configuration --name MySecurityConfig
```

출력:

```
None
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteSecurityConfiguration 섹션을 참조하세요.

`describe-cluster`

다음 코드 예시에서는 `describe-cluster` 코드를 사용하는 방법을 보여줍니다.

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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeCluster를 참조하세요.

`describe-step`

다음 코드 예시에서는 `describe-step` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

다음 명령은 클러스터 ID가 `j-3SD91U2E1L2QX`인 클러스터에서 단계 ID가 `s-3LZC0QUT43AM`인 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeStep을 참조하세요.

get

다음 코드 예시에서는 get 코드를 사용하는 방법을 보여줍니다.

AWS CLI

다음은 클러스터 ID가 j-3SD91U2E1L2QX인 클러스터의 마스터 인스턴스에서 hadoop-examples.jar 아카이브를 다운로드합니다.

```
aws emr get --cluster-id j-3SD91U2E1L2QX --key-pair-file ~/.ssh/mykey.pem --src /
home/hadoop-examples.jar --dest ~
```

- API 세부 정보는 AWS CLI 명령 참조의 Get 섹션을 참조하세요.

list-clusters

다음 코드 예시에서는 list-clusters 코드를 사용하는 방법을 보여줍니다.

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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListClusters를 참조하세요.

`list-instance-fleets`

다음 코드 예시에서는 `list-instance-fleets` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

클러스터 내 인스턴스 플릿의 구성 세부 정보 가져오기

이 예시에서는 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [ListInstanceFleets](#) 섹션을 참조하세요.

list-instances

다음 코드 예시에서는 list-instances 코드를 사용하는 방법을 보여줍니다.

AWS CLI

다음 명령은 클러스터 ID가 j-3C6XNQ39VR9WL인 클러스터의 모든 인스턴스를 나열합니다.

```
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"
            }
        ]
    }
```

- API 세부 정보는 AWS CLI 명령 참조의 ListInstances 섹션을 참조하세요.

list-security-configurations

다음 코드 예시에서는 `list-security-configurations` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

현재 리전의 보안 구성을 나열하는 방법

명령:

```
aws emr list-security-configurations
```

출력:

```
{
    "SecurityConfigurations": [
        {
            "CreationDateTime": 1473889697.417,
            "Name": "MySecurityConfig-1"
        },
        {
            "CreationDateTime": 1473889697.417,
            "Name": "MySecurityConfig-2"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListSecurityConfigurations 섹션을 참조하세요.

list-steps

다음 코드 예시에서는 list-steps 코드를 사용하는 방법을 보여줍니다.

AWS CLI

다음 명령은 클러스터 ID가 j-3SD91U2E1L2QX인 클러스터의 모든 단계를 나열합니다.

```
aws emr list-steps --cluster-id j-3SD91U2E1L2QX
```

- API 세부 정보는 AWS CLI 명령 참조의 ListSteps를 참조하세요.

modify-cluster-attributes

다음 코드 예시에서는 modify-cluster-attributes 코드를 사용하는 방법을 보여줍니다.

AWS CLI

다음 명령은 ID가 j-301CDNY0J5XM4인 EMR 클러스터의 가시성을 모든 사용자에게 설정합니다.

```
aws emr modify-cluster-attributes --cluster-id j-301CDNY0J5XM4 --visible-to-all-
users
```

- API 세부 정보는 AWS CLI 명령 참조의 ModifyClusterAttributes 섹션을 참조하세요.

modify-instance-fleet

다음 코드 예시에서는 modify-instance-fleet 코드를 사용하는 방법을 보여줍니다.

AWS CLI

인스턴스 플릿의 대상 캐패시터를 변경하는 방법

이 예시에서는 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
```

- API 세부 정보는 AWS CLI 명령 참조의 ModifyInstanceFleet 섹션을 참조하세요.

put

다음 코드 예시에서는 put 코드를 사용하는 방법을 보여줍니다.

AWS CLI

다음 명령은 클러스터 ID가 j-3SD91U2E1L2QX인 클러스터의 마스터 인스턴스에 healthcheck.sh라는 파일을 업로드합니다.

```
aws emr put --cluster-id j-3SD91U2E1L2QX --key-pair-file ~/.ssh/mykey.pem --src ~/
scripts/healthcheck.sh --dest /home/hadoop/bin/healthcheck.sh
```

- API 세부 정보는 AWS CLI 명령 참조의 Put 섹션을 참조하세요.

remove-tags

다음 코드 예시에서는 remove-tags 코드를 사용하는 방법을 보여줍니다.

AWS CLI

다음 명령은 클러스터 ID가 j-3SD91U2E1L2QX인 클러스터에서 키가 prod인 태그를 제거합니다.

```
aws emr remove-tags --resource-id j-3SD91U2E1L2QX --tag-keys prod
```

- API 세부 정보는 AWS CLI 명령 참조의 RemoveTags 섹션을 참조하세요.

schedule-hbase-backup

다음 코드 예시에서는 schedule-hbase-backup 코드를 사용하는 방법을 보여줍니다.

AWS CLI

참고: 이 명령은 AMI 버전 2.x 및 3.x의 HBase에서만 사용할 수 있습니다.

1: 전체 HBase 백업을 예약하는 방법 >>>>>>> 06ab6d6e13564b5733d75abaf3b599f93cf39a23

명령:

```
aws emr schedule-hbase-backup --cluster-id j-XXXXXXYY --type full --dir
s3://amzn-s3-demo-bucket/backup --interval 10 --unit hours --start-time
```

```
2014-04-21T05:26:10Z --consistent
```

출력:

```
None
```

2. 증분 HBase 백업을 예약하는 방법

명령:

```
aws emr schedule-hbase-backup --cluster-id j-XXXXXXYY --type incremental
 --dir s3://amzn-s3-demo-bucket/backup --interval 30 --unit minutes --start-time
2014-04-21T05:26:10Z --consistent
```

출력:

```
None
```

- API 세부 정보는 AWS CLI 명령 참조의 ScheduleHbaseBackup 섹션을 참조하세요.

socks

다음 코드 예시에서는 socks 코드를 사용하는 방법을 보여줍니다.

AWS CLI

다음 명령은 클러스터 ID가 j-3SD91U2E1L2QX인 클러스터의 마스터 인스턴스와 양말 연결을 엽니다.

```
aws emr socks --cluster-id j-3SD91U2E1L2QX --key-pair-file ~/.ssh/mykey.pem
```

키 페어 파일 옵션은 프라이빗 키 파일의 로컬 경로를 가져옵니다.

- API 세부 정보는 AWS CLI 명령 참조의 Socks 섹션을 참조하세요.

ssh

다음 코드 예시에서는 ssh 코드를 사용하는 방법을 보여줍니다.

AWS CLI

다음 명령은 클러스터 ID가 `j-3SD91U2E1L2QX`인 클러스터의 마스터 인스턴스와 ssh 연결을 엽니다.

```
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 세부 정보는 AWS CLI 명령 참조의 Ssh 섹션을 참조하세요.

AWS CLI를 사용한 Amazon EMR on EKS 예제

다음 코드 예제에서는 Amazon EMR on EKS에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여 줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`update-role-trust-policy`

다음 코드 예시는 `update-role-trust-policy`의 사용 방법을 보여 줍니다.

AWS CLI

Amazon EMR on EKS와 함께 사용할 IAM 역할의 신뢰 정책을 업데이트하는 방법

이 예제 명령은 example_iam_role이라는 역할의 신뢰 정책을 example_cluster라는 EKS 클러스터의 example_namespace 네임스페이스가 있는 Amazon EMR on EKS에서 사용할 수 있도록 업데이트합니다.

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.
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateRoleTrustPolicy를 참조하세요.

AWS CLI를 사용한 EventBridge 예제

다음 코드 예제에서는 EventBridge에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여 줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`delete-rule`

다음 코드 예시는 `delete-rule`의 사용 방법을 보여 줍니다.

AWS CLI

CloudWatch Events 규칙을 삭제하는 방법

이 예시에서는 EC2InstanceStateChanges라는 규칙을 삭제합니다.

```
aws events delete-rule --name "EC2InstanceStateChanges"
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteRule을 참조하세요.

`describe-rule`

다음 코드 예시는 `describe-rule`의 사용 방법을 보여 줍니다.

AWS CLI

CloudWatch 이벤트 규칙에 대한 정보를 표시하는 방법

이 예시에서는 DailyLambdaFunction이라는 규칙에 대한 정보를 표시합니다.

```
aws events describe-rule --name "DailyLambdaFunction"
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeRule을 참조하세요.

disable-rule

다음 코드 예시는 disable-rule의 사용 방법을 보여 줍니다.

AWS CLI

CloudWatch 이벤트 규칙을 비활성화하는 방법

이 예시에서는 DailyLambdaFunction라는 규칙을 비활성화합니다. 규칙이 삭제되지는 않습니다.

```
aws events disable-rule --name "DailyLambdaFunction"
```

- API 세부 정보는 AWS CLI 명령 참조의 DisableRule을 참조하세요.

enable-rule

다음 코드 예시는 enable-rule의 사용 방법을 보여 줍니다.

AWS CLI

CloudWatch 이벤트 규칙을 활성화하는 방법

이 예시에서는 이전에 비활성화되었던 DailyLambdaFunction이라는 규칙을 활성화합니다.

```
aws events enable-rule --name "DailyLambdaFunction"
```

- API 세부 정보는 AWS CLI 명령 참조의 EnableRule을 참조하세요.

list-rule-names-by-target

다음 코드 예시는 list-rule-names-by-target의 사용 방법을 보여 줍니다.

AWS CLI

지정된 대상이 있는 모든 규칙을 표시하는 방법

이 예시에서는 이름이 'MyFunctionName'인 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"
```

- API 세부 정보는 AWS CLI 명령 참조의 ListRuleNamesByTarget을 참조하세요.

list-rules

다음 코드 예시는 `list-rules`의 사용 방법을 보여 줍니다.

AWS CLI

모든 CloudWatch 이벤트 규칙 목록을 표시하는 방법

이 예시에서는 해당 리전 내 모든 CloudWatch 이벤트 규칙을 표시합니다.

```
aws events list-rules
```

특정 문자열로 시작하는 CloudWatch 이벤트 규칙 목록을 표시하는 방법

이 예시에서는 이름이 'Daily'로 시작하는 해당 리전 내 모든 CloudWatch Events 규칙을 표시합니다.

```
aws events list-rules --name-prefix "Daily"
```

- API 세부 정보는 AWS CLI 명령 참조의 ListRules를 참조하세요.

list-targets-by-rule

다음 코드 예시는 `list-targets-by-rule`의 사용 방법을 보여 줍니다.

AWS CLI

CloudWatch 이벤트 규칙의 모든 대상을 표시하는 방법

이 예시에서는 DailyLambdaFunction이라는 규칙의 모든 대상을 표시합니다.

```
aws events list-targets-by-rule --rule  "DailyLambdaFunction"
```

- API 세부 정보는 AWS CLI 명령 참조의 ListTargetsByRule을 참조하세요.

put-events

다음 코드 예시는 `put-events`의 사용 방법을 보여 줍니다.

AWS CLI

CloudWatch 이벤트에 사용자 지정 이벤트를 전송하는 방법

이 예시에서는 CloudWatch 이벤트에 사용자 지정 이벤트를 전송합니다. 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"
   }
]
```

- API 세부 정보는 AWS CLI 명령 참조의 [PutEvents](https://docs.aws.amazon.com/goto/aws-cli/eventbridge-2015-10-07/PutEvents)를 참조하세요.

put-rule

다음 코드 예시는 `put-rule`의 사용 방법을 보여 줍니다.

AWS CLI

CloudWatch Events 규칙을 생성하는 방법

이 예시에서는 매일 오전 09:00(UTC)에 트리거되는 규칙을 생성합니다. 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 예시에서는 리전 내 EC2 인스턴스가 상태가 변경될 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 예시에서는 리전 내 EC2 인스턴스가 정지 또는 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"
```

- API 세부 정보는 AWS CLI 명령 참조의 [PutRule](https://docs.aws.amazon.com/goto/aws-cli/eventbridge-2015-10-07/PutRule)을 참조하세요.

put-targets

다음 코드 예시는 `put-targets`의 사용 방법을 보여 줍니다.

AWS CLI

CloudWatch 이벤트 규칙에 대상을 추가하는 방법

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"
```

- API 세부 정보는 AWS CLI 명령 참조의 PutTargets를 참조하세요

`remove-targets`

다음 코드 예시는 `remove-targets`의 사용 방법을 보여 줍니다.

AWS CLI

이벤트 대상을 제거하는 방법

이 예시에서는 MyStream1이라는 이름의 Amazon Kinesis 스트림을 DailyLambdaFunction 규칙의 대상에서 제거합니다. DailyLambdaFunction을 생성할 때 이 스트림은 ID가 Target1인 대상으로 설정되었습니다.

```
aws events remove-targets --rule "DailyLambdaFunction" --ids "Target1"
```

- API 세부 정보는 AWS CLI 명령 참조의 RemoveTargets를 참조하세요.

`test-event-pattern`

다음 코드 예시는 `test-event-pattern`의 사용 방법을 보여 줍니다.

AWS CLI

이벤트 패턴이 지정된 이벤트와 일치하는지 확인하는 방법

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\"}"
```

- API 세부 정보는 AWS CLI 명령 참조의 TestEventPattern을 참조하세요.

AWS CLI를 사용한 EventBridge Pipes 예제

다음 코드 예제에서는 EventBridge Pipes에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여 줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`create-pipe`

다음 코드 예시는 `create-pipe`의 사용 방법을 보여 줍니다.

AWS CLI

파이프를 생성하려면

다음 `create-pipe` 예제에서는 SQS를 소스로, CloudWatch Log Group을 파이프의 대상으로 사용하여 `Demo_Pipe`라는 파이프를 생성합니다.

```
aws pipes create-pipe \
    --name Demo_Pipe \
    --desired-state RUNNING \
    --role-arn arn:aws:iam::123456789012:role/service-role/
Amazon_EventBridge_Pipe_Demo_Pipe_28b3aa4f \
    --source arn:aws:sqs:us-east-1:123456789012:Demo_Queue \
    --target arn:aws:logs:us-east-1:123456789012:log-group:/aws/pipes/Demo_LogGroup
```

출력:

```
{
    "Arn": "arn:aws:pipes:us-east-1:123456789012:pipe/Demo_Pipe",
    "Name": "Demo_Pipe",
    "DesiredState": "RUNNING",
    "CurrentState": "CREATING",
    "CreationTime": "2024-10-08T12:33:59-05:00",
    "LastModifiedTime": "2024-10-08T12:33:59.684839-05:00"
}
```

자세한 내용은 Amazon EventBridge 사용 설명서의 Amazon EventBridge Pipes 개념을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreatePipe를 참조하세요.

delete-pipe

다음 코드 예시는 delete-pipe의 사용 방법을 보여 줍니다.

AWS CLI

기존 파이프를 삭제하려면

다음 delete-pipe 예제에서는 지정된 계정에서 Demo_Pipe라는 파이프를 삭제합니다.

```
aws pipes delete-pipe \
    --name Demo_Pipe
```

출력:

```
{
```

```
    "Arn": "arn:aws:pipes:us-east-1:123456789012:pipe/Demo_Pipe",
    "Name": "Demo_Pipe",
    "DesiredState": "STOPPED",
    "CurrentState": "DELETING",
    "CreationTime": "2024-10-08T09:29:10-05:00",
    "LastModifiedTime": "2024-10-08T11:57:22-05:00"
}
```

자세한 내용은 Amazon EventBridge 사용 설명서의 Amazon EventBridge Pipes 개념을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeletePipe를 참조하세요.

describe-pipe

다음 코드 예시는 describe-pipe의 사용 방법을 보여 줍니다.

AWS CLI

파이프에 대한 정보를 검색하려면

다음 describe-pipe 예제는 지정된 계정에서 Demo_Pipe라는 파이프에 대한 정보를 표시합니다.

```
aws pipes describe-pipe \
    --name Demo_Pipe
```

출력:

```
{
    "Arn": "arn:aws:pipes:us-east-1:123456789012:pipe/Demo_Pipe",
    "Name": "Demo_Pipe",
    "DesiredState": "RUNNING",
    "CurrentState": "RUNNING",
    "StateReason": "User initiated",
    "Source": "arn:aws:sqs:us-east-1:123456789012:Demo_Queue",
    "SourceParameters": {
        "SqsQueueParameters": {
            "BatchSize": 1
        }
    },
    "EnrichmentParameters": {},
```

```
    "Target": "arn:aws:logs:us-east-1:123456789012:log-group:/aws/pipes/
Demo_LogGroup",
    "TargetParameters": {},
    "RoleArn": "arn:aws:iam::123456789012:role/service-role/
Amazon_EventBridge_Pipe_Demo_Pipe_28b3aa4f",
    "Tags": {},
    "CreationTime": "2024-10-08T09:29:10-05:00",
    "LastModifiedTime": "2024-10-08T10:23:47-05:00",
    "LogConfiguration": {
        "CloudwatchLogsLogDestination": {
            "LogGroupArn": "arn:aws:logs:us-east-1:123456789012:log-group:/aws/
vendedlogs/pipes/Demo_Pipe"
        },
        "Level": "ERROR"
    }
}
```

자세한 내용은 Amazon EventBridge 사용 설명서의 Amazon EventBridge Pipes 개념을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribePipe를 참조하세요.

`list-pipes`

다음 코드 예시는 `list-pipes`의 사용 방법을 보여 줍니다.

AWS CLI

파이프 목록을 검색하려면

다음 `list-pipes` 예제에서는 지정된 계정의 모든 파이프를 표시합니다.

```
aws pipes list-pipes
```

출력:

```
{
    "Pipes": [
        {
            "Name": "Demo_Pipe",
            "Arn": "arn:aws:pipes:us-east-1:123456789012:pipe/Demo_Pipe",
            "DesiredState": "RUNNING",
            "CurrentState": "RUNNING",
```

```
            "StateReason": "User initiated",
            "CreationTime": "2024-10-08T09:29:10-05:00",
            "LastModifiedTime": "2024-10-08T10:23:47-05:00",
            "Source": "arn:aws:sqs:us-east-1:123456789012:Demo_Queue",
            "Target": "arn:aws:logs:us-east-1:123456789012:log-group:/aws/pipes/
Demo_LogGroup"
        }
    ]
}
```

자세한 내용은 Amazon EventBridge 사용 설명서의 Amazon EventBridge Pipes 개념을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListPipes 섹션을 참조하세요.

`list-tags-for-resource`

다음 코드 예시는 `list-tags-for-resource`의 사용 방법을 보여 줍니다.

AWS CLI

기존 파이프와 연결된 태그를 나열하려면

다음 `list-tags-for-resource` 예제에서는 지정된 계정에서 `Demo_Pipe`라는 파이프와 연결된 모든 태그를 나열합니다.

```
aws pipes list-tags-for-resource \
    --resource-arn arn:aws:pipes:us-east-1:123456789012:pipe/Demo_Pipe
```

출력:

```
{
    "tags": {
        "stack": "Production",
        "team": "DevOps"
    }
}
```

자세한 내용은 Amazon EventBridge 사용 설명서의 Amazon EventBridge Pipes 개념을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource를 참조하세요.

start-pipe

다음 코드 예시는 start-pipe의 사용 방법을 보여 줍니다.

AWS CLI

기존 파이프를 시작하려면

다음 start-pipe 예제에서는 지정된 계정에서 Demo_Pipe라는 파이프를 시작합니다.

```
aws pipes start-pipe \
    --name Demo_Pipe
```

출력:

```
{
    "Arn": "arn:aws:pipes:us-east-1:123456789012:pipe/Demo_Pipe",
    "Name": "Demo_Pipe",
    "DesiredState": "RUNNING",
    "CurrentState": "STARTING",
    "CreationTime": "2024-10-08T09:29:10-05:00",
    "LastModifiedTime": "2024-10-08T10:17:24-05:00"
}
```

자세한 내용은 Amazon EventBridge 사용 설명서에서 Amazon EventBridge 파이프 시작 또는 중지를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartPipe 섹션을 참조하세요.

stop-pipe

다음 코드 예시는 stop-pipe의 사용 방법을 보여 줍니다.

AWS CLI

기존 파이프를 중지하려면

다음 stop-pipe 예제에서는 지정된 계정에서 Demo_Pipe라는 파이프를 중지합니다.

```
aws pipes stop-pipe \
    --name Demo_Pipe
```

출력:

```
{
    "Arn": "arn:aws:pipes:us-east-1:123456789012:pipe/Demo_Pipe",
    "Name": "Demo_Pipe",
    "DesiredState": "STOPPED",
    "CurrentState": "STOPPING",
    "CreationTime": "2024-10-08T09:29:10-05:00",
    "LastModifiedTime": "2024-10-08T09:29:49-05:00"
}
```

자세한 내용은 Amazon EventBridge 사용 설명서에서 Amazon EventBridge 파이프 시작 또는 중지를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StopPipe 섹션을 참조하세요.

tag-resource

다음 코드 예시는 tag-resource의 사용 방법을 보여 줍니다.

AWS CLI

기존 파이프에 태그를 지정하려면

다음 tag-resource 예제에서는 Demo_Pipe라는 파이프에 태그를 지정합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

```
aws pipes tag-resource \
    --resource-arn arn:aws:pipes:us-east-1:123456789012:pipe/Demo_Pipe \
    --tags stack=Production
```

자세한 내용은 Amazon EventBridge 사용 설명서의 Amazon EventBridge Pipes 개념을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

untag-resource

다음 코드 예시는 untag-resource의 사용 방법을 보여 줍니다.

AWS CLI

기존 파이프에서 태그를 제거하려면

다음 untag-resource 예제에서는 Demo_Pipe라는 파이프에서 stack 키가 있는 태그를 제거합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

```
aws pipes untag-resource \
    --resource-arn arn:aws:pipes:us-east-1:123456789012:pipe/Demo_Pipe \
    --tags stack
```

자세한 내용은 Amazon EventBridge 사용 설명서의 Amazon EventBridge Pipes 개념을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

update-pipe

다음 코드 예시는 update-pipe의 사용 방법을 보여 줍니다.

AWS CLI

기존 파이프를 업데이트하려면

다음 update-pipe 예제에서는 CloudWatch Log 구성 파라미터를 추가하여 Demo_Pipe라는 파이프를 업데이트하고, 파이프의 실행 역할을 업데이트하여 로그 대상에 대한 올바른 권한을 갖도록 합니다.

```
aws pipes update-pipe \
    --name Demo_Pipe \
    --desired-state RUNNING \
    --log-configuration CloudwatchLogsLogDestination={LogGroupArn=arn:aws:logs:us-
east-1:123456789012:log-group:/aws/vendedlogs/pipes/Demo_Pipe},Level=TRACE \
    --role-arn arn:aws:iam::123456789012:role/service-role/
Amazon_EventBridge_Pipe_Demo_Pipe_28b3aa4f
```

출력:

```
{
    "Arn": "arn:aws:pipes:us-east-1:123456789012:pipe/Demo_Pipe",
    "Name": "Demo_Pipe",
    "DesiredState": "RUNNING",
    "CurrentState": "UPDATING",
    "CreationTime": "2024-10-08T09:29:10-05:00",
    "LastModifiedTime": "2024-10-08T11:35:48-05:00"
```

```
}
```

자세한 내용은 Amazon EventBridge 사용 설명서의 Amazon EventBridge Pipes 개념을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdatePipe를 참조하세요.

AWS CLI를 사용한 Firewall Manager 예제

다음 코드 예제에서는 Firewall Manager에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여 줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`associate-admin-account`

다음 코드 예시는 `associate-admin-account`의 사용 방법을 보여 줍니다.

AWS CLI

Firewall Manager 관리자 계정을 설정하는 방법

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 AWS WAF, AWS Firewall Manager 및 AWS Shield Advanced 개발자 가이드의 AWS Firewall Manager 관리자 계정 설정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AssociateAdminAccount를 참조하세요.

delete-notification-channel

다음 코드 예시는 delete-notification-channel의 사용 방법을 보여 줍니다.

AWS CLI

Firewall Manager 로그에 대한 SNS 주제 정보를 제거하는 방법

다음 delete-notification-channel 예제에서는 SNS 주제 정보를 제거합니다.

```
aws fms delete-notification-channel
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS WAF, AWS Firewall Manager 및 AWS Shield Advanced 개발자 가이드의 Amazon SNS Notifications 및 Amazon CloudWatch 경보 구성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteNotificationChannel을 참조하세요.

delete-policy

다음 코드 예시는 delete-policy의 사용 방법을 보여 줍니다.

AWS CLI

Firewall Manager 정책을 삭제하는 방법

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 AWS WAF, AWS Firewall Manager 및 AWS Shield Advanced 개발자 가이드의 AWS Firewall Manager 정책 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeletePolicy를 참조하세요.

disassociate-admin-account

다음 코드 예시는 disassociate-admin-account의 사용 방법을 보여 줍니다.

AWS CLI

Firewall Manager 관리자 계정을 제거하는 방법

다음 disassociate-admin-account 예제에서는 Firewall Manager에서 현재 관리자 계정 연결을 제거합니다.

```
aws fms disassociate-admin-account
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS WAF, AWS Firewall Manager 및 AWS Shield Advanced 개발자 가이드의 AWS Firewall Manager 관리자 계정 설정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisassociateAdminAccount를 참조하세요.

get-admin-account

다음 코드 예시는 get-admin-account의 사용 방법을 보여 줍니다.

AWS CLI

Firewall Manager 관리자 계정을 검색하는 방법

다음 get-admin-account 예제에서는 관리자 계정을 검색합니다.

```
aws fms get-admin-account
```

출력:

```
{
    "AdminAccount": "123456789012",
    "RoleStatus": "READY"
}
```

자세한 내용은 AWS WAF, AWS Firewall Manager 및 AWS Shield Advanced 개발자 가이드의 AWS Firewall Manager 사전 조건을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetAdminAccount를 참조하세요.

get-compliance-detail

다음 코드 예시는 get-compliance-detail의 사용 방법을 보여 줍니다.

AWS CLI

계정의 규정 준수 정보를 검색하는 방법

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 개발자 가이드의 정책을 통한 리소스 규정 준수 보기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetComplianceDetail을 참조하세요.

get-notification-channel

다음 코드 예시는 get-notification-channel의 사용 방법을 보여 줍니다.

AWS CLI

Firewall Manager 로그에 대한 SNS 주제 정보를 검색하는 방법

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 AWS WAF, AWS Firewall Manager 및 AWS Shield Advanced 개발자 가이드의 Amazon SNS Notifications 및 Amazon CloudWatch 경보 구성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetNotificationChannel을 참조하세요.

`get-policy`

다음 코드 예시는 `get-policy`의 사용 방법을 보여 줍니다.

AWS CLI

Firewall Manager 정책을 검색하는 방법

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 AWS WAF, AWS Firewall Manager 및 AWS Shield Advanced 개발자 가이드의 AWS Firewall Manager 정책 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetPolicy를 참조하세요.

list-compliance-status

다음 코드 예시는 list-compliance-status의 사용 방법을 보여 줍니다.

AWS CLI

멤버 계정에 대한 정책 규정 준수 정보를 검색하는 방법

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 개발자 가이드의 정책을 통한 리소스 규정 준수 보기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListComplianceStatus를 참조하세요.

`list-member-accounts`

다음 코드 예시는 `list-member-accounts`의 사용 방법을 보여 줍니다.

AWS CLI

조직의 멤버 계정을 검색하는 방법

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 AWS WAF, AWS Firewall Manager 및 AWS Shield Advanced 개발자 가이드의 AWS Firewall Manager를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListMemberAccounts를 참조하세요.

list-policies

다음 코드 예시는 `list-policies`의 사용 방법을 보여 줍니다.

AWS CLI

모든 Firewall Manager 정책을 검색하는 방법

다음 `list-policies` 예제에서는 계정의 정책 목록을 검색합니다. 이 예제에서 출력은 요청 당 2개의 결과로 제한됩니다. 각 호출은 목록에 대한 다음 결과 집합을 얻기 위해 다음 `list-policies` 호출에서 `--starting-token` 파라미터의 값으로 사용할 수 있는 `NextToken`을 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 AWS WAF, AWS Firewall Manager 및 AWS Shield Advanced 개발자 가이드의 AWS Firewall Manager 정책 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListPolicies를 참조하세요.

put-notification-channel

다음 코드 예시는 put-notification-channel의 사용 방법을 보여 줍니다.

AWS CLI

Firewall Manager 로그에 대한 SNS 주제 정보를 설정하는 방법

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 AWS WAF, AWS Firewall Manager 및 AWS Shield Advanced 개발자 가이드의 Amazon SNS Notifications 및 Amazon CloudWatch 경보 구성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutNotificationChannel을 참조하세요.

put-policy

다음 코드 예시는 put-policy의 사용 방법을 보여 줍니다.

AWS CLI

Firewall Manager 정책을 생성하는 방법

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 AWS WAF, AWS Firewall Manager 및 AWS Shield Advanced 개발자 가이드의 AWS Firewall Manager 정책 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutPolicy를 참조하세요.

AWS CLI를 사용한 AWS FIS 예시

다음 코드 예시에서는 AWS FIS에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`create-experiment-template`

다음 코드 예시는 `create-experiment-template`의 사용 방법을 보여 줍니다.

AWS CLI

실험 템플릿을 생성하는 방법

다음 `create-experiment-template` 예제에서는 AWS FIS 계정에 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 Create an experiment template을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateExperimentTemplate을 참조하세요.

delete-experiment-template

다음 코드 예시는 delete-experiment-template의 사용 방법을 보여 줍니다.

AWS CLI

실험 템플릿을 삭제하는 방법

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 Delete an experiment template를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteExperimentTemplate을 참조하세요.

get-action

다음 코드 예시는 get-action의 사용 방법을 보여 줍니다.

AWS CLI

작업 세부 정보를 가져오는 방법

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 Actions를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetAction을 참조하세요.

get-experiment-template

다음 코드 예시는 get-experiment-template의 사용 방법을 보여 줍니다.

AWS CLI

실험 템플릿 세부 정보를 가져오는 방법

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 Experiment templates를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetExperimentTemplate을 참조하세요.

get-experiment

다음 코드 예시는 `get-experiment`의 사용 방법을 보여 줍니다.

AWS CLI

실험 세부 정보를 가져오는 방법

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 Experiments for AWS FIS을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetExperiment를 참조하세요.

`list-actions`

다음 코드 예시는 `list-actions`의 사용 방법을 보여 줍니다.

AWS CLI

작업을 나열하는 방법

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 Actions를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListActions를 참조하세요.

`list-experiment-templates`

다음 코드 예시는 `list-experiment-templates`의 사용 방법을 보여 줍니다.

AWS CLI

실험 템플릿을 나열하는 방법

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 Experiment templates를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListExperimentTemplates를 참조하세요.

`list-experiments`

다음 코드 예시는 `list-experiments`의 사용 방법을 보여 줍니다.

AWS CLI

실험을 나열하는 방법

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 Experiments를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListExperiments를 참조하세요.

list-tags-for-resource

다음 코드 예시는 `list-tags-for-resource`의 사용 방법을 보여 줍니다.

AWS CLI

리소스의 태그 나열

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 Tag your AWS FIS resources를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource를 참조하세요.

start-experiment

다음 코드 예시는 start-experiment의 사용 방법을 보여 줍니다.

AWS CLI

실험을 시작하는 방법

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 Experiments for AWS FIS을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartExperiment를 참조하세요.

stop-experiment

다음 코드 예시는 stop-experiment의 사용 방법을 보여 줍니다.

AWS CLI

실험을 중지하는 방법

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 Experiments for AWS FIS을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StopExperiment를 참조하세요.

tag-resource

다음 코드 예시는 tag-resource의 사용 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 Tag your AWS FIS resources를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

untag-resource

다음 코드 예시에서는 untag-resource 코드를 사용하는 방법을 보여줍니다.

AWS CLI

리소스에서 태그를 제거하는 방법

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 Tag your AWS FIS resources를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

update-experiment-template

다음 코드 예시는 update-experiment-template의 사용 방법을 보여 줍니다.

AWS CLI

실험 템플릿을 업데이트하는 방법

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 Update an experiment template을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateExperimentTemplate을 참조하세요.

AWS CLI를 사용한 Amazon GameLift Servers 예제

다음 코드 예제에서는 Amazon GameLift Servers에서 AWS Command Line Interface를 사용하여 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

create-build

다음 코드 예시에서는 create-build 코드를 사용하는 방법을 보여줍니다.

AWS CLI

예시 1: S3 버킷의 파일에서 게임 빌드를 생성하는 방법

다음 create-build 예시에서는 사용자 지정 게임 빌드 리소스를 생성합니다. 사용자가 제어하는 AWS 계정의 S3 위치에 저장된 압축 파일을 사용합니다. 이 예시에서는 Amazon GameLift에 S3 위치에 액세스할 수 있는 권한을 부여하는 IAM 역할을 이미 생성했다고 가정합니다. 요청에 운영 체제가 지정되지 않았으므로 새 빌드 리소스는 기본적으로 WINDOWS_2012로 설정됩니다.

```
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"
    }
}
```

예시 2: GameLift에 파일을 수동으로 업로드하기 위한 게임 빌드 리소스를 생성하는 방법

다음 `create-build` 예시에서는 새로운 빌드 리소스를 생성합니다. 또한 게임 빌드를 Amazon S3의 GameLift 위치에 수동으로 업로드할 수 있는 스토리지 위치와 임시 자격 증명을 얻습니다. 빌드를 성공적으로 업로드하면 GameLift 서비스에서 빌드의 유효성을 검사하고 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 Upload a Custom Server Build to GameLift를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateBuild 섹션을 참조하세요.

create-fleet

다음 코드 예시에서는 `create-fleet` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

예시 1: 기본 Linux 플릿 생성

다음 `create-fleet` 예시는 사용자 지정 서버 빌드를 호스팅하기 위해 최소한의 온디맨드 Linux 인스턴스로 구성된 플릿을 생성합니다. `update-fleet` 명령을 사용하여 이를 확인할 수 있습니다.

```
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"
    }
}
```

예시 2: 기본 Windows 플릿 생성

다음 `create-fleet` 예시는 사용자 지정 서버 빌드를 호스팅하기 위해 최소한의 스팟 Windows 인스턴스로 구성된 플릿을 생성합니다. `update-fleet` 명령을 사용하여 이를 확인할 수 있습니다.

```
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"
    }
}
```

예시 3: 완전히 구성된 플릿 생성

다음 `create-fleet` 예시는 사용자 지정 서버 빌드를 위한 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.json`의 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"
        }
    }
}
```


예시 4: Realtime Servers 플릿 생성

다음 `create-fleet` 예시에서는 Amazon GameLift에 업로드된 Realtime 구성 스크립트를 사용하여 스팟 인스턴스 플릿을 생성합니다. 모든 Realtime Servers는 Linux 머신에 배포됩니다. 이 예시에서는 업로드된 Realtime 스크립트에 여러 스크립트 파일이 포함되어 있고 `Init()` 함수는 스크립트 파일에 있는 `MainScript.js`로 가정합니다. 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateFleet 섹션을 참조하세요.

`create-game-session-queue`

다음 코드 예시에서는 `create-game-session-queue` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

예시 1: 순서가 지정된 게임 세션 대기열을 설정하는 방법

다음 `create-game-session-queue` 예시에서는 두 리전의 대상으로 새 게임 세션 대기열을 생성합니다. 또한 게임 세션 요청이 10분 동안 대기한 후 시간 초과되도록 대기열을 구성합니다. 지연 시간 정책이 정의되어 있지 않으므로 GameLift는 모든 게임 세션을 첫 번째 대상에 배치하려고 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"}
            ]
        }
    ]
}
```

예시 2: 플레이어 지연 시간 정책을 사용하여 게임 세션 대기열을 설정하는 방법

다음 create-game-session-queue 예시에서는 두 개의 플레이어 지연 시간 정책을 사용하여 새 게임 세션 대기열을 생성합니다. 첫 번째 정책은 게임 세션 배치 시도의 첫 1분 동안 적용되는 지연 시간 제한을 100ms로 설정합니다. 두 번째 정책은 배치 요청이 3분으로 제한될 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.json의 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.json의 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 Create a Queue를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateGameSessionQueue 섹션을 참조하세요.

delete-build

다음 코드 예시에서는 `delete-build` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용자 지정 게임 빌드를 삭제하는 방법

다음 `delete-build` 예시에서는 Amazon GameLift 계정에서 빌드를 제거합니다. 빌드가 삭제된 후에는 새 플릿을 만드는 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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteBuild 섹션을 참조하세요.

delete-fleet

다음 코드 예시에서는 `delete-fleet` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

더 이상 사용되지 않는 플릿을 삭제하는 방법

다음 delete-fleet 예시에서는 인스턴스가 0으로 스케일 다운된 플릿을 제거합니다. 플릿 용량이 0보다 크면 HTTP 400 오류와 함께 요청이 실패합니다.

```
aws gamelift delete-fleet \
   --fleet-id fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon GameLift 개발자 안내서의 Manage GameLift Fleets를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteFleet 섹션을 참조하세요.

delete-game-session-queue

다음 코드 예시에서는 delete-game-session-queue 코드를 사용하는 방법을 보여줍니다.

AWS CLI

게임 세션 대기열을 삭제하는 방법

다음 delete-game-session-queue 예시에서는 지정된 게임 세션 대기열을 삭제합니다.

```
aws gamelift delete-game-session-queue \
    --name MegaFrogRace-NA
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteGameSessionQueue 섹션을 참조하세요.

describe-build

다음 코드 예시에서는 describe-build 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용자 지정 게임 빌드에 대한 정보를 얻으려면

다음 describe-build 예시에서는 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 Upload a Custom Server Build to GameLift를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeBuild 섹션을 참조하세요.

describe-ec2-instance-limits

다음 코드 예시에서는 describe-ec2-instance-limits 코드를 사용하는 방법을 보여줍니다.

AWS CLI

EC2 인스턴스 유형에 대한 서비스 제한을 검색하는 방법

다음 describe-ec2-instance-limits 예시에서는 현재 리전에서 지정된 EC2 인스턴스 유형에 대해 사용 중인 최대 허용 인스턴스와 현재 인스턴스를 보여줍니다. 그 결과 허용된 20개의 인스턴스 중 5개만 사용되고 있는 것으로 나타났습니다.

```
aws gamelift describe-ec2-instance-limits \
    --ec2-instance-type m5.large
```

출력:

```
{
    "EC2InstanceLimits": [
        {
            "EC2InstanceType": ""m5.large",
            "CurrentInstances": 5,
            "InstanceLimit": 20
        }
    ]
}
```

자세한 내용은 Amazon CloudFront 개발자 안내서의 Choose Computing Resources를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeEc2InstanceLimits 섹션을 참조하세요.

describe-fleet-attributes

다음 코드 예시에서는 describe-fleet-attributes 코드를 사용하는 방법을 보여줍니다.

AWS CLI

예시 1: 플릿 목록의 속성을 보는 방법

다음 describe-fleet-attributes 예시에서는 지정된 두 플릿에 대한 플릿 속성을 검색합니다. 표시된 바와 같이 요청된 플릿은 동일한 빌드로 배포되며, 하나는 온디맨드 인스턴스용이고 다른 하나는 스팟 인스턴스용이지만 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"
            }
        }
    ]
}
```

예시 2: 모든 플릿에 대한 속성을 요청하는 방법

다음 describe-fleet-attributes 예시에서는 모든 상태의 모든 플릿에 대한 플릿 속성을 반환합니다. 이 예시에서는 페이지 매김 파라미터를 사용하여 한 번에 하나의 플릿을 반환하는 방법을 설명합니다.

```
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 출력에 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 Setting Up GameLift Fleets를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeFleetAttributes 섹션을 참조하세요.

describe-fleet-capacity

다음 코드 예시에서는 describe-fleet-capacity 코드를 사용하는 방법을 보여줍니다.

AWS CLI

플릿 목록의 용량 상태를 보는 방법

다음 describe-fleet-capacity 예시는 지정된 두 개의 플릿에 대한 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 Metrics for Fleets를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeFleetCapacity 섹션을 참조하세요.

describe-fleet-events

다음 코드 예시에서는 describe-fleet-events 코드를 사용하는 방법을 보여줍니다.

AWS CLI

지정된 기간 동안 이벤트를 요청하는 방법

다음 describe-fleet-events 예시에서는 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
```

```
%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 GDebug GameLift Fleet Issues를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeFleetEvents 섹션을 참조하세요.

describe-fleet-port-settings

다음 코드 예시에서는 `describe-fleet-port-settings` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

플릿에 대한 인바운드 연결 권한 보기

다음 `describe-fleet-port-settings` 예시에서는 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 Setting Up GameLift Fleets를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeFleetPortSettings 섹션을 참조하세요

describe-fleet-utilization

다음 코드 예시에서는 describe-fleet-utilization 코드를 사용하는 방법을 보여줍니다.

AWS CLI

예시 1: 플릿 목록의 사용량 데이터를 보는 방법

다음 describe-fleet-utilization 예시에서는 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
            }
    ]
}
```

예시 2: 모든 플릿에 대한 사용 데이터를 요청하는 방법

다음 `describe-fleet-utilization` 예시에서는 모든 상태의 모든 플릿에 대한 플릿 사용량 데이터를 반환합니다. 이 예시에서는 페이지 매김 파라미터를 사용하여 한 번에 두 개의 플릿에 대한 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 다시 한 번 호출하여 `--next-token` 파라미터에 `NextToken` 값을 인자로 전달하면 다음 두 가지 결과를 확인할 수 있습니다.

```
aws gamelift describe-fleet-utilization \
    --limit 2 \
    --next-
token eyJhd3NBY2NvdW50SWQiOnsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjp7InMiOiJidWlsZC01NWYxZT
```

응답에 더 이상 NextToken 값이 출력에 포함되지 않을 때까지 반복합니다.

자세한 내용은 Amazon GameLift 개발자 안내서의 GameLift Metrics for Fleets를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeFleetUtilization 섹션을 참조하세요.

describe-game-session-queues

다음 코드 예시에서는 describe-game-session-queues 코드를 사용하는 방법을 보여줍니다.

AWS CLI

게임 세션 대기열을 보는 방법

다음 describe-game-session-queues는 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 Using Multi-Region Queues를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeGameSessionQueues 섹션을 참조하세요.

describe-runtime-configuration

다음 코드 예시에서는 `describe-runtime-configuration` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

플릿에 대한 런타임 구성 요청

다음 `describe-runtime-configuration` 예시에서는 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 Run Multiple Processes on a Fleet을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeRuntimeConfiguration 섹션을 참조하세요.

`list-builds`

다음 코드 예시에서는 `list-builds` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

예시 1: 사용자 지정 게임 빌드 목록을 가져오는 방법

다음 `list-builds` 예시에서는 현재 리전의 모든 게임 서버 빌드에 대한 속성을 검색합니다. 샘플 요청은 페이지 매김 파라미터인 `Limit` 및 `NextToken`을 사용하여 순차적으로 결과를 검색하는 방법을 보여줍니다. 첫 번째 명령은 처음 두 개의 빌드를 검색합니다. 사용 가능한 결과가 두 개 이상이기 때문에 응답에는 더 많은 결과를 사용할 수 있음을 나타내는 `NextToken`이 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 호출하면 다음 두 개의 빌드를 확인할 수 있습니다.

```
aws gamelift list-builds \
    --limit 2
    --next-token eyJhd3NBY2NvdW50SWQiOnsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjp7InMiOiJidWlsZC01NWYxZT
```

응답에 NextToken 값이 포함되지 않을 때까지 반복합니다.

예시 2: 실패 상태의 사용자 지정 게임 빌드 목록을 가져오는 방법

다음 list-builds 예시에서는 현재 실패 상태가 있는 현재 리전의 모든 게임 서버 빌드에 대한 속성을 검색합니다.

```
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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListBuilds 섹션을 참조하세요.

list-fleets

다음 코드 예시에서는 list-fleets 코드를 사용하는 방법을 보여줍니다.

AWS CLI

예시 1: 리전의 모든 플릿 목록을 가져오는 방법

다음 list-fleets 예시에서는 현재 리전에 있는 모든 플릿의 플릿 ID를 표시합니다. 이 예시에서는 페이지 매김 파라미터를 사용하여 한 번에 두 개의 플릿 ID를 검색합니다. 응답에는 검색할 결과가 더 있음을 나타내는 next-token 속성이 포함되어 있습니다.

```
aws gamelift list-fleets \
    --limit 2
```

출력:

```
{
    "FleetIds": [
        "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"
    ],
    "NextToken":
 "eyJhd3NBY2NvdW50SWQiOnsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjp7InMiOiJidWlsZC01NWYxZTZmMS
}
```

여기에 표시된 것처럼 다음 명령에 이전 응답의 NextToken 값을 전달하여 다음 두 가지 결과를 얻을 수 있습니다.

```
aws gamelift list-fleets \
    --limit 2 \
    --next-
token eyJhd3NBY2NvdW50SWQiOnsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjp7InMiOiJidWlsZC00NDRlZj
```

예시 2: 특정 빌드 또는 스크립트가 있는 리전의 모든 플릿 목록을 가져오는 방법

다음 list-builds 예시에서는 지정된 게임 빌드와 함께 배포된 플릿의 ID를 검색합니다. Realtime Servers로 작업하는 경우 빌드 ID 대신 스크립트 ID를 제공할 수 있습니다. 이 예시에서는 제한 파라미터를 지정하지 않았으므로 결과에 최대 16개의 플릿 ID가 포함될 수 있습니다.

```
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"
    ]
```

```
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [ListFleets] 섹션을 참조하세요.

request-upload-credentials

다음 코드 예시에서는 request-upload-credentials 코드를 사용하는 방법을 보여줍니다.

AWS CLI

빌드 업로드를 위한 액세스 자격 증명을 새로 고치려면

다음 create-build 예시에서는 GameLift 빌드 파일을 Amazon S3 위치에 업로드하기 위한 새롭고 유효한 액세스 자격 증명을 가져옵니다. 자격 증명은 수명이 제한되어 있습니다. 원래 CreateBuild 요청에 대한 응답에서 빌드 ID를 얻습니다.

```
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 [Upload a Custom Server Build to GameLift]를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [RequestUploadCredentials] 섹션을 참조하세요.

start-fleet-actions

다음 코드 예시에서는 start-fleet-actions 코드를 사용하는 방법을 보여줍니다.

AWS CLI

플릿 자동 조정 활동을 다시 시작하는 방법

다음 start-fleet-actions 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 StartFleetActions 섹션을 참조하세요.

stop-fleet-actions

다음 코드 예시에서는 stop-fleet-actions 코드를 사용하는 방법을 보여줍니다.

AWS CLI

플릿의 자동 조정 활동을 중지하는 방법

다음 stop-fleet-actions 예시에서는 지정된 플릿에 정의된 모든 조정 정책의 사용을 중지합니다. 정책이 일시 중단된 후에도 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.

- API 세부 정보는 AWS CLI 명령 참조의 StopFleetActions 섹션을 참조하세요.

update-build

다음 코드 예시에서는 update-build 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용자 지정 게임 빌드를 업데이트하는 방법

다음 update-build 예시에서는 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 Update Your Build Files를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateBuild 섹션을 참조하세요.

update-game-session-queue

다음 코드 예시에서는 update-game-session-queue 코드를 사용하는 방법을 보여줍니다.

AWS CLI

게임 세션 대기열 구성 업데이트

다음 update-game-session-queue 예시에서는 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.json의 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 Using Multi-Region Queues를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateGameSessionQueue 섹션을 참조하세요.

upload-build

다음 코드 예시에서는 upload-build 코드를 사용하는 방법을 보여줍니다.

AWS CLI

예시 1: Linux 게임 서버 빌드 업로드

다음 upload-build 예시에서는 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
```

예시 2: Windows 게임 서버 빌드 업로드

다음 `upload-build` 예시에서는 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 Upload a Custom Server Build to GameLift를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UploadBuild 섹션을 참조하세요.

AWS CLI를 사용한 Global Accelerator 예시

다음 코드 예시에서는 에서 Global Accelerator에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

add-custom-routing-endpoints

다음 코드 예시에서는 add-custom-routing-endpoints 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용자 지정 라우팅 액셀러레이터의 엔드포인트 그룹에 VPC 서브넷 엔드포인트를 추가하는 방법

다음 add-custom-routing-endpoints 예시는 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 AWS Global Accelerator 개발자 안내서의 VPC subnet endpoints for custom routing accelerators in AWS Global Accelerator를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AddCustomRoutingEndpoints 섹션을 참조하세요.

advertise-byoip-cidr

다음 코드 예시에서는 advertise-byoip-cidr 코드를 사용하는 방법을 보여줍니다.

AWS CLI

주소 범위를 광고하는 방법

다음 advertise-byoip-cidr 예시에서는 AWS 리소스와 함께 사용하도록 프로비저닝한 주소 범위를 광고AWS하라는 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 Bring Your Own IP Address in AWS Global Accelerator를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AdvertiseByoipCidr 섹션을 참조하세요.

allow-custom-routing-traffic

다음 코드 예시에서는 allow-custom-routing-traffic 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용자 지정 라우팅 액셀러레이터에 대해 VPC 서브넷의 특정 Amazon EC2 인스턴스 대상으로 트래픽을 허용하는 방법

다음 allow-custom-routing-traffic 예시에서는 사용자 지정 라우팅 액셀러레이터에서 특정 Amazon EC2 인스턴스(대상) IP 주소와 포트가 트래픽을 수신할 수 있는 VPC 서브넷 엔드포인트에 대해 트래픽을 허용하도록 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 AWS Global Accelerator 개발자 안내서의 [VPC subnet endpoints for custom routing accelerators in AWS Global Accelerator]를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [AllowCustomRoutingTraffic] 섹션을 참조하세요.

`create-accelerator`

다음 코드 예시에서는 `create-accelerator` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

액셀러레이터 생성

다음 `create-accelerator` 예시에서는 두 개의 BYOIP 고정 IP 주소를 가진 두 개의 태그를 사용하여 액셀러레이터를 생성합니다. 액셀러레이터를 만들거나 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 Accelerators in AWS Global Accelerator를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateAccelerator 섹션을 참조하세요.

create-custom-routing-accelerator

다음 코드 예시에서는 create-custom-routing-accelerator 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용자 지정 라우팅 액셀러레이터 생성

다음 create-custom-routing-accelerator 예시에서는 태그 Name 및 Project를 사용하여 사용자 지정 라우팅 액셀러레이터를 생성합니다.

```
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 Custom routing accelerators in AWS Global Accelerator를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateCustomRoutingAccelerator 섹션을 참조하세요.

`create-custom-routing-endpoint-group`

다음 코드 예시에서는 `create-custom-routing-endpoint-group` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용자 지정 라우팅 액셀러레이터에 대한 엔드포인트 그룹을 생성하는 방법

다음 `create-custom-routing-endpoint-group` 예시에서는 사용자 지정 라우팅 액셀러레이터에 대한 엔드포인트 그룹을 만듭니다.

```
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 Endpoint groups for custom routing accelerators in AWS Global Accelerator를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateCustomRoutingEndpointGroup 섹션을 참조하세요.

create-custom-routing-listener

다음 코드 예시에서는 `create-custom-routing-listener` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용자 지정 라우팅 액셀러레이터에 대한 리스너를 생성하는 방법

다음 `create-custom-routing-listener` 예시에서는 사용자 지정 라우팅 액셀러레이터를 위해 포트 범위가 5000에서 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 Listeners for custom routing accelerators in AWS Global Accelerator를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateCustomRoutingListener 섹션을 참조하세요.

create-endpoint-group

다음 코드 예시에서는 `create-endpoint-group` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

엔드포인트 그룹을 생성하는 방법

다음 `create-endpoint-group` 예시에서는 하나의 엔드포인트가 있는 엔드포인트 그룹을 만듭니다.

```
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 Endpoint groups in AWS Global Accelerator를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateEndpointGroup 섹션을 참조하세요.

create-listener

다음 코드 예시에서는 `create-listener` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

리스너 생성

다음 `create-listener` 예시는 두 개의 포트를 가진 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 [Listeners in AWS Global Accelerator](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [CreateListener](#)를 참조하세요.

deny-custom-routing-traffic

다음 코드 예시에서는 `deny-custom-routing-traffic` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용자 지정 라우팅 액셀러레이터에서 트래픽을 수신할 수 없는 대상 주소를 지정하는 방법

다음 `deny-custom-routing-traffic` 예시에서는 사용자 지정 라우팅 액셀러레이터의 트래픽을 수신할 수 없는 서브넷 엔드포인트의 대상 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 AWS Global Accelerator 개발자 안내서의 [VPC subnet endpoints for custom routing accelerators in AWS Global Accelerator](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DenyCustomRoutingTraffic 섹션을 참조하세요.

deprovision-byoip-cidr

다음 코드 예시에서는 deprovision-byoip-cidr 코드를 사용하는 방법을 보여줍니다.

AWS CLI

주소 범위를 프로비저닝 해제하는 방법

다음 deprovision-byoip-cidr 예시에서는 AWS 리소스와 함께 사용하도록 프로비저닝한 지정된 주소 범위를 해제합니다.

```
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 Bring your own IP address in AWS Global Accelerator를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeprovisionByoipCidr 섹션을 참조하세요.

describe-accelerator-attributes

다음 코드 예시에서는 describe-accelerator-attributes 코드를 사용하는 방법을 보여줍니다.

AWS CLI

액셀러레이터의 속성을 설명하는 방법

다음 describe-accelerator-attributes 예시에서는 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 Accelerators in AWS Global Accelerator를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeAcceleratorAttributes 섹션을 참조하세요.

describe-accelerator

다음 코드 예시에서는 describe-accelerator 코드를 사용하는 방법을 보여줍니다.

AWS CLI

액셀러레이터 설명

다음 describe-accelerator 예시에서는 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 Accelerators in AWS Global Accelerator를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeAccelerator 섹션을 참조하세요.

describe-custom-routing-accelerator-attributes

다음 코드 예시에서는 describe-custom-routing-accelerator-attributes 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용자 지정 라우팅 액셀러레이터의 속성 설명

다음 describe-custom-routing-accelerator-attributes 예시에서는 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 Custom routing accelerators in AWS Global Accelerator를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeCustomRoutingAcceleratorAttributes 섹션을 참조하세요.

describe-custom-routing-accelerator

다음 코드 예시에서는 describe-custom-routing-accelerator 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용자 지정 라우팅 액셀러레이터 설명

다음 describe-custom-routing-accelerator 예시에서는 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 [Custom routing accelerators in AWS Global Accelerator](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeCustomRoutingAccelerator](#) 섹션을 참조하세요.

`describe-custom-routing-endpoint-group`

다음 코드 예시에서는 `describe-custom-routing-endpoint-group` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용자 지정 라우팅 액셀러레이터의 엔드포인트 그룹 설명

다음 `describe-custom-routing-endpoint-group` 예시에서는 사용자 지정 라우팅 액셀러레이터에 대한 엔드포인트 그룹을 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 Endpoint groups for custom routing accelerators in AWS Global Accelerator를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeCustomRoutingEndpointGroup 섹션을 참조하세요.

describe-custom-routing-listener

다음 코드 예시에서는 describe-custom-routing-listener 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용자 지정 라우팅 액셀러레이터의 리스너 설명

다음 describe-custom-routing-listener 예시에서는 사용자 지정 라우팅 액셀러레이터에 대한 리스너를 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 Listeners for custom routing accelerators in AWS Global Accelerator를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeCustomRoutingListener 섹션을 참조하세요.

describe-endpoint-group

다음 코드 예시에서는 describe-endpoint-group 코드를 사용하는 방법을 보여줍니다.

AWS CLI

엔드포인트 그룹 설명

다음 describe-endpoint-group 예시에서는 Amazon EC2 인스턴스, ALB 및 NLB 엔드포인트가 있는 엔드포인트 그룹에 대한 세부 정보를 검색합니다

```
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 Endpoint groups in AWS Global Accelerator를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeEndpointGroup 섹션을 참조하세요.

`describe-listener`

다음 코드 예시에서는 `describe-listener` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

리스너 설명

다음 `describe-listener` 예시에서는 리스터를 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 Listeners in AWS Global Accelerator를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeListener 섹션을 참조하세요.

list-accelerators

다음 코드 예시에서는 `list-accelerators` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

액셀러레이터 나열

다음 `list-accelerators` 예시에서는 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 Accelerators in AWS Global Accelerator를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListAccelerators 섹션을 참조하세요.

`list-byoip-cidr`

다음 코드 예시에서는 `list-byoip-cidr` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

주소 범위를 나열하는 방법

다음 `list-byoip-cidr` 예시에서는 Global Accelerator에서 사용하도록 프로비저닝한 기존 보유 IP 주소 사용(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 [Bring your own IP address in AWS Global Accelerator](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ListByoipCidr](#) 섹션을 참조하세요.

`list-custom-routing-accelerators`

다음 코드 예시에서는 `list-custom-routing-accelerators` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용자 지정 라우팅 액셀러레이터를 나열하는 방법

다음 `list-custom-routing-accelerators` 예시에서는 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": "6a6a6a6a6a6a6a.awsglobalaccelerator.com",
            "Status": "DEPLOYED",
            "CreatedTime": 1575585564.0,
            "LastModifiedTime": 1579809243.0
        },
    ]
}
```

자세한 내용은 AWS Global Accelerator 개발자 안내서의 Custom routing accelerators in AWS Global Accelerator를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListCustomRoutingAccelerators 섹션을 참조하세요.

list-custom-routing-endpoint-groups

다음 코드 예시에서는 `list-custom-routing-endpoint-groups` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용자 지정 라우팅 액셀러레이터에서 리스너의 엔드포인트 그룹 나열

다음 `list-custom-routing-endpoint-groups` 예시에서는 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 Endpoint groups for custom routing accelerators in AWS Global Accelerator를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListCustomRoutingEndpointGroups 섹션을 참조하세요.

list-custom-routing-listeners

다음 코드 예시에서는 `list-custom-routing-listeners` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용자 지정 라우팅 액셀러레이터의 리스너 나열

다음 `list-custom-routing-listeners` 예시에서는 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 Listeners for custom routing accelerators in AWS Global Accelerator를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListCustomRoutingListeners 섹션을 참조하세요.

`list-custom-routing-port-mappings-by-destination`

다음 코드 예시에서는 `list-custom-routing-port-mappings-by-destination` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

특정 사용자 지정 라우팅 액셀러레이터 대상의 포트 매핑 나열

다음 `list-custom-routing-port-mappings-by-destination` 예시에서는 사용자 지정 라우팅 액셀러레이터에 대한 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 How custom routing accelerators work in AWS Global Accelerator를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListCustomRoutingPortMappingsByDestination 섹션을 참조하세요.

`list-custom-routing-port-mappings`

다음 코드 예시에서는 `list-custom-routing-port-mappings` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용자 지정 라우팅 액셀러레이터의 정적 포트 매핑 나열

다음 `list-custom-routing-port-mappings` 예시에서는 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 How custom routing accelerators work in AWS Global Accelerator를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListCustomRoutingPortMappings 섹션을 참조하세요.

list-endpoint-groups

다음 코드 예시에서는 `list-endpoint-groups` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

엔드포인트 그룹을 나열하는 방법

다음 `list-endpoint-groups` 예시에서는 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 Endpoint Groups in AWS Global Accelerator를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListEndpointGroups 섹션을 참조하세요.

list-listeners

다음 코드 예시에서는 `list-listeners` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

리스너를 나열하는 방법

다음 `list-listeners` 예시에서는 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 Listeners in AWS Global Accelerator를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListListeners 섹션을 참조하세요.

list-tags-for-resource

다음 코드 예시에서는 `list-tags-for-resource` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

액셀러레이터의 태그를 나열하는 방법

다음 `list-tags-for-resource` 예시에서는 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 안내서에서 Tagging in AWS Global Accelerator를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource를 참조하세요.

provision-byoip-cidr

다음 코드 예시에서는 provision-byoip-cidr 코드를 사용하는 방법을 보여줍니다.

AWS CLI

주소 범위 프로비저닝

다음 provision-byoip-cidr 예시에서는 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 Bring your own IP address in AWS Global Accelerator를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ProvisionByoipCidr 섹션을 참조하세요.

tag-resource

다음 코드 예시에서는 tag-resource 코드를 사용하는 방법을 보여줍니다.

AWS CLI

액셀러레이터에 태그를 지정하는 방법

다음 tag-resource 예시에서는 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 안내서에서 Tagging in AWS Global Accelerator를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

untag-resource

다음 코드 예시에서는 untag-resource 코드를 사용하는 방법을 보여줍니다.

AWS CLI

액셀러레이터에서 태그를 제거하는 방법

다음 untag-resource 예시에서는 액셀러레이터에서 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 안내서에서 Tagging in AWS Global Accelerator를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

update-accelerator-attributes

다음 코드 예시에서는 update-accelerator-attributes 코드를 사용하는 방법을 보여줍니다.

AWS CLI

액셀러레이터의 속성을 업데이트하는 방법

다음 update-accelerator-attributes 예시에서는 액셀러레이터를 업데이트하여 흐름 로그를 활성화합니다. 액셀러레이터 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 Accelerators in AWS Global Accelerator를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateAcceleratorAttributes 섹션을 참조하세요.

`update-accelerator`

다음 코드 예시에서는 `update-accelerator` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

액셀러레이터를 업데이트하는 방법

다음 `update-accelerator` 예시에서는 액셀러레이터 이름을 `ExampleAcceleratorNew`로 변경하도록 액셀러레이터를 수정합니다. 액셀러레이터를 만들거나 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 Accelerators in AWS Global Accelerator를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateAccelerator 섹션을 참조하세요.

`update-custom-routing-accelerator-attributes`

다음 코드 예시에서는 `update-custom-routing-accelerator-attributes` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용자 지정 라우팅 액셀러레이터의 속성 업데이트

다음 `update-custom-routing-accelerator-attributes` 예시에서는 사용자 지정 라우팅 액셀러레이터를 업데이트하여 흐름 로그를 활성화합니다.

```
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 Custom routing accelerators in AWS Global Accelerator를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateCustomRoutingAcceleratorAttributes 섹션을 참조하세요.

`update-custom-routing-accelerator`

다음 코드 예시에서는 `update-custom-routing-accelerator` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용자 지정 라우팅 액셀러레이터 업데이트

다음 `update-custom-routing-accelerator` 예시에서는 사용자 지정 액셀러레이터 이름을 변경하도록 액셀러레이터를 수정합니다.

```
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 Custom routing accelerators in AWS Global Accelerator를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateCustomRoutingAccelerator 섹션을 참조하세요.

`update-custom-routing-listener`

다음 코드 예시에서는 `update-custom-routing-listener` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용자 지정 라우팅 액셀러레이터의 리스너 업데이트

다음 `update-custom-routing-listener` 예시에서는 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 Listeners for custom routing accelerators in AWS Global Accelerator를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateCustomRoutingListener 섹션을 참조하세요.

update-endpoint-group

다음 코드 예시에서는 update-endpoint-group 코드를 사용하는 방법을 보여줍니다.

AWS CLI

엔드포인트 그룹을 업데이트하는 방법

다음 update-endpoint-group 예시에서는 엔드포인트 그룹에 탄력적 IP 주소, ALB 및 NLB의 세 가지 엔드포인트를 추가합니다.

```
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 Endpoint groups in AWS Global Accelerator를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateEndpointGroup 섹션을 참조하세요.

update-listener

다음 코드 예시에서는 update-listener 코드를 사용하는 방법을 보여줍니다.

AWS CLI

리스너 업데이트

다음 `update-listener` 예시에서는 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 Listeners in AWS Global Accelerator를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateListener 섹션을 참조하세요.

`withdraw-byoip-cidr`

다음 코드 예시에서는 `withdraw-byoip-cidr` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

주소 범위를 철회하는 방법

다음 withdraw-byoip-cidr 예시에서는 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 Bring your own IP address in AWS Global Accelerator를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 WithdrawByoipCidr 섹션을 참조하세요.

AWS CLI를 사용한 AWS Glue 예시

다음 코드 예시에서는 AWS Glue에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

batch-stop-job-run

다음 코드 예시는 batch-stop-job-run의 사용 방법을 보여 줍니다.

AWS CLI

작업 실행을 중지하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 BatchStopJobRun을 참조하세요.

`create-connection`

다음 코드 예시는 `create-connection`의 사용 방법을 보여 줍니다.

AWS CLI

AWS Glue 데이터 스토어에 대한 연결을 생성하는 방법

다음 `create-connection` 예제에서는 Kafka 데이터 스토어에 대한 연결 정보를 제공하는 연결을 AWS Glue Data Catalog에 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 AWS Glue Data Catalog에서 연결 정의를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateConnection을 참조하세요.

`create-database`

다음 코드 예시는 `create-database`의 사용 방법을 보여 줍니다.

AWS CLI

데이터베이스를 생성하려면

다음 `create-database` 예제에서는 AWS Glue Data Catalog에 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 정의]를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [CreateDatabase]를 참조하세요.

create-job

다음 코드 예시는 create-job의 사용 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://amzn-s3-demo-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 개발자 안내서의 AWS Glue에 작업 작성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateJob을 참조하세요.

create-table

다음 코드 예시는 `create-table`의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: Kinesis 데이터 스트림에 대한 테이블을 생성하는 방법

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 AWS Glue 데이터 카탈로그의 테이블 정의를 참조하세요.

예제 2: Kafka 데이터 스토어에 대한 테이블을 생성하는 방법

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 AWS Glue 데이터 카탈로그의 테이블 정의를 참조하세요.

예제 3: AWS S3 데이터 스토어에 대한 테이블을 생성하는 방법

다음 `create-table` 예제에서는 AWS Glue Data Catalog에 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 AWS Glue 데이터 카탈로그의 테이블 정의를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateTable을 참조하세요.

delete-job

다음 코드 예시는 `delete-job`의 사용 방법을 보여 줍니다.

AWS CLI

작업 삭제

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 안내서에서 AWS Glue 콘솔에서 작업 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteJob을 참조하세요.

get-databases

다음 코드 예시는 get-databases의 사용 방법을 보여 줍니다.

AWS CLI

AWS Glue 데이터 카탈로그의 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.

- API에 대한 세부 정보는 AWS CLI 명령 참조의 GetDatabases를 참조하세요.

get-job-run

다음 코드 예시는 get-job-run의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 GetJobRun을 참조하세요.

get-job-runs

다음 코드 예시는 `get-job-runs`의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 GetJobRuns를 참조하세요.

get-job

다음 코드 예시는 get-job의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 GetJob을 참조하세요.

get-plan

다음 코드 예시는 get-plan의 사용 방법을 보여 줍니다.

AWS CLI

소스 테이블에서 대상 테이블로 데이터를 매핑하기 위해 생성된 코드를 가져오는 방법

다음 get-plan 코드는 데이터 소스에서 데이터 대상으로 열을 매핑하기 위해 생성된 코드를 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 AWS Glue에서 스크립트 편집을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetPlan을 참조하세요.

get-tables

다음 코드 예시는 get-tables의 사용 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 AWS Glue 데이터 카탈로그의 테이블 정의를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetTables를 참조하세요.

start-crawler

다음 코드 예시는 start-crawler의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 StartCrawler를 참조하세요.

start-job-run

다음 코드 예시는 start-job-run의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 StartJobRun을 참조하세요.

AWS CLI를 사용한 GuardDuty 예제

다음 코드 예제에서는 GuardDuty와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

작업

accept-invitation

다음 코드 예시는 `accept-invitation`의 사용 방법을 보여 줍니다.

AWS CLI

현재 리전에서 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.

- API 세부 정보는 AWS CLI 명령 참조의 AcceptInvitation을 참조하세요.

archive-findings

다음 코드 예시는 `archive-findings`의 사용 방법을 보여 줍니다.

AWS CLI

현재 리전에서 조사 결과를 아카이브하려면

이 `archive-findings` 예제에서는 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 금지 규칙 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ArchiveFindings를 참조하세요.

create-detector

다음 코드 예시는 `create-detector`의 사용 방법을 보여 줍니다.

AWS CLI

현재 리전에서 GuardDuty를 활성화하려면

이 예제에서는 현재 리전에서 GuardDuty를 활성화하는 새 감지기를 생성하는 방법을 보여줍니다.

```
aws guardduty create-detector \
    --enable
```

출력:

```
{
    "DetectorId": "b6b992d6d2f48e64bc59180bfexample"
}
```

자세한 내용은 GuardDuty 사용 설명서의 Amazon GuardDuty 활성화를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateDetector를 참조하세요.

create-filter

다음 코드 예시는 `create-filter`의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 현재 리전에 대한 새 필터 생성

다음 `create-filter` 예제에서는 특정 이미지에서 생성된 인스턴스의 모든 Portscan 조사 결과와 일치하는 필터를 생성합니다. 이렇게 하면 이러한 조사 결과도 금지되지 않습니다.

```
aws guardduty create-filter \
    --detector-id b6b992d6d2f48e64bc59180bfexample \
    --name myFilterExample \
    --finding-criteria '{"Criterion": {"type": {"Eq": ["Recon:EC2/
Portscan"]},"resource.instanceDetails.imageId": {"Eq": ["ami-0a7a207083example"]}}}'
```

출력:

```
{
    "Name": "myFilterExample"
}
```

자세한 내용은 GuardDuty 사용 설명서의 [GuardDuty 조사 결과 필터링]을 참조하세요.

예제 2: 현재 리전에서 새 필터를 생성하고 조사 결과를 금지

다음 `create-filter` 예제에서는 특정 이미지에서 생성된 인스턴스의 모든 Portscan 조사 결과와 일치하는 필터를 생성합니다. 이 필터는 현재 조사 결과에 나타나지 않도록 해당 조사 결과를 아카이브합니다.

```
aws guardduty create-filter \
    --detector-id b6b992d6d2f48e64bc59180bfexample \
    --action ARCHIVE \
    --name myFilterSecondExample \
    --finding-criteria '{"Criterion": {"type": {"Eq": ["Recon:EC2/
Portscan"]},"resource.instanceDetails.imageId": {"Eq": ["ami-0a7a207083example"]}}}'
```

출력:

```
{
    "Name": "myFilterSecondExample"
}
```

자세한 내용은 GuardDuty 사용 설명서의 [GuardDuty 조사 결과 필터링]을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [CreateFilter](#) 섹션을 참조하세요.

create-ip-set

다음 코드 예시는 create-ip-set의 사용 방법을 보여 줍니다.

AWS CLI

신뢰할 수 있는 IP 세트를 생성하고 활성화하려면

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-example \
    --format TXT \
    --location s3://amzn-s3-demo-bucket/customtrustlist.csv \
    --activate
```

출력:

```
{
    "IpSetId": "d4b94fc952d6912b8f3060768example"
}
```

자세한 내용은 GuardDuty 사용 설명서의 [신뢰할 수 있는 IP 목록 및 위협 목록 작업](#)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [CreateIpSet](#)를 참조하세요.

create-members

다음 코드 예시는 create-members의 사용 방법을 보여 줍니다.

AWS CLI

새 멤버를 현재 리전의 GuardDuty 마스터 계정에 연결하려면

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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateMembers를 참조하세요.

create-publishing-destination

다음 코드 예시는 create-publishing-destination의 사용 방법을 보여 줍니다.

AWS CLI

현재 리전의 GuardDuty 조사 결과를 내보낼 게시 대상을 만들려면

다음 create-publishing-destination 예제에서는 현재(아카이빙되지 않음) GuardDuty 조사 결과를 내보내 과거 조사 결과 데이터를 추적하도록 게시 대상을 설정하는 방법을 보여줍니다.

```
aws guardduty create-publishing-destination \
    --detector-id b6b992d6d2f48e64bc59180bfexample \
    --destination-type S3 \
    --destination-properties 'DestinationArn=arn:aws:s3:::amzn-s3-demo-
bucket,KmsKeyArn=arn:aws:kms:us-west-1:111122223333:key/84cee9c5-dea1-401a-ab6d-
e1de7example'
```

출력:

```
{
    "DestinationId": "46b99823849e1bbc242dfbe3cexample"
}
```

자세한 내용은 GuardDuty 사용 설명서의 Amazon S3 버킷으로 생성된 GuardDuty 조사 결과 내보내기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreatePublishingDestination을 참조하세요.

create-sample-findings

다음 코드 예시는 create-sample-findings의 사용 방법을 보여 줍니다.

AWS CLI

현재 리전에서 샘플 GuardDuty 조사 결과를 만들려면

이 예제에서는 제공된 유형의 샘플 조사 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 결과를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateSampleFindings를 참조하세요.

create-threat-intel-set

다음 코드 예시는 create-threat-intel-set의 사용 방법을 보여 줍니다.

AWS CLI

새로운 위협 IP 세트를 생성하고 활성화하려면

다음 create-threat-intel-set 예제에서는 현재 리전에서 위협 IP 세트를 생성하고 활성화합니다.

```
aws guardduty create-threat-intel-set \
    --detector-id b6b992d6d2f48e64bc59180bfexample \
    --name myThreatSet-example \
    --format TXT \
    --location s3://amzn-s3-demo-bucket/threatlist.csv \
    --activate
```

출력:

```
{
    "ThreatIntelSetId": "20b9a4691aeb33506b808878cexample"
```

```
}
```

자세한 내용은 GuardDuty 사용 설명서의 신뢰할 수 있는 IP 목록 및 위협 목록 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateThreatIntelSet를 참조하세요.

`decline-invitations`

다음 코드 예시는 `decline-invitations`의 사용 방법을 보여 줍니다.

AWS CLI

현재 리전의 다른 계정에서 Guardduty를 관리하도록 초대를 거부하려면

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.

- API 세부 정보는 AWS CLI 명령 참조의 DeclineInvitations를 참조하세요.

`delete-detector`

다음 코드 예시는 `delete-detector`의 사용 방법을 보여 줍니다.

AWS CLI

현재 리전에서 감지기를 삭제하고 GuardDuty를 비활성화하려면

이 예제에서는 감지기를 삭제하는 방법을 보여줍니다. 성공하면 해당 감지기와 연결된 리전에서 GuardDuty가 비활성화됩니다.

```
aws guardduty delete-detector \
    --detector-id b6b992d6d2f48e64bc59180bfexample
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 GuardDuty 사용 설명서의 GuardDuty 일시 중지 또는 비활성화를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteDetector를 참조하세요.

delete-filter

다음 코드 예시는 delete-filter의 사용 방법을 보여 줍니다.

AWS CLI

현재 리전에서 기존 필터를 삭제하려면

이 예제에서는 필터를 생성 및 삭제하는 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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteFilter를 참조하세요.

disable-organization-admin-account

다음 코드 예시는 disable-organization-admin-account의 사용 방법을 보여 줍니다.

AWS CLI

조직 내에서 GuardDuty의 위임된 관리자로서 계정을 제거하려면

이 예제에서는 GuardDuty의 위임된 관리자로서 계정을 제거하는 방법을 보여줍니다.

```
aws guardduty disable-organization-admin-account \
    --admin-account-id 111122223333
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 GuardDuty 사용 설명서의 AWS Organizations를 사용하여 계정 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisableOrganizationAdminAccount를 참조하세요.

disassociate-from-master-account

다음 코드 예시는 disassociate-from-master-account의 사용 방법을 보여 줍니다.

AWS CLI

현재 리전의 현재 마스터 계정에서 연결을 해제하려면

다음 disassociate-from-master-account 예제에서는 현재 AWS 리전의 현재 GuardDuty 관리자 계정에서 계정 연결을 해제합니다.

```
aws guardduty disassociate-from-master-account \
    --detector-id d4b040365221be2b54a6264dcexample
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 GuardDuty 사용 설명서의 GuardDuty 관리자 계정 및 멤버 계정 간의 관계 이해를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisassociateFromMasterAccount 섹션을 참조하세요.

get-detector

다음 코드 예시는 get-detector의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 GetDetector를 참조하세요.

get-findings

다음 코드 예시는 `get-findings`의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 특정 조사 결과의 세부 정보를 검색하려면

다음 `get-findings` 예제에서는 지정된 조사 결과의 전체 JSON 조사 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.

- API 세부 정보는 AWS CLI 명령 참조의 GetFindings를 참조하세요.

get-ip-set

다음 코드 예시는 get-ip-set의 사용 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://amzn-s3-demo-bucket.s3-us-west-2.amazonaws.com/
customlist.csv",
    "Tags": {},
    "Format": "TXT",
    "Name": "test-ip-set-example"
}
```

자세한 내용은 GuardDuty 사용 설명서의 신뢰할 수 있는 IP 목록 및 위협 목록 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetIpSet를 참조하세요.

get-master-account

다음 코드 예시는 get-master-account의 사용 방법을 보여 줍니다.

AWS CLI

현재 리전의 마스터 계정에 대한 세부 정보를 검색하려면

다음 `get-master-account` 예제에서는 현재 리전에서 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": "111122223333"
    }
}
```

자세한 내용은 GuardDuty 사용 설명서의 GuardDuty 관리자 계정 및 멤버 계정 간의 관계 이해를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetMasterAccount 섹션을 참조하세요.

`list-detectors`

다음 코드 예시는 `list-detectors`의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListDetectors를 참조하세요.

list-findings

다음 코드 예시는 `list-findings`의 사용 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 현재 리전의 모든 findingIds 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 조사 결과 기준과 일치하는 현재 리전의 조사 결과를 나열하려면

다음 `list-findings` 예제에서는 지정된 조사 결과 유형과 일치하는 모든 findingIds 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 조사 결과 기준 세트와 일치하는 현재 리전의 조사 결과를 나열하려면

다음 `list-findings` 예제에서는 JSON 파일에 지정된 대로 아카이브되지 않고 'testuser'라는 IAM 사용자와 관련된 모든 findingIds 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListFindings를 참조하세요.

`list-invitations`

다음 코드 예시는 `list-invitations`의 사용 방법을 보여 줍니다.

AWS CLI

현재 리전의 멤버 계정이 되기 위한 초대의 세부 정보를 나열하려면

다음 `list-invitations` 예제에서는 현재 리전에서 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListInvitations를 참조하세요.

list-ip-sets

다음 코드 예시는 list-ip-sets의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListIpSets를 참조하세요.

list-members

다음 코드 예시는 list-members의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 현재 리전의 현재 멤버만 나열

다음 list-members 예제에서는 현재 리전에서 GuardDuty 관리자 계정과 연결된 현재 멤버 계정의 세부 정보만 나열하고 제공합니다.

```
aws guardduty list-members \
    --detector-id 12abc34d567e8fa901bc2d34eexample \
    --only-associated="true"
```

출력:

```
{
    "Members": [
        {
            "RelationshipStatus": "Enabled",
            "InvitedAt": "2020-06-09T22:49:00.910Z",
            "MasterId": "111122223333",
            "DetectorId": "7ab8b2f61b256c87f793f6a86example",
            "UpdatedAt": "2020-06-09T23:08:22.512Z",
            "Email": "your+member@example.com",
            "AccountId": "123456789012"
        }
    ]
}
```

자세한 내용은 GuardDuty 사용 설명서의 GuardDuty 관리자 계정 및 멤버 계정 간의 관계 이해를 참조하세요.

예제 2: 현재 리전의 모든 멤버 나열

다음 `list-members` 예제에서는 현재 리전에서 GuardDuty 관리자의 초대를 연결 해제했거나 아직 수락하지 않은 계정을 포함하여 모든 멤버 계정의 세부 정보를 나열하고 제공합니다.

```
aws guardduty list-members \
    --detector-id 12abc34d567e8fa901bc2d34eexample \
    --only-associated="false"
```

출력:

```
{
    "Members": [
        {
            "RelationshipStatus": "Enabled",
            "InvitedAt": "2020-06-09T22:49:00.910Z",
            "MasterId": "111122223333",
            "DetectorId": "7ab8b2f61b256c87f793f6a86example",
            "UpdatedAt": "2020-06-09T23:08:22.512Z",
            "Email": "your+other+member@example.com",
            "AccountId": "555555555555"
        }
    ]
}
```

자세한 내용은 GuardDuty 사용 설명서의 GuardDuty 관리자 계정 및 멤버 계정 간의 관계 이해를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListMembers를 참조하세요.

update-ip-set

다음 코드 예시는 update-ip-set의 사용 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://amzn-s3-demo-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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateIpSet를 참조하세요.

AWS CLI를 사용한 AWS Health 예시

다음 코드 예시에서는 AWS Health에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

describe-affected-entities

다음 코드 예시는 describe-affected-entities의 사용 방법을 보여 줍니다.

AWS CLI

지정된 AWS Health 이벤트의 영향을 받는 엔터티를 나열하는 방법

다음 describe-affected-entities 예제에서는 지정된 AWS Health 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 Health 사용 설명서의 Event log를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeAffectedEntities를 참조하세요.

describe-event-details

다음 코드 예시는 describe-event-details의 사용 방법을 보여 줍니다.

AWS CLI

AWS Health 이벤트에 대한 정보를 나열하는 방법

다음 `describe-event-details` 예제에서는 지정된 AWS Health 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 Health 사용 설명서의 [Event details pane](#)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeEventDetails를 참조하세요.

describe-events

다음 코드 예시는 describe-events의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: AWS Health 이벤트를 나열하는 방법

다음 describe-events 예제에서는 최근 AWS Health 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 AWS Health 사용 설명서의 Getting started with the AWS Personal Health Dashboard를 참조하세요.

예제 2: 서비스 및 이벤트 상태 코드별 AWS Health 이벤트를 나열하는 방법

다음 `describe-events` 예제에서는 이벤트 상태가 종료된 Amazon Elastic Compute Cloud(Amazon EC2)에 대한 AWS Health 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 AWS Health 사용 설명서의 Getting started with the AWS Personal Health Dashboard를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeEvents를 참조하세요.

AWS CLI를 사용한 HealthImaging 예시

다음 코드 예시는 HealthImaging과 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

copy-image-set

다음 코드 예시에서는 copy-image-set의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 대상 없이 이미지 세트 복사

다음 copy-image-set 예시에서는 대상 없이 이미지 세트의 복제본을 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"
}
```

예시 2: 대상과 함께 이미지 세트 복사

다음 `copy-image-set` 예시에서는 대상과 함께 이미지 세트의 복제본을 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"
}
```

예시 3: 소스 이미지 세트의 인스턴스 하위 집합을 대상 이미지 세트로 복사

다음 copy-image-set 예시에서는 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 복사]를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [CopyImageSet]를 참조하세요.

`create-datastore`

다음 코드 예시에서는 `create-datastore`의 사용 방법을 보여줍니다.

AWS CLI

데이터 스토어 생성

다음 `create-datastore` 코드 예시에서는 `my-datastore`라는 데이터 스토어를 생성합니다.

```
aws medical-imaging create-datastore \
    --datastore-name "my-datastore"
```

출력:

```
{
    "datastoreId": "12345678901234567890123456789012",
    "datastoreStatus": "CREATING"
}
```

자세한 내용은 AWS HealthImaging 개발자 안내서의 데이터 스토어 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateDatastore를 참조하세요.

`delete-datastore`

다음 코드 예시에서는 `delete-datastore`의 사용 방법을 보여줍니다.

AWS CLI

데이터 스토어 삭제

다음 `delete-datastore` 코드 예시에서는 데이터 스토어를 삭제합니다.

```
aws medical-imaging delete-datastore \
    --datastore-id "12345678901234567890123456789012"
```

출력:

```
{
    "datastoreId": "12345678901234567890123456789012",
```

```
    "datastoreStatus": "DELETING"
}
```

자세한 내용은 AWS HealthImaging 개발자 안내서의 데이터 스토어 삭제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteDatastore를 참조하세요.

delete-image-set

다음 코드 예시에서는 delete-image-set의 사용 방법을 보여줍니다.

AWS CLI

이미지 세트 삭제

다음 delete-image-set 코드 예시에서는 이미지 세트를 삭제합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteImageSet를 참조하세요.

get-datastore

다음 코드 예시에서는 get-datastore의 사용 방법을 보여줍니다.

AWS CLI

데이터 스토어 속성 가져오기

다음 get-datastore 코드 예시에서는 데이터 스토어 속성을 가져옵니다.

```
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 가져오기](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [GetDatastore](#)를 참조하세요.

get-dicom-import-job

다음 코드 예시에서는 get-dicom-import-job의 사용 방법을 보여줍니다.

AWS CLI

dicom 가져오기 작업의 속성 가져오기

다음 get-dicom-import-job 코드 예시에서는 dicom 가져오기 작업의 속성을 가져옵니다.

```
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 세부 정보는 AWS CLI 명령 참조의 GetDICOMImportJob을 참조하세요.

get-image-frame

다음 코드 예시에서는 get-image-frame의 사용 방법을 보여줍니다.

AWS CLI

이미지 세트 픽셀 데이터 가져오기

다음 get-image-frame 코드 예시에서는 이미지 프레임을 가져옵니다.

```
aws medical-imaging get-image-frame \
    --datastore-id "12345678901234567890123456789012" \
    --image-set-id "98765412345612345678907890789012" \
    --image-frame-information imageFrameId=3abf5d5d7ae72f80a0ec81b2c0de3ef4 \
    imageframe.jph
```

참고: GetImageFrame 작업이 픽셀 데이터 스트림을 imageframe.jph 파일에 반환하므로 이 코드 예시에는 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.

- API 세부 정보는 AWS CLI 명령 참조의 [GetImageFrame](https://docs.aws.amazon.com/goto/aws-cli/medical-imaging-2023-07-19/GetImageFrame)을 참조하세요.

get-image-set-metadata

다음 코드 예시에서는 get-image-set-metadata의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 버전 없이 이미지 세트 메타데이터 가져오기

다음 get-image-set-metadata 코드 예시에서는 버전을 지정하지 않고 이미지 세트의 메타데이터를 가져옵니다.

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 콘텐츠를 보려면 먼저 압축을 풀어야 합니다.

출력:

```
{
    "contentType": "application/json",
    "contentEncoding": "gzip"
}
```

예시 2: 버전과 함께 이미지 세트 메타데이터 가져오기

다음 get-image-set-metadata 코드 예시에서는 지정된 버전의 이미지 세트에 대한 메타데이터를 가져옵니다.

참고: outfile은 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 콘텐츠를 보려면 먼저 압축을 풀어야 합니다.

출력:

```
{
    "contentType": "application/json",
    "contentEncoding": "gzip"
}
```

자세한 내용은 AWS HealthImaging 개발자 안내서의 이미지 세트 메타데이터 가져오기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetImageSetMetadata를 참조하세요.

get-image-set

다음 코드 예시에서는 `get-image-set`의 사용 방법을 보여줍니다.

AWS CLI

이미지 세트 속성 가져오기

다음 `get-image-set` 코드 예시에서는 이미지 세트의 속성을 가져옵니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 GetImageSet를 참조하세요.

list-datastores

다음 코드 예시에서는 list-datastores의 사용 방법을 보여줍니다.

AWS CLI

데이터 스토어 나열

다음 list-datastores 코드 예시에서는 사용 가능한 데이터 스토어를 나열합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 ListDatastores를 참조하세요.

list-dicom-import-jobs

다음 코드 예시에서는 list-dicom-import-jobs의 사용 방법을 보여줍니다.

AWS CLI

dicom 가져오기 작업 나열

다음 `list-dicom-import-jobs` 코드 예시에서는 dicom 가져오기 작업을 나열합니다.

```
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 세부 정보는 AWS CLI 명령 참조의 ListDICOMImportJobs를 참조하세요.

`list-image-set-versions`

다음 코드 예시에서는 `list-image-set-versions`의 사용 방법을 보여줍니다.

AWS CLI

이미지 세트 버전 나열

다음 `list-image-set-versions` 코드 예시에서는 이미지 세트의 버전 기록을 나열합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 [ListImageSetVersions](https://docs.aws.amazon.com/goto/aws-cli/medical-imaging-2023-07-19/ListImageSetVersions)를 참조하세요.

`list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 데이터 스토어의 리소스 태그 나열

다음 `list-tags-for-resource` 코드 예시에서는 데이터 스토어의 태그를 나열합니다.

```
aws medical-imaging list-tags-for-resource \
    --resource-arn "arn:aws:medical-imaging:us-
east-1:123456789012:datastore/12345678901234567890123456789012"
```

출력:

```
{
    "tags":{
        "Deployment":"Development"
    }
}
```

예시 2: 이미지 세트의 리소스 태그 나열

다음 `list-tags-for-resource` 코드 예시에서는 이미지 세트의 태그를 나열합니다.

```
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 AWS HealthImaging을 사용하여 리소스에 태그 지정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource를 참조하세요.

search-image-sets

다음 코드 예시에서는 search-image-sets의 사용 방법을 보여줍니다.

AWS CLI

예시 1: EQUAL 연산자로 이미지 세트 검색

다음 search-image-sets 코드 예시에서는 EQUAL 연산자를 사용하여 특정 값을 기준으로 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"
    }]
}
```

예시 2: DICOMStudyDate 및 DICOMStudyTime을 사용하여 BETWEEN 연산자로 이미지 세트 검색

다음 `search-image-sets` 코드 예시에서는 1990년 1월 1일(오전 12시)에서 2023년 1월 1일(오전 12시) 사이에 생성된 DICOM 연구가 있는 이미지 세트를 검색합니다.

참고: DICOMStudyTime은 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.json`의 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"
    }]
}
```

예시 3: CreatedAt을 사용하여 BETWEEN 연산자로 이미지 세트 검색(시간 연구가 이전에 지속됨)

다음 `search-image-sets` 코드 예시에서는 UTC 시간대의 시간 범위에서 HealthImaging에 대한 DICOM 연구가 지속된 이미지 세트를 검색합니다.

참고: createdAt을 예시 형식('1985-04-12T23:20:50.52Z')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"
    }]
}
```

예시 4: updatedAt의 DICOMSeriesInstanceUID 및 BETWEEN에서 EQUAL 연산자로 이미지 세트를 검색하고 updatedAt 필드의 ASC 순서로 응답 정렬

다음 `search-image-sets` 코드 예시에서는 DICOMSeriesInstanceUID의 EQUAL 연산자와 updatedAt의 BETWEEN을 사용하여 이미지 세트를 검색하고 updatedAt 필드의 ASC 순서로 응답을 정렬합니다.

참고: updatedAt을 예시 형식('1985-04-12T23:20:50.52Z')으로 제공합니다.

```
aws medical-imaging search-image-sets \
```

```
    --datastore-id 12345678901234567890123456789012 \
    --search-criteria  file://search-criteria.json
```

search-criteria.json의 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.

- API 세부 정보는 AWS CLI 명령 참조의 SearchImageSets를 참조하세요.

start-dicom-import-job

다음 코드 예시에서는 `start-dicom-import-job`의 사용 방법을 보여줍니다.

AWS CLI

dicom 가져오기 작업 시작

다음 `start-dicom-import-job` 코드 예시에서는 dicom 가져오기 작업을 시작합니다.

```
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 세부 정보는 AWS CLI 명령 참조의 StartDICOMImportJob을 참조하세요.

tag-resource

다음 코드 예시에서는 `tag-resource`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 데이터 스토어에 태그 지정

다음 `tag-resource` 코드 예시에서는 데이터 스토어에 태그를 지정합니다.

```
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.

예시 2: 이미지 세트에 태그 지정

다음 `tag-resource` 코드 예시에서는 이미지 세트에 태그를 지정합니다.

```
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 AWS HealthImaging을 사용하여 리소스에 태그 지정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

untag-resource

다음 코드 예시에서는 `untag-resource`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 데이터 스토어의 태그 해제

다음 `untag-resource` 코드 예시에서는 데이터 스토어의 태그를 해제합니다.

```
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.

예시 2: 이미지 세트의 태그 해제

다음 `untag-resource` 코드 예시에서는 이미지 세트의 태그를 해제합니다.

```
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 AWS HealthImaging을 사용하여 리소스에 태그 지정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

update-image-set-metadata

다음 코드 예시에서는 `update-image-set-metadata`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 이미지 세트 메타데이터에 속성을 삽입하거나 업데이트

다음 `update-image-set-metadata` 예시에서는 이미지 세트 메타데이터의 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.json`의 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"
}
```

예시 2: 이미지 세트 메타데이터에서 속성 제거

다음 `update-image-set-metadata` 예시에서는 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.json`의 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"
}
```

예시 3: 이미지 세트 메타데이터에서 인스턴스 제거

다음 `update-image-set-metadata` 예시에서는 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.json`의 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"
```

```
}
```

예시 4: 이미지 세트를 이전 버전으로 되돌리기

다음 `update-image-set-metadata` 예시에서는 이미지 세트를 이전 버전으로 되돌리는 방법을 보여줍니다. CopyImageSet 및 UpdateImageSetMetadata 작업은 새 버전의 이미지 세트를 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
}
```

예시 5: 인스턴스에 프라이빗 DICOM 데이터 요소 추가

다음 `update-image-set-metadata` 예시에서는 이미지 세트 내 지정된 인스턴스에 프라이빗 요소를 추가하는 방법을 보여줍니다. DICOM 표준은 표준 데이터 요소에 포함할 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.json`의 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"
}
```

예시 6: 프라이빗 DICOM 데이터 요소를 인스턴스로 업데이트

다음 `update-image-set-metadata` 예시에서는 이미지 세트 내 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.json`의 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"
}
```

예시 7: SOPInstanceUID를 강제 파라미터로 업데이트

다음 `update-image-set-metadata` 예시에서는 강제 파라미터를 사용하여 DICOM 메타데이터 제약 조건을 재정의하여 SOPInstanceUID를 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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateImageSetMetadata를 참조하세요.

AWS CLI를 사용한 HealthLake 예제

다음 코드 예제에서는 HealthLake에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여 줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`create-fhir-datastore`

다음 코드 예시는 `create-fhir-datastore`의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: SigV4-enabled HealthLake 데이터 스토어 생성

다음 `create-fhir-datastore` 예제에서는 AWS HealthLake에서 새 데이터 스토어를 생성하는 방법을 보여 줍니다.

```
aws healthlake create-fhir-datastore \
    --datastore-type-version R4 \
    --datastore-name "FhirTestDatastore"
```

출력:

```
{
    "DatastoreEndpoint": "https://healthlake.us-east-1.amazonaws.com/datastore/(Data
 store ID)/r4/",
    "DatastoreArn": "arn:aws:healthlake:us-east-1:(AWS Account ID):datastore/(Data
 store ID)",
    "DatastoreStatus": "CREATING",
    "DatastoreId": "(Data store ID)"
}
```

예제 2: FHIR 지원 HealthLake 데이터 스토어에서 SMART 생성

다음 `create-fhir-datastore` 예제에서는 AWS HealthLake의 FHIR 지원 데이터 스토어에서 새 SMART를 생성하는 방법을 보여 줍니다.

```
aws healthlake create-fhir-datastore \
    --datastore-name "your-data-store-name" \
    --datastore-type-version R4 \
    --preload-data-config PreloadDataType="SYNTHEA" \
    --sse-configuration '{ "KmsEncryptionConfig": {  "CmkType":
 "CUSTOMER_MANAGED_KMS_KEY", "KmsKeyId": "arn:aws:kms:us-east-1:your-account-id:key/
your-key-id" } }' \
    --identity-provider-configuration  file://identity_provider_configuration.json
```

`identity_provider_configuration.json`의 콘텐츠:

```
{
    "AuthorizationStrategy": "SMART_ON_FHIR_V1",
    "FineGrainedAuthorizationEnabled": true,
    "IdpLambdaArn": "arn:aws:lambda:your-region:your-account-id:function:your-
lambda-name",
    "Metadata": "{\"issuer\":\"https://ehr.example.com\", \"jwks_uri\":\"https://
ehr.example.com/.well-known/jwks.json\",\"authorization_endpoint\":\"https://
```

```
ehr.example.com/auth/authorize\",\"token_endpoint\":\"https://ehr.token.com/auth/
token\",\"token_endpoint_auth_methods_supported\":[\"client_secret_basic\",\"foo\"],
\"grant_types_supported\":[\"client_credential\",\"foo\"],\"registration_endpoint\":
\"https://ehr.example.com/auth/register\",\"scopes_supported\":[\"openId\",\"profile
\",\"launch\"],\"response_types_supported\":[\"code\"],\"management_endpoint\":
\"https://ehr.example.com/user/manage\",\"introspection_endpoint\":\"https://
ehr.example.com/user/introspect\",\"revocation_endpoint\":\"https://ehr.example.com/
user/revoke\",\"code_challenge_methods_supported\":[\"S256\"],\"capabilities\":
[\"launch-ehr\",\"sso-openid-connect\",\"client-public\"]}"
}
```

출력:

```
{
    "DatastoreEndpoint": "https://healthlake.us-east-1.amazonaws.com/datastore/(Data
 store ID)/r4/",
    "DatastoreArn": "arn:aws:healthlake:us-east-1:(AWS Account ID):datastore/(Data
 store ID)",
    "DatastoreStatus": "CREATING",
    "DatastoreId": "(Data store ID)"
}
```

자세한 내용은 AWS HealthLake 개발자 안내서의 FHIR 데이터 스토어 생성 및 모니터링을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateFHIRDatastore 섹션을 참조하세요.

delete-fhir-datastore

다음 코드 예시는 delete-fhir-datastore의 사용 방법을 보여 줍니다.

AWS CLI

FHIR 데이터 스토어를 삭제하려면

다음 delete-fhir-datastore 예제에서는 AWS HealthLake에서 데이터 스토어와 모든 콘텐츠를 삭제하는 방법을 보여 줍니다.

```
aws healthlake delete-fhir-datastore \
    --datastore-id (Data store ID)
```

출력:

```
{
    "DatastoreEndpoint": "https://healthlake.us-east-1.amazonaws.com/datastore/(Data
 store ID)/r4/",
    "DatastoreArn": "arn:aws:healthlake:us-east-1:(AWS Account ID):datastore/(Data
 store ID)",
    "DatastoreStatus": "DELETING",
    "DatastoreId": "(Data store ID)"
}
```

자세한 내용은 AWS HealthLake 개발자 안내서의 FHIR 데이터 스토어 <https://docs.aws.amazon.com/healthlake/latest/devguide/working-with-FHIR-healthlake.html> 생성 및 모니터링을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteFHIRDatastore 섹션을 참조하세요.

describe-fhir-datastore

다음 코드 예시는 describe-fhir-datastore의 사용 방법을 보여 줍니다.

AWS CLI

FHIR 데이터 스토어를 설명하려면

다음 describe-fhir-datastore 예제에서는 AWS HealthLake에서 데이터 스토어의 속성을 찾는 방법을 보여 줍니다.

```
aws healthlake describe-fhir-datastore \
    --datastore-id "1f2f459836ac6c513ce899f9e4f66a59"
```

출력:

```
{
    "DatastoreProperties": {
        "PreloadDataConfig": {
            "PreloadDataType": "SYNTHEA"
        },
        "SseConfiguration": {
            "KmsEncryptionConfig": {
                "CmkType": "CUSTOMER_MANAGED_KMS_KEY",
                "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
            }
```

```
        },
        "DatastoreName": "Demo",
        "DatastoreArn": "arn:aws:healthlake:us-east-1:<AWS Account ID>:datastore/
<Data store ID>",
        "DatastoreEndpoint": "https://healthlake.us-east-1.amazonaws.com/datastore/
<Data store ID>/r4/",
        "DatastoreStatus": "ACTIVE",
        "DatastoreTypeVersion": "R4",
        "CreatedAt": 1603761064.881,
        "DatastoreId": "<Data store ID>",
        "IdentityProviderConfiguration": {
            "AuthorizationStrategy": "AWS_AUTH",
            "FineGrainedAuthorizationEnabled": false
        }
    }
}
```

자세한 내용은 AWS HealthLake 개발자 안내서의 FHIR 데이터 스토어 생성 및 모니터링을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeFHIRDatastore 섹션을 참조하세요.

describe-fhir-export-job

다음 코드 예시는 describe-fhir-export-job의 사용 방법을 보여 줍니다.

AWS CLI

FHIR 내보내기 작업을 설명하는 방법

다음 describe-fhir-export-job 예제는 AWS HealthLake 에서 FHIR 내보내기 작업의 속성을 찾는 방법을 보여 줍니다.

```
aws healthlake describe-fhir-export-job \
    --datastore-id (Data store ID) \
    --job-id 9b9a51943afaedd0a8c0c26c49135a31
```

출력:

```
{
    "ExportJobProperties": {
        "DataAccessRoleArn": "arn:aws:iam::(AWS Account ID):role/(Role Name)",
```

```
        "JobStatus": "IN_PROGRESS",
        "JobId": "9009813e9d69ba7cf79bcb3468780f16",
        "SubmitTime": "2024-11-20T11:31:46.672000-05:00",
        "EndTime": "2024-11-20T11:34:01.636000-05:00",
        "OutputDataConfig": {
            "S3Configuration": {
            "S3Uri": "s3://(Bucket Name)/(Prefix Name)/",
            "KmsKeyId": "arn:aws:kms:us-east-1:012345678910:key/d330e7fc-b56c-4216-
a250-f4c43ef46e83"
        }

        },
        "DatastoreId": "(Data store ID)"
    }
}
```

자세한 내용은 AWS HealthLake 개발자 안내서의 FHIR 데이터 스토어에서 파일 내보내기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeFHIRExportJob 섹션을 참조하세요.

describe-fhir-import-job

다음 코드 예시는 describe-fhir-import-job의 사용 방법을 보여 줍니다.

AWS CLI

FHIR 가져오기 작업을 설명하는 방법

다음 describe-fhir-import-job 예제에서는 AWS HealthLake를 사용하여 FHIR 가져오기 작업의 속성을 학습하는 방법을 보여 줍니다.

```
aws healthlake describe-fhir-import-job \
    --datastore-id (Data store ID) \
    --job-id c145fbb27b192af392f8ce6e7838e34f
```

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": "(Data store ID)"
    }
}
```

자세한 내용은 AWS HealthLake 개발자 안내서의 FHIR 데이터 스토어에 파일 가져오기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeFHIRImportJob 섹션을 참조하세요.

list-fhir-datastores

다음 코드 예시는 list-fhir-datastores의 사용 방법을 보여 줍니다.

AWS CLI

FHIR 데이터 스토어를 나열하려면

다음 list-fhir-datastores 예제는 명령을 사용하는 방법과 사용자가 AWS HealthLake의 데이터 스토어 상태를 기반으로 결과를 필터링하는 방법을 보여 줍니다.

```
aws healthlake list-fhir-datastores \
    --filter DatastoreStatus=ACTIVE
```

출력:

```
{
    "DatastorePropertiesList": [
    {
        "PreloadDataConfig": {
            "PreloadDataType": "SYNTHEA"
        },
        "SseConfiguration": {
            "KmsEncryptionConfig": {
                "CmkType": "CUSTOMER_MANAGED_KMS_KEY",
                "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

```
            }
        },
        "DatastoreName": "Demo",
        "DatastoreArn": "arn:aws:healthlake:us-east-1:<AWS Account ID>:datastore/
<Data store ID>",
        "DatastoreEndpoint": "https://healthlake.us-east-1.amazonaws.com/datastore/
<Data store ID>/r4/",
        "DatastoreStatus": "ACTIVE",
        "DatastoreTypeVersion": "R4",
        "CreatedAt": 1603761064.881,
        "DatastoreId": "<Data store ID>",
        "IdentityProviderConfiguration": {
            "AuthorizationStrategy": "AWS_AUTH",
            "FineGrainedAuthorizationEnabled": false
        }
    }
    ]
}
```

자세한 내용은 AWS HealthLake 개발자 안내서의 FHIR 데이터 스토어 생성 및 모니터링을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListFHIRDatastores 섹션을 참조하세요.

list-fhir-export-jobs

다음 코드 예시는 `list-fhir-export-jobs`의 사용 방법을 보여 줍니다.

AWS CLI

모든 FHIR 내보내기 작업을 나열하는 방법

다음 `list-fhir-export-jobs` 예제에서는 명령을 사용하여 계정과 연결된 내보내기 작업 목록을 보는 방법을 보여 줍니다.

```
aws healthlake list-fhir-export-jobs \
    --datastore-id (Data store 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:

```
{
    "ExportJobPropertiesList": [
        {
            "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": "2024-11-20T11:31:46.672000-05:00",
                "EndTime": "2024-11-20T11:34:01.636000-05:00",
                "DatastoreId": "(Data store ID)"
            }
        }
    ]
}
```

자세한 내용은 AWS HealthLake 개발자 안내서의 FHIR 데이터 스토어에서 파일 내보내기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListFHIRExportJobs 섹션을 참조하세요.

list-fhir-import-jobs

다음 코드 예시는 `list-fhir-import-jobs`의 사용 방법을 보여 줍니다.

AWS CLI

모든 FHIR 가져오기 작업을 나열하는 방법

다음 `list-fhir-import-jobs` 예제에서는 명령을 사용하여 계정과 연결된 모든 가져오기 작업 목록을 보는 방법을 보여 줍니다.

```
aws healthlake list-fhir-import-jobs \
```

```
    --datastore-id (Data store 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:


```
{
    "ImportJobPropertiesList": [
        {
            "JobId": "c0fddbf76f238297632d4aebdbfc9ddf",
            "JobStatus": "COMPLETED",
            "SubmitTime": "2024-11-20T10:08:46.813000-05:00",
            "EndTime": "2024-11-20T10:10:09.093000-05:00",
            "DatastoreId": "(Data store ID)",
            "InputDataConfig": {
                "S3Uri": "s3://(Bucket Name)/(Prefix Name)/"
            },
            "JobOutputDataConfig": {
                "S3Configuration": {
                    "S3Uri": "s3://(Bucket Name)/
import/6407b9ae4c2def3cb6f1a46a0c599ec0-FHIR_IMPORT-
c0fddbf76f238297632d4aebdbfc9ddf/",
                    "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/b7f645cb-
e564-4981-8672-9e012d1ff1a0"
                }
            },
            "JobProgressReport": {
                "TotalNumberOfScannedFiles": 1,
                "TotalSizeOfScannedFilesInMB": 0.001798,
                "TotalNumberOfImportedFiles": 1,
                "TotalNumberOfResourcesScanned": 1,
                "TotalNumberOfResourcesImported": 1,
                "TotalNumberOfResourcesWithCustomerError": 0,
                "TotalNumberOfFilesReadWithCustomerError": 0,
                "Throughput": 0.0
            },
            "DataAccessRoleArn": "arn:aws:iam::(AWS Account ID):role/(Role Name)"
        }
    ]
}
```

자세한 내용은 AWS HealthLake 개발자 안내서의 FHIR 데이터 스토어에 파일 가져오기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListFHIRImportJobs 섹션을 참조하세요.

list-tags-for-resource

다음 코드 예시는 list-tags-for-resource의 사용 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-1:123456789012:datastore/
fhir/0725c83f4307f263e16fd56b6d8ebdbe"
```

출력:

```
{
    "tags": {
        "key": "value",
        "key1": "value1"
    }
}
```

자세한 내용은 AWS HealthLake 개발자 안내서의 AWS HealthLake의 리소스에 태그 지정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource를 참조하세요.

start-fhir-export-job

다음 코드 예시는 start-fhir-export-job의 사용 방법을 보여 줍니다.

AWS CLI

FHIR 내보내기 작업을 시작하는 방법

다음 `start-fhir-export-job` 예제에서는 AWS HealthLake를 사용하여 FHIR 내보내기 작업을 시작하는 방법을 보여줍니다.

```
aws healthlake start-fhir-export-job \
    --output-data-config '{"S3Configuration": {"S3Uri":"s3://(Bucket Name)/(Prefix Name)/","KmsKeyId":"arn:aws:kms:us-east-1:012345678910:key/d330e7fc-b56c-4216-a250-f4c43ef46e83"}}' \
    --datastore-id (Data store ID) \
    --data-access-role-arn arn:aws:iam::(AWS Account ID):role/(Role Name)
```

출력:

```
{
    "DatastoreId": "(Data store ID)",
    "JobStatus": "SUBMITTED",
    "JobId": "9b9a51943afaedd0a8c0c26c49135a31"
}
```

자세한 내용은 AWS HealthLake 개발자 안내서의 FHIR 데이터 스토어에서 파일 내보내기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartFHIRExportJob 섹션을 참조하세요.

`start-fhir-import-job`

다음 코드 예시는 `start-fhir-import-job`의 사용 방법을 보여 줍니다.

AWS CLI

FHIR 가져오기 작업을 시작하는 방법

다음 `start-fhir-import-job` 예제에서는 AWS HealthLake를 사용하여 FHIR 가져오기 작업을 시작하는 방법을 보여줍니다.

```
aws healthlake start-fhir-import-job \
    --input-data-config S3Uri="s3://(Bucket Name)/(Prefix Name)/" \
    --job-output-data-config '{"S3Configuration": {"S3Uri":"s3://(Bucket Name)/(Prefix Name)/","KmsKeyId":"arn:aws:kms:us-east-1:012345678910:key/d330e7fc-b56c-4216-a250-f4c43ef46e83"}}' \
    --datastore-id (Data store ID) \
    --data-access-role-arn "arn:aws:iam::(AWS Account ID):role/(Role Name)"
```

출력:

```
{
    "DatastoreId": "(Data store ID)",
    "JobStatus": "SUBMITTED",
    "JobId": "c145fbb27b192af392f8ce6e7838e34f"
}
```

자세한 내용은 AWS HealthLake 개발자 안내서의 FHIR 데이터 스토어에 파일 가져오기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartFHIRImportJob 섹션을 참조하세요.

`tag-resource`

다음 코드 예시는 `tag-resource`의 사용 방법을 보여 줍니다.

AWS CLI

데이터 스토어에 태그를 추가하려면

다음 `tag-resource` 예제에서는 데이터 스토어에 태그를 추가하는 방법을 보여 줍니다.

```
aws healthlake tag-resource \
    --resource-arn "arn:aws:healthlake:us-east-1:123456789012:datastore/
fhir/0725c83f4307f263e16fd56b6d8ebdbe" \
    --tags '[{"Key": "key1", "Value": "value1"}]'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS HealthLake개발자 안내서의 데이터 스토어에 태그 추가를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

`untag-resource`

다음 코드 예시는 `untag-resource`의 사용 방법을 보여 줍니다.

AWS CLI

데이터 스토어에서 태그를 제거하려면

다음 `untag-resource` 예제에서는 데이터 스토어에서 태그를 제거하는 방법을 보여 줍니다.

```
aws healthlake untag-resource \
    --resource-arn "arn:aws:healthlake:us-east-1:123456789012:datastore/fhir/
b91723d65c6fdeb1d26543a49d2ed1fa" \
    --tag-keys '["key1"]'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS HealthLake 개발자 안내서의 데이터 스토어에서 태그 제거를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

AWS CLI를 사용한 HealthOmics 예시

다음 코드 예시는 HealthOmics와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

abort-multipart-read-set-upload

다음 코드 예시에서는 `abort-multipart-read-set-upload`의 사용 방법을 보여줍니다.

AWS CLI

멀티파트 읽기 세트 업로드 중지

다음 `abort-multipart-read-set-upload` 예시에서는 HealthOmics 시퀀스 저장소로의 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 사용자 안내서의 시퀀스 저장소에 직접 업로드를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AbortMultipartReadSetUpload를 참조하세요.

accept-share

다음 코드 예시에서는 accept-share의 사용 방법을 보여줍니다.

AWS CLI

분석 저장소 데이터의 공유 수락

다음 accept-share 예시에서는 HealthOmics 분석 저장소 데이터의 공유를 수락합니다.

```
aws omics accept-share \
    ----share-id "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a"
```

출력:

```
{
    "status": "ACTIVATING"
}
```

자세한 내용은 AWS HealthOmics 사용자 안내서의 교차 계정 공유를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AcceptShare를 참조하세요.

batch-delete-read-set

다음 코드 예시에서는 batch-delete-read-set의 사용 방법을 보여줍니다.

AWS CLI

여러 읽기 세트 삭제

다음 batch-delete-read-set 예시에서는 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 Storage를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 BatchDeleteReadSet를 참조하세요.

`cancel-annotation-import-job`

다음 코드 예시에서는 `cancel-annotation-import-job`의 사용 방법을 보여줍니다.

AWS CLI

주석 가져오기 작업 취소

다음 `cancel-annotation-import-job` 예시에서는 ID가 `04f57618-xmpl-4fd0-9349-e5a85aefb997`인 주석 가져오기 작업을 취소합니다.

```
aws omics cancel-annotation-import-job \
    --job-id 04f57618-xmpl-4fd0-9349-e5a85aefb997
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics Analytics를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CancelAnnotationImportJob을 참조하세요.

`cancel-run`

다음 코드 예시에서는 `cancel-run`의 사용 방법을 보여줍니다.

AWS CLI

실행 취소

다음 `cancel-run` 예시에서는 ID가 `1234567`인 실행을 취소합니다.

```
aws omics cancel-run \
    --id 1234567
```

자세한 내용은 AWS HealthOmics 사용 설명서의 Run lifecycle in a workflow를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CancelRun을 참조하세요.

cancel-variant-import-job

다음 코드 예시에서는 cancel-variant-import-job의 사용 방법을 보여줍니다.

AWS CLI

변형 가져오기 작업 취소

다음 cancel-variant-import-job 예시에서는 ID가 69cb65d6-xmpl-4a4a-9025-4565794b684e인 변형 가져오기 작업을 취소합니다.

```
aws omics cancel-variant-import-job \
    --job-id 69cb65d6-xmpl-4a4a-9025-4565794b684e
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics Analytics를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CancelVariantImportJob을 참조하세요.

complete-multipart-read-set-upload

다음 코드 예시에서는 complete-multipart-read-set-upload의 사용 방법을 보여줍니다.

AWS CLI

모든 구성 요소를 업로드한 후 멀티파트 업로드 완료

다음 complete-multipart-read-set-upload 예시에서는 모든 구성 요소가 업로드되면 시퀀스 저장소에 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 사용자 안내서의 시퀀스 저장소에 직접 업로드를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CompleteMultipartReadSetUpload를 참조하세요.

create-annotation-store-version

다음 코드 예시에서는 `create-annotation-store-version`의 사용 방법을 보여줍니다.

AWS CLI

주석 저장소의 새 버전 생성

다음 `create-annotation-store-version` 예시에서는 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 사용자 안내서의 새 버전의 주석 저장소 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateAnnotationStoreVersion을 참조하세요.

`create-annotation-store`

다음 코드 예시에서는 `create-annotation-store`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: VCF 주석 저장소 생성

다음 `create-annotation-store` 예시에서는 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"
}
```

예시 2: TSV 주석 저장소 생성

다음 `create-annotation-store` 예시에서는 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`은 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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateAnnotationStore를 참조하세요.

create-multipart-read-set-upload

다음 코드 예시에서는 create-multipart-read-set-upload의 사용 방법을 보여줍니다.

AWS CLI

멀티파트 읽기 세트 업로드 시작

다음 create-multipart-read-set-upload 예시에서는 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 사용자 안내서의 시퀀스 저장소에 직접 업로드를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateMultipartReadSetUpload를 참조하세요.

create-reference-store

다음 코드 예시에서는 create-reference-store의 사용 방법을 보여줍니다.

AWS CLI

참조 저장소 생성

다음 create-reference-store 예시에서는 참조 저장소 my-ref-store를 생성합니다.

```
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 Storage를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateReferenceStore를 참조하세요.

`create-run-group`

다음 코드 예시에서는 `create-run-group`의 사용 방법을 보여줍니다.

AWS CLI

실행 그룹 생성

다음 `create-run-group` 예시에서는 `cram-converter`라는 실행 그룹을 생성합니다.

```
aws omics create-run-group \
    --name cram-converter \
    --max-cpus 20 \
    --max-gpus 10 \
    --max-duration 600 \
    --max-runs 5
```

출력:

```
{
    "arn": "arn:aws:omics:us-west-2:123456789012:runGroup/1234567",
    "id": "1234567",
    "tags": {}
}
```

자세한 내용은 AWS HealthOmics 사용 설명서의 Creating run groups를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateRunGroup을 참조하세요.

`create-sequence-store`

다음 코드 예시에서는 `create-sequence-store`의 사용 방법을 보여줍니다.

AWS CLI

시퀀스 저장소 생성

다음 `create-sequence-store` 예시에서는 시퀀스 저장소를 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 Storage를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateSequenceStore를 참조하세요.

create-share

다음 코드 예시에서는 create-share의 사용 방법을 보여줍니다.

AWS CLI

HealthOmics 분석 저장소의 공유 생성

다음 create-share 예시에서는 계정 외부의 구독자가 수락할 수 있는 HealthOmics 분석 저장소의 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 사용자 안내서의 교차 계정 공유를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateShare를 참조하세요.

create-variant-store

다음 코드 예시에서는 create-variant-store의 사용 방법을 보여줍니다.

AWS CLI

변형 저장소 생성

다음 create-variant-store 예시에서는 my_var_store라는 변형 저장소를 생성합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateVariantStore를 참조하세요.

create-workflow

다음 코드 예시에서는 create-workflow의 사용 방법을 보여줍니다.

AWS CLI

워크플로 생성

다음 create-workflow 예시에서는 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.zip은 워크플로 정의를 포함하는 ZIP 아카이브입니다. workflow-params.json은 워크플로의 런타임 파라미터를 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 AWS HealthOmics 사용 설명서의 Creating private workflows를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateWorkflow를 참조하세요.

delete-annotation-store-versions

다음 코드 예시에서는 delete-annotation-store-versions의 사용 방법을 보여줍니다.

AWS CLI

주석 저장소 버전 삭제

다음 delete-annotation-store-versions 예시에서는 주석 저장소 버전을 삭제합니다.

```
aws omics delete-annotation-store-versions \
    --name my_annotation_store \
    --versions my_version
```

출력:

```
{
    "errors": []
}
```

자세한 내용은 AWS HealthOmics 사용자 안내서의 새 버전의 주석 저장소 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteAnnotationStoreVersions를 참조하세요.

delete-annotation-store

다음 코드 예시에서는 delete-annotation-store의 사용 방법을 보여줍니다.

AWS CLI

주석 저장소 삭제

다음 delete-annotation-store 예시에서는 my_vcf_store라는 주석 저장소를 삭제합니다.

```
aws omics delete-annotation-store \
    --name my_vcf_store
```

출력:

```
{
    "status": "DELETING"
}
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics Analytics를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteAnnotationStore를 참조하세요.

delete-reference-store

다음 코드 예시에서는 delete-reference-store의 사용 방법을 보여줍니다.

AWS CLI

참조 저장소 삭제

다음 delete-reference-store 예시에서는 ID가 1234567890인 참조 저장소를 삭제합니다.

```
aws omics delete-reference-store \
    --id 1234567890
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics Storage를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteReferenceStore를 참조하세요.

delete-reference

다음 코드 예시에서는 delete-reference의 사용 방법을 보여줍니다.

AWS CLI

참조 삭제

다음 delete-reference 예시에서는 참조를 삭제합니다.

```
aws omics delete-reference \
    --reference-store-id 1234567890 \
    --id 1234567890
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics Storage를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteReference를 참조하세요.

delete-run-group

다음 코드 예시에서는 delete-run-group의 사용 방법을 보여줍니다.

AWS CLI

실행 그룹 삭제

다음 `delete-run-group` 예시에서는 ID가 1234567인 실행 그룹을 삭제합니다.

```
aws omics delete-run-group \
    --id 1234567
```

자세한 내용은 AWS HealthOmics 사용 설명서의 Deleting runs and run groups를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteRunGroup을 참조하세요.

delete-run

다음 코드 예시에서는 `delete-run`의 사용 방법을 보여줍니다.

AWS CLI

워크플로 실행 삭제

다음 `delete-run` 예시에서는 ID가 1234567인 실행을 삭제합니다.

```
aws omics delete-run \
    --id 1234567
```

자세한 내용은 AWS HealthOmics 사용 설명서의 Deleting runs and run groups를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteRun을 참조하세요.

delete-sequence-store

다음 코드 예시에서는 `delete-sequence-store`의 사용 방법을 보여줍니다.

AWS CLI

시퀀스 저장소 삭제

다음 `delete-sequence-store` 예시에서는 ID가 1234567890인 시퀀스 저장소를 삭제합니다.

```
aws omics delete-sequence-store \
```

```
    --id 1234567890
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics Storage를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteSequenceStore를 참조하세요.

delete-share

다음 코드 예시에서는 delete-share의 사용 방법을 보여줍니다.

AWS CLI

HealthOmics 분석 데이터의 공유 삭제

다음 delete-share 예시에서는 분석 데이터의 교차 계정 공유를 삭제합니다.

```
aws omics delete-share \
    --share-id "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a"
```

출력:

```
{
    "status": "DELETING"
}
```

자세한 내용은 AWS HealthOmics 사용자 안내서의 교차 계정 공유를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteShare를 참조하세요.

delete-variant-store

다음 코드 예시에서는 delete-variant-store의 사용 방법을 보여줍니다.

AWS CLI

변형 저장소 삭제

다음 delete-variant-store 예시에서는 my_var_store라는 변형 저장소를 삭제합니다.

```
aws omics delete-variant-store \
    --name my_var_store
```

출력:

```
{
    "status": "DELETING"
}
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics Analytics를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteVariantStore를 참조하세요.

delete-workflow

다음 코드 예시에서는 delete-workflow의 사용 방법을 보여줍니다.

AWS CLI

워크플로 삭제

다음 delete-workflow 예시에서는 ID가 1234567인 워크플로를 삭제합니다.

```
aws omics delete-workflow \
    --id 1234567
```

자세한 내용은 AWS HealthOmics 사용 설명서의 Delete a private workflow를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteWorkflow를 참조하세요.

get-annotation-import-job

다음 코드 예시에서는 get-annotation-import-job의 사용 방법을 보여줍니다.

AWS CLI

주석 가져오기 작업 보기

다음 get-annotation-import-job 예시에서는 주석 가져오기 작업의 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.

- API 세부 정보는 AWS CLI 명령 참조의 GetAnnotationImportJob을 참조하세요.

get-annotation-store-version

다음 코드 예시에서는 get-annotation-store-version의 사용 방법을 보여줍니다.

AWS CLI

주석 저장소 버전의 메타데이터 가져오기

다음 get-annotation-store-version 예시에서는 요청된 주석 저장소 버전의 메타데이터를 가져옵니다.

```
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 사용자 안내서의 새 버전의 주석 저장소 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetAnnotationStoreVersion을 참조하세요.

get-annotation-store

다음 코드 예시에서는 get-annotation-store의 사용 방법을 보여줍니다.

AWS CLI

주석 저장소 보기

다음 get-annotation-store 예시에서는 my_ann_store이라는 주석 저장소의 세부 정보를 가져옵니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 GetAnnotationStore를 참조하세요.

get-read-set-activation-job

다음 코드 예시에서는 get-read-set-activation-job의 사용 방법을 보여줍니다.

AWS CLI

읽기 세트 활성화 작업 보기

다음 get-read-set-activation-job 예시에서는 읽기 세트 활성화 작업의 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 Storage를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetReadSetActivationJob을 참조하세요.

get-read-set-export-job

다음 코드 예시에서는 get-read-set-export-job의 사용 방법을 보여줍니다.

AWS CLI

읽기 세트 내보내기 작업 보기

다음 get-read-set-export-job 예시에서는 읽기 세트 내보내기 작업의 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 Storage를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetReadSetExportJob을 참조하세요.

get-read-set-import-job

다음 코드 예시에서는 get-read-set-import-job의 사용 방법을 보여줍니다.

AWS CLI

읽기 세트 가져오기 작업 보기

다음 get-read-set-import-job 예시에서는 읽기 세트 가져오기 작업의 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 Storage를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetReadSetImportJob을 참조하세요.

`get-read-set-metadata`

다음 코드 예시에서는 `get-read-set-metadata`의 사용 방법을 보여줍니다.

AWS CLI

읽기 세트 보기

다음 `get-read-set-metadata` 예시에서는 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 Storage](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [GetReadSetMetadata](#)를 참조하세요.

get-read-set

다음 코드 예시에서는 get-read-set의 사용 방법을 보여줍니다.

AWS CLI

읽기 세트 다운로드

다음 get-read-set 예시에서는 읽기 세트의 파트 3을 1234567890.3.bam으로 다운로드합니다.

```
aws omics get-read-set \
    --sequence-store-id 1234567890 \
    --id 1234567890 \
    --part-number 3  1234567890.3.bam
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics Storage를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetReadSet를 참조하세요.

get-reference-import-job

다음 코드 예시에서는 get-reference-import-job의 사용 방법을 보여줍니다.

AWS CLI

참조 가져오기 작업 보기

다음 get-reference-import-job 예시에서는 참조 가져오기 작업의 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 Storage를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetReferenceImportJob을 참조하세요.

get-reference-metadata

다음 코드 예시에서는 get-reference-metadata의 사용 방법을 보여줍니다.

AWS CLI

참조 보기

다음 get-reference-metadata 예시에서는 참조의 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 Storage를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetReferenceMetadata를 참조하세요.

get-reference-store

다음 코드 예시에서는 get-reference-store의 사용 방법을 보여줍니다.

AWS CLI

참조 저장소 보기

다음 get-reference-store 예시에서는 참조 저장소의 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 Storage를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetReferenceStore를 참조하세요.

`get-reference`

다음 코드 예시에서는 `get-reference`의 사용 방법을 보여줍니다.

AWS CLI

유전체 참조 다운로드

다음 `get-reference` 예시에서는 유전체의 파트 1을 `hg38.1.fa`로 다운로드합니다.

```
aws omics get-reference \
    --reference-store-id 1234567890 \
    --id 1234567890 \
    --part-number 1 hg38.1.fa
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics Storage를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetReference를 참조하세요.

`get-run-group`

다음 코드 예시에서는 `get-run-group`의 사용 방법을 보여줍니다.

AWS CLI

실행 그룹 보기

다음 `get-run-group` 예시에서는 실행 그룹의 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 AWS HealthOmics 사용 설명서의 Creating run groups를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetRunGroup을 참조하세요.

get-run-task

다음 코드 예시에서는 get-run-task의 사용 방법을 보여줍니다.

AWS CLI

작업 보기

다음 get-run-task 예시에서는 워크플로 작업의 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 AWS HealthOmics 사용 설명서의 Task lifecycle in a HealthOmics run을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetRunTask를 참조하세요.

get-run

다음 코드 예시에서는 get-run의 사용 방법을 보여줍니다.

AWS CLI

워크플로 실행 보기

다음 `get-run` 예시에서는 워크플로 실행의 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 AWS HealthOmics 사용 설명서의 Run lifecycle in a workflow를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetRun을 참조하세요.

get-sequence-store

다음 코드 예시에서는 get-sequence-store의 사용 방법을 보여줍니다.

AWS CLI

시퀀스 저장소 보기

다음 get-sequence-store 예시에서는 ID가 1234567890인 시퀀스 저장소의 세부 정보를 가져옵니다.

```
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 Storage를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetSequenceStore를 참조하세요.

get-share

다음 코드 예시에서는 get-share의 사용 방법을 보여줍니다.

AWS CLI

HealthOmics 분석 데이터의 공유에 대한 메타데이터 가져오기

다음 get-share 예시에서는 분석 데이터의 교차 계정 공유에 대한 메타데이터를 가져옵니다.

```
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 사용자 안내서의 교차 계정 공유를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetShare를 참조하세요.

get-variant-import-job

다음 코드 예시에서는 get-variant-import-job의 사용 방법을 보여줍니다.

AWS CLI

변형 가져오기 작업 보기

다음 get-variant-import-job 예시에서는 변형 가져오기 작업의 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 Analytics를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetVariantImportJob을 참조하세요.

get-variant-store

다음 코드 예시에서는 get-variant-store의 사용 방법을 보여줍니다.

AWS CLI

변형 저장소 보기

다음 get-variant-store 예시에서는 변형 저장소의 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 Analytics를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetVariantStore를 참조하세요.

get-workflow

다음 코드 예시에서는 `get-workflow`의 사용 방법을 보여줍니다.

AWS CLI

워크플로 보기

다음 `get-workflow` 예시에서는 ID가 1234567인 워크플로의 세부 정보를 가져옵니다.

```
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 AWS HealthOmics 사용 설명서의 Creating private workflows를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetWorkflow를 참조하세요.

list-annotation-import-jobs

다음 코드 예시에서는 `list-annotation-import-jobs`의 사용 방법을 보여줍니다.

AWS CLI

주석 가져오기 작업 목록 가져오기

다음 `list-annotation-import-jobs`는 주석 가져오기 작업 목록을 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListAnnotationImportJobs를 참조하세요.

list-annotation-store-versions

다음 코드 예시에서는 list-annotation-store-versions의 사용 방법을 보여줍니다.

AWS CLI

주석 저장소의 모든 버전 나열

다음 list-annotation-store-versions 예시에서는 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 사용자 안내서의 새 버전의 주석 저장소 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListAnnotationStoreVersions를 참조하세요.

list-annotation-stores

다음 코드 예시에서는 `list-annotation-stores`의 사용 방법을 보여줍니다.

AWS CLI

주석 저장소 목록 가져오기

다음 `list-annotation-stores` 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListAnnotationStores를 참조하세요.

`list-multipart-read-set-uploads`

다음 코드 예시에서는 `list-multipart-read-set-uploads`의 사용 방법을 보여줍니다.

AWS CLI

모든 멀티파트 읽기 세트 업로드 및 해당 상태 나열

다음 `list-multipart-read-set-uploads` 예시에서는 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 사용자 안내서의 시퀀스 저장소에 직접 업로드를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListMultipartReadSetUploads를 참조하세요.

`list-read-set-activation-jobs`

다음 코드 예시에서는 `list-read-set-activation-jobs`의 사용 방법을 보여줍니다.

AWS CLI

읽기 세트 활성화 작업 목록 가져오기

다음 `list-read-set-activation-jobs` 예시에서는 ID가 1234567890인 시퀀스 저장소의 활성화 작업 목록을 가져옵니다.

```
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 Storage를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListReadSetActivationJobs를 참조하세요.

`list-read-set-export-jobs`

다음 코드 예시에서는 `list-read-set-export-jobs`의 사용 방법을 보여줍니다.

AWS CLI

읽기 세트 내보내기 작업 목록 가져오기

다음 `list-read-set-export-jobs` 예시에서는 ID가 1234567890인 시퀀스 저장소의 내보내기 작업 목록을 가져옵니다.

```
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 Storage를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListReadSetExportJobs를 참조하세요.

`list-read-set-import-jobs`

다음 코드 예시에서는 `list-read-set-import-jobs`의 사용 방법을 보여줍니다.

AWS CLI

읽기 세트 가져오기 작업 목록 가져오기

다음 `list-read-set-import-jobs` 예시에서는 ID가 `1234567890`인 시퀀스 저장소의 가져오기 작업 목록을 가져옵니다.

```
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 Storage를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListReadSetImportJobs를 참조하세요.

`list-read-set-upload-parts`

다음 코드 예시에서는 `list-read-set-upload-parts`의 사용 방법을 보여줍니다.

AWS CLI

시퀀스 저장소에 대해 요청된 멀티파트 업로드의 모든 부분 나열

다음 `list-read-set-upload-parts` 예시에서는 시퀀스 저장소에 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 사용자 안내서의 시퀀스 저장소에 직접 업로드를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListReadSetUploadParts를 참조하세요.

`list-read-sets`

다음 코드 예시에서는 `list-read-sets`의 사용 방법을 보여줍니다.

AWS CLI

읽기 세트 목록 가져오기

다음 `list-read-sets` 예시에서는 ID가 1234567890인 시퀀스 저장소의 읽기 세트 목록을 가져옵니다.

```
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 Storage를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListReadSets를 참조하세요.

`list-reference-import-jobs`

다음 코드 예시에서는 `list-reference-import-jobs`의 사용 방법을 보여줍니다.

AWS CLI

참조 가져오기 작업 목록 가져오기

다음 `list-reference-import-jobs` 예시에서는 ID가 1234567890인 참조 저장소의 참조 가져오기 작업 목록을 가져옵니다.

```
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 Storage를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListReferenceImportJobs를 참조하세요.

list-reference-stores

다음 코드 예시에서는 `list-reference-stores`의 사용 방법을 보여줍니다.

AWS CLI

참조 저장소 목록 가져오기

다음 `list-reference-stores` 예시에서는 참조 저장소 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 Storage를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListReferenceStores를 참조하세요.

list-references

다음 코드 예시에서는 `list-references`의 사용 방법을 보여줍니다.

AWS CLI

참조 목록 가져오기

다음 `list-references` 예시에서는 ID가 `1234567890`인 참조 저장소의 유전체 참조 목록을 가져옵니다.

```
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 Storage를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListReferences를 참조하세요.

list-run-groups

다음 코드 예시에서는 `list-run-groups`의 사용 방법을 보여줍니다.

AWS CLI

실행 그룹 목록 가져오기

다음 `list-run-groups` 예시에서는 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 AWS HealthOmics 사용 설명서의 Creating run groups를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListRunGroups를 참조하세요.

list-run-tasks

다음 코드 예시에서는 list-run-tasks의 사용 방법을 보여줍니다.

AWS CLI

작업 목록 가져오기

다음 list-run-tasks 예시에서는 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 AWS HealthOmics 사용 설명서의 Task lifecycle in a HealthOmics run을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ListRunTasks](https://docs.aws.amazon.com/cli/latest/reference/omics/list-run-tasks.html)를 참조하세요.

list-runs

다음 코드 예시에서는 `list-runs`의 사용 방법을 보여줍니다.

AWS CLI

워크플로 실행 목록 가져오기

다음 `list-runs` 예시에서는 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 AWS HealthOmics 사용 설명서의 Run lifecycle in a workflow를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListRuns를 참조하세요.

list-sequence-stores

다음 코드 예시에서는 list-sequence-stores의 사용 방법을 보여줍니다.

AWS CLI

시퀀스 저장소 목록 가져오기

다음 list-sequence-stores 예시에서는 시퀀스 저장소 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 Storage를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListSequenceStores를 참조하세요.

list-shares

다음 코드 예시에서는 list-shares의 사용 방법을 보여줍니다.

AWS CLI

HealthOmics 분석 데이터의 사용 가능한 공유 나열

다음 list-shares 예시에서는 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 사용자 안내서의 교차 계정 공유를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListShares를 참조하세요.

list-tags-for-resource

다음 코드 예시에서는 `list-tags-for-resource`의 사용 방법을 보여줍니다.

AWS CLI

태그 목록 가져오기

다음 `list-tags-for-resource` 예시에서는 ID가 `1234567`인 워크플로의 태그 목록을 가져옵니다.

```
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 Omics의 리소스 태그 지정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource를 참조하세요.

list-variant-import-jobs

다음 코드 예시에서는 `list-variant-import-jobs`의 사용 방법을 보여줍니다.

AWS CLI

변형 가져오기 작업 목록 가져오기

다음 `list-variant-import-jobs` 예시에서는 변형 가져오기 작업의 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 Analytics를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListVariantImportJobs를 참조하세요.

`list-variant-stores`

다음 코드 예시에서는 `list-variant-stores`의 사용 방법을 보여줍니다.

AWS CLI

변형 저장소 목록 가져오기

다음 `list-variant-stores` 예시에서는 변형 저장소 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListVariantStores를 참조하세요.

`list-workflows`

다음 코드 예시에서는 `list-workflows`의 사용 방법을 보여줍니다.

AWS CLI

워크플로 목록 가져오기

다음 `list-workflows` 예시에서는 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 AWS HealthOmics 사용 설명서의 Creating private workflows를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListWorkflows를 참조하세요.

start-annotation-import-job

다음 코드 예시에서는 `start-annotation-import-job`의 사용 방법을 보여줍니다.

AWS CLI

주석 가져오기

다음 `start-annotation-import-job` 예시에서는 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-W8O1XMPL7QZ \
    --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.

- API 세부 정보는 AWS CLI 명령 참조의 StartAnnotationImportJob을 참조하세요.

`start-read-set-activation-job`

다음 코드 예시에서는 `start-read-set-activation-job`의 사용 방법을 보여줍니다.

AWS CLI

아카이브된 읽기 세트 활성화

다음 `start-read-set-activation-job` 예시에서는 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 Storage를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartReadSetActivationJob을 참조하세요.

`start-read-set-export-job`

다음 코드 예시에서는 `start-read-set-export-job`의 사용 방법을 보여줍니다.

AWS CLI

읽기 세트 내보내기

다음 `start-read-set-export-job` 예시에서는 두 개의 읽기 세트를 Amazon S3으로 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 Storage를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartReadSetExportJob을 참조하세요.

start-read-set-import-job

다음 코드 예시에서는 start-read-set-import-job의 사용 방법을 보여줍니다.

AWS CLI

읽기 세트 가져오기

다음 start-read-set-import-job 예시에서는 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-
W8O1XMPL7QZ \
    --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-
W8O1XMPL7QZ",
    "sequenceStoreId": "1234567890",
```

```
    "status": "SUBMITTED"
}
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics Storage를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartReadSetImportJob을 참조하세요.

start-reference-import-job

다음 코드 예시에서는 `start-reference-import-job`의 사용 방법을 보여줍니다.

AWS CLI

참조 유전체 가져오기

다음 `start-reference-import-job` 예시에서는 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-
W8O1XMPL7QZ \
    --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-
W8O1XMPL7QZ",
    "status": "SUBMITTED"
}
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics Storage를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartReferenceImportJob을 참조하세요.

start-run

다음 코드 예시에서는 `start-run`의 사용 방법을 보여줍니다.

AWS CLI

워크플로 실행

다음 `start-run` 예시에서는 ID가 1234567인 워크플로를 실행합니다.

```
aws omics start-run \
    --workflow-id 1234567 \
    --role-arn arn:aws:iam::123456789012:role/omics-service-role-serviceRole-
W8O1XMPL7QZ \
    --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 AWS HealthOmics 사용 설명서의 Starting a run을 참조하세요.

Amazon Omics에서 소스 파일 로드

서비스별 URI를 사용하여 Amazon Omics Storage에서 소스 파일을 로드할 수도 있습니다. 다음 예시 workflow-inputs.json 파일은 읽기 세트 및 참조 유전체 소스에 Amazon Omics URI를 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 StartRun을 참조하세요.

start-variant-import-job

다음 코드 예시에서는 start-variant-import-job의 사용 방법을 보여줍니다.

AWS CLI

변형 파일 가져오기

다음 start-variant-import-job 예시에서는 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-W8O1XMPL7QZ \
    --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.

- API 세부 정보는 AWS CLI 명령 참조의 StartVariantImportJob을 참조하세요.

tag-resource

다음 코드 예시에서는 tag-resource의 사용 방법을 보여줍니다.

AWS CLI

리소스에 태그 지정

다음 tag-resource 예시에서는 ID가 1234567인 워크플로에 department 태그를 추가합니다.

```
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 Omics의 리소스 태그 지정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

untag-resource

다음 코드 예시에서는 untag-resource의 사용 방법을 보여줍니다.

AWS CLI

리소스에서 태그 제거

다음 untag-resource 예시에서는 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 Storage를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

update-annotation-store

다음 코드 예시에서는 update-annotation-store의 사용 방법을 보여줍니다.

AWS CLI

주석 저장소 업데이트

다음 update-annotation-store 예시에서는 my_vcf_store라는 주석 저장소에 대한 설명을 업데이트합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateAnnotationStore를 참조하세요.

update-run-group

다음 코드 예시에서는 update-run-group의 사용 방법을 보여줍니다.

AWS CLI

실행 그룹 업데이트

다음 update-run-group 예시에서는 ID가 1234567인 실행 그룹의 설정을 업데이트합니다.

```
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 Workflows를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateRunGroup을 참조하세요.

update-variant-store

다음 코드 예시에서는 update-variant-store의 사용 방법을 보여줍니다.

AWS CLI

변형 저장소 업데이트

다음 update-variant-store 예시에서는 my_var_store라는 변형 저장소에 대한 설명을 업데이트합니다.

```
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 Analytics를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateVariantStore를 참조하세요.

update-workflow

다음 코드 예시에서는 update-workflow의 사용 방법을 보여줍니다.

AWS CLI

워크플로 업데이트

다음 update-workflow 예시에서는 ID가 1234567인 워크플로에 대한 설명을 업데이트합니다.

```
aws omics update-workflow \
    --id 1234567 \
    --description "copy workflow"
```

자세한 내용은 AWS HealthOmics 사용 설명서의 Creating or updating a workflow를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateWorkflow를 참조하세요.

upload-read-set-part

다음 코드 예시에서는 upload-read-set-part의 사용 방법을 보여줍니다.

AWS CLI

읽기 세트 부분 업로드

다음 upload-read-set-part 예시에서는 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 사용자 안내서의 시퀀스 저장소에 직접 업로드를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UploadReadSetPart를 참조하세요.

AWS CLI를 사용한 IAM 예제

다음 코드 예제는 IAM과 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`add-client-id-to-open-id-connect-provider`

다음 코드 예시는 `add-client-id-to-open-id-connect-provider`의 사용 방법을 보여 줍니다.

AWS CLI

OIDC(Open-ID Connect) 제공업체에 클라이언트 ID(대상) 추가

다음 `add-client-id-to-open-id-connect-provider` 명령은 `server.example.com`이라는 OIDC 제공업체에게 클라이언트 ID `my-application-ID`를 추가합니다.

```
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 제공업체를 생성하려면 `create-open-id-connect-provider` 명령을 사용합니다.

자세한 내용은 AWS IAM 사용 설명서의 IAM에서 OIDC(OpenID Connect) ID 제공업체 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AddClientIdToOpenIdConnectProvider를 참조하세요.

`add-role-to-instance-profile`

다음 코드 예시는 `add-role-to-instance-profile`의 사용 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 설명서의 IAM 역할을 사용하여 Amazon EC2 인스턴스에서 실행되는 애플리케이션에 권한 부여를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AddRoleToInstanceProfile을 참조하세요.

`add-user-to-group`

다음 코드 예시는 `add-user-to-group`의 사용 방법을 보여 줍니다.

AWS CLI

IAM 그룹에 사용자 추가

다음 add-user-to-group 명령은 이름이 Admins인 IAM 그룹에 이름이 Bob인 IAM 사용자를 추가합니다.

```
aws iam add-user-to-group \
    --user-name Bob \
    --group-name Admins
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 IAM 사용자 그룹에서 사용자 추가 및 제거를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AddUserToGroup을 참조하세요.

attach-group-policy

다음 코드 예시는 attach-group-policy의 사용 방법을 보여 줍니다.

AWS CLI

IAM 그룹에 관리형 정책 연결

다음 attach-group-policy 명령은 이름이 Finance인 IAM 그룹에 이름이 ReadOnlyAccess인 AWS 관리형 정책을 연결합니다.

```
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 정책과 인라인 정책을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AttachGroupPolicy를 참조하세요.

attach-role-policy

다음 코드 예시는 attach-role-policy의 사용 방법을 보여 줍니다.

AWS CLI

IAM 역할에 관리형 정책 연결

다음 attach-role-policy 명령은 이름이 ReadOnlyRole인 IAM 역할에 이름이 ReadOnlyAccess인 AWS 관리형 정책을 연결합니다.

```
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 정책과 인라인 정책을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AttachRolePolicy를 참조하세요.

attach-user-policy

다음 코드 예시는 attach-user-policy의 사용 방법을 보여 줍니다.

AWS CLI

IAM 사용자에게 관리형 정책 연결

다음 attach-user-policy 명령은 이름이 Alice인 IAM 사용자에게 이름이 AdministratorAccess인 AWS 관리형 정책을 연결합니다.

```
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 정책과 인라인 정책을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AttachUserPolicy를 참조하세요.

change-password

다음 코드 예시는 change-password의 사용 방법을 보여 줍니다.

AWS CLI

IAM 사용자의 암호 변경

IAM 사용자의 암호를 변경하려면 `--cli-input-json` 파라미터를 사용하여 기존 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 입력하는 데 사용할 수 있는 change-password.json이라는 JSON 파일을 생성합니다. 예를 들어 파일은 다음과 같을 수 있습니다.

```
{
    "OldPassword": "3s0K_;xh4~8XXI",
    "NewPassword": "]35d/{pB9Fo9wJ"
}
```

다음으로 암호를 변경하려면 `change-password` 명령을 다시 사용하되 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 `InvalidUserType` 오류가 반환됩니다.

자세한 내용은 AWS IAM 사용 설명서의 [IAM 사용자가 자신의 암호를 변경하는 방법]을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ChangePassword] 섹션을 참조하세요.

`create-access-key`

다음 코드 예시는 `create-access-key`의 사용 방법을 보여 줍니다.

AWS CLI

IAM 사용자의 액세스 키 생성

다음 `create-access-key` 명령은 이름이 Bob인 IAM 사용자의 액세스 키(액세스 키 ID 및 비밀 액세스 키)를 생성합니다.

```
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 사용자의 액세스 키 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateAccessKey를 참조하세요.

`create-account-alias`

다음 코드 예시는 `create-account-alias`의 사용 방법을 보여 줍니다.

AWS CLI

계정 별칭 생성

다음 `create-account-alias` 명령은 AWS 계정의 별칭 `examplecorp`를 생성합니다.

```
aws iam create-account-alias \
    --account-alias examplecorp
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 AWS 계정 ID 및 별칭을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateAccountAlias를 참조하세요.

create-group

다음 코드 예시는 create-group의 사용 방법을 보여 줍니다.

AWS CLI

IAM 그룹을 생성하려면

다음 create-group 명령은 이름이 Admins인 IAM 그룹을 생성합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateGroup을 참조하세요.

create-instance-profile

다음 코드 예시는 create-instance-profile의 사용 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 설명서의 IAM 역할을 사용하여 Amazon EC2 인스턴스에서 실행되는 애플리케이션에 권한 부여를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateInstanceProfile을 참조하세요.

create-login-profile

다음 코드 예시는 create-login-profile의 사용 방법을 보여 줍니다.

AWS CLI

IAM 사용자의 암호 생성

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 입력하는 데 사용할 수 있는 create-login-profile.json이라는 JSON 파일을 생성합니다. 예:

```
{
    "UserName": "Bob",
    "Password": "&1-3a6u:RA0djs",
    "PasswordResetRequired": true
}
```

다음으로 IAM 사용자의 암호를 생성하려면 `create-login-profile` 명령을 다시 사용하되 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-profile`을 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 관리]를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [CreateLoginProfile](https://docs.aws.amazon.com/goto/aws-cli/iam-2010-05-08/CreateLoginProfile)을 참조하세요.

`create-open-id-connect-provider`

다음 코드 예시는 `create-open-id-connect-provider`의 사용 방법을 보여 줍니다.

AWS CLI

OIDC(OpenID Connect) 제공업체 생성

OIDC(OpenID Connect) 제공업체를 생성하려면 `--cli-input-json` 파라미터를 사용하여 필수 파라미터가 포함된 JSON 파일을 전달하는 것이 좋습니다. OIDC 제공업체를 생성할 때는 제공업체의 URL을 전달해야 하며 URL은 `https://`로 시작해야 합니다. 일부 명령줄 환경에서는 콜론(:)과 슬래시(/) 문자가 특별한 의미를 갖기 때문에 URL을 명령줄 파라미터로 전달하기 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 후속 `create-open-id-connect-provider` 명령에 대한 정보를 입력하는 데 사용할 수 있는 create-open-id-connect-provider.json이라는 JSON 파일을 생성합니다. 예:

```
{
    "Url": "https://server.example.com",
    "ClientIDList": [
        "example-application-ID"
    ],
    "ThumbprintList": [
        "c3768084dfb3d2b68b7897bf5f565da8eEXAMPLE"
    ]
}
```

다음으로, OIDC(OpenID Connect) 제공업체를 생성하려면 `create-open-id-connect-provider` 명령을 다시 사용하되 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 제공업체에 대한 자세한 내용은 AWS IAM 사용 설명서의 IAM에서 OIDC(OpenID Connect) ID 제공업체 생성을 참조하세요.

OIDC 제공업체의 지문 가져오기에 대한 자세한 내용은 AWS IAM 사용 설명서의 OpenID Connect 자격 증명 제공업체의 지문 얻기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateOpenIdConnectProvider를 참조하세요.

create-policy-version

다음 코드 예시는 create-policy-version의 사용 방법을 보여 줍니다.

AWS CLI

관리형 정책의 새 버전 생성

이 예제는 ARN이 arn:aws:iam::123456789012:policy/MyPolicy인 IAM 정책의 새 v2 버전을 생성하고 이를 기본 버전으로 만듭니다.

```
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 IAM 정책 버전 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreatePolicyVersion을 참조하세요.

create-policy

다음 코드 예시는 `create-policy`의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 고객 관리형 정책 생성

다음 명령은 이름이 `my-policy`인 고객 관리형 정책을 생성합니다. `policy.json` 파일은 이름이 `amzn-s3-demo-bucket`인 Amazon S3 버킷의 `shared` 폴더에 대한 읽기 전용 액세스 권한을 부여하는 현재 폴더의 JSON 문서입니다.

```
aws iam create-policy \
    --policy-name my-policy \
    --policy-document file://policy.json
```

policy.json의 콘텐츠:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:Get*",
                "s3:List*"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-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"
    }
}
```

문자열 파라미터의 입력으로 파일 사용에 대한 자세한 내용은 AWS CLI 사용 설명서의 AWS CLI에 파라미터 값 지정을 참조하세요.

예제 2: 설명이 포함된 고객 관리형 정책 생성

다음 명령은 변경 불가능한 설명이 포함된 이름이 `my-policy`인 고객 관리형 정책을 생성합니다.

`policy.json` 파일은 이름이 `amzn-s3-demo-bucket`인 Amazon S3 버킷의 모든 Put, List, Get 작업에 대한 액세스 권한을 부여하는 현재 폴더의 JSON 문서입니다.

```
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
 amzn-s3-demo-bucket"
```

policy.json의 콘텐츠:

```
{
   "Version": "2012-10-17",
   "Statement": [
       {
           "Effect": "Allow",
           "Action": [
                "s3:ListBucket*",
                "s3:PutBucket*",
                "s3:GetBucket*"
            ],
```

```
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-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"
    }
}
```

자격 증명 기반 정책에 대한 자세한 내용은 AWS IAM 사용 설명서의 [자격 증명 기반 정책 및 리소스 기반 정책]을 참조하세요.

예제 3: 태그가 포함된 고객 관리형 정책을 생성하는 방법

다음 명령은 태그가 포함된 이름이 `my-policy`인 고객 관리형 정책을 생성합니다. 이 예제에서는 JSON 형식의 태그(`'{"Key": "Department", "Value": "Accounting"}' '{"Key": "Location", "Value": "Seattle"}'`)와 함께 `--tags` 파라미터를 사용합니다. 또는 `--tags` 파라미터를 단축 형식의 태그(`'Key=Department,Value=Accounting Key=Location,Value=Seattle'`)와 함께 사용할 수도 있습니다.

`policy.json` 파일은 이름이 `amzn-s3-demo-bucket`인 Amazon S3 버킷의 모든 Put, List, Get 작업에 대한 액세스 권한을 부여하는 현재 폴더의 JSON 문서입니다.

```
aws iam create-policy \
    --policy-name my-policy \
    --policy-document file://policy.json \
```

```
    --tags '{"Key": "Department", "Value": "Accounting"}' '{"Key": "Location", 
"Value": "Seattle"}'
```

policy.json의 콘텐츠:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket*",
                "s3:PutBucket*",
                "s3:GetBucket*"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-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"
```

```
                {
            ]
        }
    }
```

정책 태그 지정에 대한 자세한 내용은 AWS IAM 사용 설명서의 고객 관리형 정책 태깅을 참조하세요.

• API 세부 정보는 AWS CLI 명령 참조의 CreatePolicy를 참조하세요.

create-role

다음 코드 예시는 create-role의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: IAM 역할 생성

다음 create-role 명령은 이름이 Test-Role인 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 파일에 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 시간을 포함한 IAM 역할 생성

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: 태그가 포함된 IAM 역할 생성

다음 명령은 태그가 포함된 IAM 역할 `Test-Role`을 생성합니다. 이 예제에서는 다음 JSON 형식의 태그(`'{"Key": "Department", "Value": "Accounting"}' '{"Key": "Location",` "Value": "Seattle"}')에 --tags 파라미터 플래그를 사용합니다. 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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateRole을 참조하세요.

create-saml-provider

다음 코드 예시는 `create-saml-provider`의 사용 방법을 보여 줍니다.

AWS CLI

SAML 공급자 생성

이 예제는 IAM에 이름이 `MySAMLProvider`인 새 SAML 공급자를 생성합니다. `SAMLMetaData.xml` 파일에 있는 SAML 메타데이터 문서에 설명되어 있습니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateSAMLProvider를 참조하세요.

create-service-linked-role

다음 코드 예시는 `create-service-linked-role`의 사용 방법을 보여 줍니다.

AWS CLI

서비스 연결 역할 생성

다음 `create-service-linked-role` 예제에서는 지정된 AWS 서비스에 대한 서비스 연결 역할을 생성하고 지정된 설명을 첨부합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateServiceLinkedRole을 참조하세요.

`create-service-specific-credential`

다음 코드 예시는 `create-service-specific-credential`의 사용 방법을 보여 줍니다.

AWS CLI

사용자에 대한 서비스별 보안 인증 정보 세트 생성

다음 `create-service-specific-credential` 예시에서는 구성된 서비스에만 액세스할 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 Create Git credentials for HTTPS connections to CodeCommit 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateServiceSpecificCredential 섹션을 참조하세요.

`create-user`

다음 코드 예시는 `create-user`의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: IAM 사용자 생성

다음 `create-user` 명령은 현재 계정에서 이름이 Bob인 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 경로에 IAM 사용자 생성

다음 `create-user` 명령은 지정된 경로에서 이름이 Bob인 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 포함된 IAM 사용자 생성

다음 `create-user` 명령은 태그가 포함된 이름이 Bob인 IAM 사용자를 생성합니다. 이 예제에서는 다음 JSON 형식의 태그(`'{"Key": "Department", "Value": "Accounting"}'`

'{"Key": "Location", "Value": "Seattle"}')에 --tags 파라미터 플래그를 사용합니다. 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 태깅을 참조하세요.

예제 3: 권한 경계가 설정된 IAM 사용자 생성

다음 create-user 명령은 Amazons3FullAccess의 권한 경계가 포함되어 있으며 이름이 Bob인 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 정보는 AWS IAM 사용 설명서의 IAM 엔터티의 권한 범위를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateUser를 참조하세요.

create-virtual-mfa-device

다음 코드 예시는 create-virtual-mfa-device의 사용 방법을 보여 줍니다.

AWS CLI

가상 MFA 디바이스 생성

이 예제는 BobsMFADevice라는 새 가상 MFA 디바이스를 생성합니다. 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 AWS에서 멀티 팩터 인증(MFA) 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateVirtualMfaDevice를 참조하세요.

deactivate-mfa-device

다음 코드 예시는 deactivate-mfa-device의 사용 방법을 보여 줍니다.

AWS CLI

MFA 디바이스 비활성화

이 명령은 사용자 Bob과 연결된 ARN arn:aws:iam::210987654321:mfa/BobsMFADevice의 가상 MFA 디바이스를 비활성화합니다.

```
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 AWS에서 멀티 팩터 인증(MFA) 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeactivateMfaDevice를 참조하세요.

decode-authorization-message

다음 코드 예시는 decode-authorization-message의 사용 방법을 보여 줍니다.

AWS CLI

권한 부여 실패 메시지를 디코딩하는 방법

다음 decode-authorization-message 예시에서는 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-
```

```
_PE0OaUqHCCcsSrFtvxm6yQD1nbm6VTIVrfa0Bzy8lsoMo7SjIaJ2r5vph6SY5vCCwg6o2JKe3hIHTa8zRrDbZSFMkcX
Xx9AYAAIr6bhcis7C__bZh4dlAAWooHFGKgfoJcWGwgdzgbu9hWyVvKTpeot5hsb8qANYjJRCPXTKpi6PZfdijIkwb6g
```

출력은 모든 JSON 텍스트 프로세서로 구문 분석할 수 있는 한 줄의 JSON 텍스트 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:Post의 How can I decode an authorization failure message after receiving an "UnauthorizedOperation" error during an EC2 instance launch?를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DecodeAuthorizationMessage를 참조하세요.

delete-access-key

다음 코드 예시는 delete-access-key의 사용 방법을 보여 줍니다.

AWS CLI

IAM 사용자의 액세스 키 삭제

다음 delete-access-key 명령은 이름이 Bob인 IAM 사용자의 지정된 액세스 키(액세스 키 ID 및 비밀 액세스 키)를 삭제합니다.

```
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 사용자의 액세스 키 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteAccessKey를 참조하세요.

delete-account-alias

다음 코드 예시는 delete-account-alias의 사용 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 별칭을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteAccountAlias를 참조하세요.

delete-account-password-policy

다음 코드 예시는 delete-account-password-policy의 사용 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 사용자의 계정 암호 정책 설정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteAccountPasswordPolicy를 참조하세요.

delete-group-policy

다음 코드 예시는 delete-group-policy의 사용 방법을 보여 줍니다.

AWS CLI

IAM 그룹에서 정책 삭제

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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteGroupPolicy를 참조하세요.

delete-group

다음 코드 예시는 delete-group의 사용 방법을 보여 줍니다.

AWS CLI

IAM 그룹 삭제

다음 delete-group 명령은 이름이 MyTestGroup인 IAM 그룹을 삭제합니다.

```
aws iam delete-group \
```

```
    --group-name MyTestGroup
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 IAM 사용자 그룹 삭제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteGroup을 참조하세요.

delete-instance-profile

다음 코드 예시는 delete-instance-profile의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteInstanceProfile을 참조하세요.

delete-login-profile

다음 코드 예시는 delete-login-profile의 사용 방법을 보여 줍니다.

AWS CLI

IAM 사용자의 암호 삭제

다음 delete-login-profile 명령은 이름이 Bob인 IAM 사용자의 암호를 삭제합니다.

```
aws iam delete-login-profile \
    --user-name Bob
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 IAM 사용자 암호 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteLoginProfile을 참조하세요.

delete-open-id-connect-provider

다음 코드 예시는 delete-open-id-connect-provider의 사용 방법을 보여 줍니다.

AWS CLI

IAM OpenID Connect ID 제공업체 삭제

이 예제는 제공업체 example.oidcprovider.com에 연결되는 IAM OIDC 제공업체를 삭제합니다.

```
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 IAM에서 OIDC(OpenID Connect) ID 제공업체 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteOpenIdConnectProvider를 참조하세요.

delete-policy-version

다음 코드 예시는 delete-policy-version의 사용 방법을 보여 줍니다.

AWS CLI

관리형 정책의 버전 삭제

이 예제는 ARN이 arn:aws:iam::123456789012:policy/MySamplePolicy인 정책에서 v2로 식별된 버전을 삭제합니다.

```
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 IAM의 정책 및 권한을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeletePolicyVersion을 참조하세요.

delete-policy

다음 코드 예시는 delete-policy의 사용 방법을 보여 줍니다.

AWS CLI

IAM 정책 삭제

이 예제에서는 ARN이 arn:aws:iam::123456789012:policy/MySamplePolicy인 정책을 삭제합니다.

```
aws iam delete-policy \
    --policy-arn arn:aws:iam::123456789012:policy/MySamplePolicy
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 IAM의 정책 및 권한을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeletePolicy를 참조하세요.

delete-role-permissions-boundary

다음 코드 예시는 delete-role-permissions-boundary의 사용 방법을 보여 줍니다.

AWS CLI

IAM 역할에서 권한 경계 삭제

다음 delete-role-permissions-boundary 예제는 지정된 IAM 역할의 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 IAM의 정책 및 권한을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteRolePermissionsBoundary를 참조하세요.

delete-role-policy

다음 코드 예시는 delete-role-policy의 사용 방법을 보여 줍니다.

AWS CLI

IAM 역할에서 정책 제거

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 변경을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteRolePolicy를 참조하세요.

delete-role

다음 코드 예시는 delete-role의 사용 방법을 보여 줍니다.

AWS CLI

IAM 역할 삭제

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.

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteRole]을 참조하세요.

delete-saml-provider

다음 코드 예시는 delete-saml-provider의 사용 방법을 보여 줍니다.

AWS CLI

SAML 공급자 삭제

이 예제에서는 ARN이 arn:aws:iam::123456789012:saml-provider/SAMLADFSProvider인 IAM SAML 2.0 공급자를 삭제합니다.

```
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 생성]을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteSAMLProvider]를 참조하세요.

delete-server-certificate

다음 코드 예시는 delete-server-certificate의 사용 방법을 보여 줍니다.

AWS CLI

AWS 계정에서 서버 인증서 삭제

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 사용 가능한 서버 인증서를 나열하려면 list-server-certificates 명령을 사용합니다.

자세한 내용은 AWS IAM 사용 설명서의 [IAM에서 서버 인증서 관리]를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteServerCertificate를 참조하세요.

delete-service-linked-role

다음 코드 예시는 delete-service-linked-role의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteServiceLinkedRole을 참조하세요.

delete-service-specific-credential

다음 코드 예시는 delete-service-specific-credential의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 요청 사용자의 서비스별 자격 증명 삭제

다음 delete-service-specific-credential 예시에서는 요청을 수행하는 사용자의 지정된 서비스별 자격 증명을 삭제합니다. service-specific-credential-id는 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.

예시 2: 요청 사용자의 서비스별 자격 증명 삭제

다음 `delete-service-specific-credential` 예시에서는 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 Create Git credentials for HTTPS connections to CodeCommit 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteServiceSpecificCredential 섹션을 참조하세요.

`delete-signing-certificate`

다음 코드 예시는 `delete-signing-certificate`의 사용 방법을 보여 줍니다.

AWS CLI

IAM 사용자의 서명 인증서 삭제

다음 `delete-signing-certificate` 명령은 `Bob`이라는 IAM 사용자에 대해 지정된 서명 인증서를 삭제합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteSigningCertificate를 참조하세요.

delete-ssh-public-key

다음 코드 예시는 delete-ssh-public-key의 사용 방법을 보여 줍니다.

AWS CLI

IAM 사용자에게 연결된 SSH 퍼블릭 키를 삭제하는 방법

다음 delete-ssh-public-key 명령은 IAM 사용자 sofia에 연결된 지정된 SSH 퍼블릭 키를 삭제합니다.

```
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 CodeCommit에 SSH 키 및 SSH 사용 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteSshPublicKey 섹션을 참조하세요.

delete-user-permissions-boundary

다음 코드 예시는 delete-user-permissions-boundary의 사용 방법을 보여 줍니다.

AWS CLI

IAM 사용자에게서 권한 경계 삭제

다음 delete-user-permissions-boundary 예제는 intern이라는 IAM 사용자에게 연결된 권한 경계를 삭제합니다. 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 IAM의 정책 및 권한을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteUserPermissionsBoundary를 참조하세요.

delete-user-policy

다음 코드 예시는 delete-user-policy의 사용 방법을 보여 줍니다.

AWS CLI

IAM 사용자에서 정책 제거

다음 delete-user-policy 명령은 이름이 Bob인 IAM 사용자에서 지정된 정책을 제거합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteUserPolicy를 참조하세요.

delete-user

다음 코드 예시는 delete-user의 사용 방법을 보여 줍니다.

AWS CLI

IAM 사용자 삭제

다음 delete-user 명령은 현재 계정에서 이름이 Bob인 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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteUser를 참조하세요.

delete-virtual-mfa-device

다음 코드 예시는 delete-virtual-mfa-device의 사용 방법을 보여 줍니다.

AWS CLI

가상 MFA 디바이스 제거

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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteVirtualMfaDevice를 참조하세요.

detach-group-policy

다음 코드 예시는 `detach-group-policy`의 사용 방법을 보여 줍니다.

AWS CLI

그룹에서 정책 분리

이 예제는 `Testers`라는 그룹에서 ARN `arn:aws:iam::123456789012:policy/TesterAccessPolicy`가 있는 관리형 정책을 제거합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 DetachGroupPolicy를 참조하세요.

detach-role-policy

다음 코드 예시는 `detach-role-policy`의 사용 방법을 보여 줍니다.

AWS CLI

역할에서 정책 분리

이 예제에서는 FedTesterRole을 호출한 역할에서 ARN이 arn:aws:iam::123456789012:policy/FederatedTesterAccessPolicy인 관리형 정책을 제거합니다.

```
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 변경을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DetachRolePolicy를 참조하세요.

detach-user-policy

다음 코드 예시는 detach-user-policy의 사용 방법을 보여 줍니다.

AWS CLI

사용자에서 정책 분리

이 예제에서는 사용자 Bob에서 ARN이 arn:aws:iam::123456789012:policy/TesterPolicy인 관리형 정책을 제거합니다.

```
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 사용자의 권한 변경을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DetachUserPolicy를 참조하세요.

disable-organizations-root-credentials-management

다음 코드 예시는 disable-organizations-root-credentials-management의 사용 방법을 보여 줍니다.

AWS CLI

조직에서 RootCredentialsManagement 기능을 비활성화하려면

다음 `disable-organizations-root-credentials-management` 명령은 조직의 멤버 계정에서 권한 있는 루트 사용자 자격 증명의 관리를 비활성화합니다.

```
aws iam disable-organizations-root-credentials-management
```

출력:

```
{
    "EnabledFeatures": [
        "RootSessions"
    ]
    "OrganizationId": "o-aa111bb222"
}
```

자세한 내용은 AWS IAM 사용 설명서의 [멤버 계정에 대한 루트 액세스 중앙 집중화](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DisableOrganizationsRootCredentialsManagement](#) 섹션을 참조하세요.

`disable-organizations-root-sessions`

다음 코드 예시는 `disable-organizations-root-sessions`의 사용 방법을 보여 줍니다.

AWS CLI

조직에서 RootSessions 기능을 비활성화하려면

다음 `disable-organizations-root-sessions` 명령은 조직의 멤버 계정 전체에서 권한이 있는 작업에 대한 루트 사용자 세션을 비활성화합니다.

```
aws iam disable-organizations-root-sessions
```

출력:

```
{
    "EnabledFeatures": [
        "RootCredentialsManagement"
```

```
    ]
    "OrganizationId": "o-aa111bb222"
}
```

자세한 내용은 AWS IAM 사용 설명서의 멤버 계정에 대한 루트 액세스 중앙 집중화를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisableOrganizationsRootSessions를 참조하세요.

enable-mfa-device

다음 코드 예시는 enable-mfa-device의 사용 방법을 보여 줍니다.

AWS CLI

MFA 디바이스 활성화

create-virtual-mfa-device 명령을 사용하여 새 가상 MFA 디바이스를 생성한 후 사용자에게 MFA 디바이스를 할당할 수 있습니다. 다음 enable-mfa-device 예제는 일련 번호가 arn:aws:iam::210987654321:mfa/BobsMFADevice인 MFA 디바이스를 사용자 Bob에게 할당합니다. 또한 이 명령은 가상 MFA 디바이스의 처음 두 코드를 순서대로 포함하여 디바이스를 AWS와 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 멀티 팩터 인증(MFA) 디바이스 활성화를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 EnableMfaDevice를 참조하세요.

enable-organizations-root-credentials-management

다음 코드 예시는 enable-organizations-root-credentials-management의 사용 방법을 보여 줍니다.

AWS CLI

조직에서 RootCredentialsManagement 기능을 활성화하려면

다음 `enable-organizations-root-credentials-management` 명령은 조직의 멤버 계정에서 권한 있는 루트 사용자 자격 증명의 관리를 활성화합니다.

```
aws iam enable-organizations-root-credentials-management
```

출력:

```
{
    "EnabledFeatures": [
        "RootCredentialsManagement"
    ]
    "OrganizationId": "o-aa111bb222"
}
```

자세한 내용은 AWS IAM 사용 설명서의 멤버 계정에 대한 루트 액세스 중앙 집중화를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 EnableOrganizationsRootCredentialsManagement 섹션을 참조하세요.

enable-organizations-root-sessions

다음 코드 예시는 `enable-organizations-root-sessions`의 사용 방법을 보여 줍니다.

AWS CLI

조직에서 RootSessions 기능을 활성화하려면

다음 `enable-organizations-root-sessions` 명령을 사용하면 관리 계정 또는 위임된 관리자가 조직의 멤버 계정에서 권한 있는 작업을 수행할 수 있습니다.

```
aws iam enable-organizations-root-sessions
```

출력:

```
{
    "EnabledFeatures": [
        "RootSessions"
    ]
    "OrganizationId": "o-aa111bb222"
}
```

자세한 내용은 AWS IAM 사용 설명서의 멤버 계정에 대한 루트 액세스 중앙 집중화를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 EnableOrganizationsRootSessions 섹션을 참조하세요.

generate-credential-report

다음 코드 예시는 generate-credential-report의 사용 방법을 보여 줍니다.

AWS CLI

보안 인증 보고서 생성

다음 예제에서는 AWS 계정에 대해 보안 인증 보고서 생성을 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 계정의 보안 인증 보고서 가져오기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GenerateCredentialReport를 참조하세요.

generate-organizations-access-report

다음 코드 예시는 generate-organizations-access-report의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 조직의 루트에 대한 액세스 보고서 생성

다음 generate-organizations-access-report 예시에서는 백그라운드 작업을 시작하여 조직의 지정된 루트에 대한 액세스 보고서를 생성합니다. 보고서가 생성된 후에는 get-organizations-access-report 명령을 실행하여 보고서를 표시할 수 있습니다.

```
aws iam generate-organizations-access-report \
    --entity-path o-4fxmplt198/r-c3xb
```

출력:

```
{
    "JobId": "a8b6c06f-aaa4-8xmp-28bc-81da71836359"
}
```

예시 2: 조직의 계정에 대한 액세스 보고서 생성

다음 `generate-organizations-access-report` 예시에서는 백그라운드 작업을 시작하여 조직의 계정 ID123456789012에 대한 액세스 보고서를 생성합니다`o-4fxmplt198`. 보고서가 생성된 후에는 `get-organizations-access-report` 명령을 실행하여 보고서를 표시할 수 있습니다.

```
aws iam generate-organizations-access-report \
    --entity-path o-4fxmplt198/r-c3xb/123456789012
```

출력:

```
{
    "JobId": "14b6c071-75f6-2xmp-fb77-faf6fb4201d2"
}
```

예시 3: 조직의 계정에 대한 액세스 보고서 생성

다음 `generate-organizations-access-report` 예시에서는 백그라운드 작업을 시작하여 `o-4fxmplt198` 조직 내 조직 단위 `ou-c3xb-lmu7j2yg`의 계정 ID 234567890123에 대한 액세스 보고서를 생성합니다. 보고서가 생성된 후에는 `get-organizations-access-report` 명령을 실행하여 보고서를 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 사용하여 AWS에서의 권한 재정의]를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [GenerateOrganizationsAccessReport] 섹션을 참조하세요.

generate-service-last-accessed-details

다음 코드 예시는 generate-service-last-accessed-details의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 사용자 지정 정책에 대한 서비스 액세스 보고서 생성

다음 generate-service-last-accessed-details 예제는 백그라운드 작업을 시작하여 intern-boundary라는 사용자 지정 정책으로 IAM 사용자와 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::123456789012:policy/intern-boundary
```

출력:

```
{
    "JobId": "2eb6c2b8-7b4c-3xmp-3c13-03b72c8cdfdc"
}
```

예제 2: AWS 관리형 AdministratorAccess 정책에 대한 서비스 액세스 보고서 생성

다음 generate-service-last-accessed-details 예제는 백그라운드 작업을 시작하여 AWS 관리형 AdministratorAccess 정책으로 IAM 사용자와 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 사용하여 AWS에서의 권한 재정의를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GenerateServiceLastAccessedDetails를 참조하세요.

get-access-key-last-used

다음 코드 예시는 `get-access-key-last-used`의 사용 방법을 보여 줍니다.

AWS CLI

지정된 액세스 키가 마지막으로 사용된 경우에 대한 정보 검색

다음 예제에서는 액세스 키 `ABCDEXAMPLE`이 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 사용자의 액세스 키 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetAccessKeyLastUsed를 참조하세요.

get-account-authorization-details

다음 코드 예시는 `get-account-authorization-details`의 사용 방법을 보여 줍니다.

AWS CLI

AWS 계정의 IAM 사용자, 그룹, 역할 및 정책을 나열하는 방법

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:::amzn-s3-demo-bucket",
                                    "arn:aws:s3:::amzn-s3-demo-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.

- API 세부 정보는 AWS CLI 명령 참조의 GetAccountAuthorizationDetails를 참조하세요.

`get-account-password-policy`

다음 코드 예시는 `get-account-password-policy`의 사용 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 사용자의 계정 암호 정책 설정]을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [GetAccountPasswordPolicy]를 참조하세요.

`get-account-summary`

다음 코드 예시는 `get-account-summary`의 사용 방법을 보여 줍니다.

AWS CLI

현재 계정의 IAM 엔터티 사용량 및 IAM 할당량에 대한 정보 가져오기

다음 `get-account-summary` 명령은 계정의 현재 IAM 엔터티 사용량과 현재 IAM 엔터티 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.

- API 세부 정보는 AWS CLI 명령 참조의 GetAccountSummary를 참조하세요.

get-context-keys-for-custom-policy

다음 코드 예시는 get-context-keys-for-custom-policy의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 명령줄에서 파라미터로 제공된 하나 이상의 사용자 지정 JSON 정책에서 참조하는 컨텍스트 키 나열

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-policy`를 성공적으로 사용하기 위해 제공해야 하는 컨텍스트 키 값을 식별합니다. `get-context-keys-for-custom-policy` 명령을 사용하여 IAM 사용자 또는 역할과 관련된 모든 정책에서 사용되는 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 키 나열

다음 `get-context-keys-for-custom-policy` 명령은 정책이 파라미터 대신 파일로 제공된다는 점을 제외하면 이전 예와 동일합니다. 명령에는 JSON 구조 목록이 아닌 JSON 문자열 목록이 필요하므로 파일은 하나로 축소할 수 있지만 다음과 같이 구성되어야 합니다.

```
[
    "Policy1",
    "Policy2"
]
```

예를 들어 이전 예제의 정책이 포함된 파일은 다음과 같아야 합니다. 정책 문자열 내에 포함된 각 큰따옴표는 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 시뮬레이터의 사용(AWS CLI 및 AWS API)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetContextKeysForCustomPolicy를 참조하세요.

`get-context-keys-for-principal-policy`

다음 코드 예시는 `get-context-keys-for-principal-policy`의 사용 방법을 보여 줍니다.

AWS CLI

IAM 보안 주체와 연결된 모든 정책에서 참조하는 컨텍스트 키 나열

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 임의의 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 시뮬레이터의 사용(AWS CLI 및 AWS API)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetContextKeysForPrincipalPolicy를 참조하세요.

get-credential-report

다음 코드 예시는 get-credential-report의 사용 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 계정의 보안 인증 보고서 가져오기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetCredentialReport를 참조하세요.

get-group-policy

다음 코드 예시는 get-group-policy의 사용 방법을 보여 줍니다.

AWS CLI

IAM 그룹에 연결된 정책에 대한 정보 가져오기

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.

- API 세부 정보는 AWS CLI 명령 참조의 GetGroupPolicy를 참조하세요.

get-group

다음 코드 예시는 get-group의 사용 방법을 보여 줍니다.

AWS CLI

IAM 그룹 가져오기

이 예제는 IAM 그룹 Admins에 대한 세부 정보를 반환합니다.

```
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 ID(사용자, 그룹 및 역할)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetGroup을 참조하세요.

get-instance-profile

다음 코드 예시는 `get-instance-profile`의 사용 방법을 보여 줍니다.

AWS CLI

인스턴스 프로파일 정보 가져오기

다음 `get-instance-profile` 명령은 이름이 `ExampleInstanceProfile`인 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.

- API 세부 정보는 AWS CLI 명령 참조의 GetInstanceProfile을 참조하세요.

get-login-profile

다음 코드 예시는 get-login-profile의 사용 방법을 보여 줍니다.

AWS CLI

IAM 사용자의 암호 정보 가져오기

다음 get-login-profile 명령은 이름이 Bob인 IAM 사용자의 암호에 대한 정보를 가져옵니다.

```
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 관리]를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [GetLoginProfile]을 참조하세요.

get-mfa-device

다음 코드 예시는 get-mfa-device의 사용 방법을 보여 줍니다.

AWS CLI

FIDO 보안 키에 대한 정보를 검색하는 방법

다음 get-mfa-device 명령 예시에서는 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 [AWS에서 멀티 팩터 인증(MFA) 사용]을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [GetMfaDevice] 섹션을 참조하세요.

get-open-id-connect-provider

다음 코드 예시는 get-open-id-connect-provider의 사용 방법을 보여 줍니다.

AWS CLI

지정된 OpenID Connect 제공업체에 대한 정보 반환

이 예제는 ARN이 `arn:aws:iam::123456789012:oidc-provider/server.example.com`인 OpenID Connect 제공업체에 대한 세부 정보를 반환합니다.

```
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 IAM에서 OIDC(OpenID Connect) ID 제공업체 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetOpenIdConnectProvider를 참조하세요.

get-organizations-access-report

다음 코드 예시는 `get-organizations-access-report`의 사용 방법을 보여 줍니다.

AWS CLI

서비스 액세스 보고서 검색

다음 `get-organizations-access-report` 예시에서는 AWS Organizations 엔터티에 대해 이전에 생성된 액세스 보고서를 표시합니다. 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 사용하여 AWS에서의 권한 재정의를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetOrganizationsAccessReport 섹션을 참조하세요.

`get-policy-version`

다음 코드 예시는 `get-policy-version`의 사용 방법을 보여 줍니다.

AWS CLI

지정된 관리형 정책의 지정된 버전에 대한 정보 검색

이 예제는 ARN이 `arn:aws:iam::123456789012:policy/MyManagedPolicy`인 정책의 v2 버전에 대한 정책 문서를 반환합니다.

```
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 IAM의 정책 및 권한을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetPolicyVersion을 참조하세요.

get-policy

다음 코드 예시는 get-policy의 사용 방법을 보여 줍니다.

AWS CLI

지정된 관리형 정책에 대한 정보 검색

이 예제는 ARN이 arn:aws:iam::123456789012:policy/MySamplePolicy인 관리형 정책에 대한 세부 정보를 반환합니다.

```
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 IAM의 정책 및 권한을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetPolicy를 참조하세요.

get-role-policy

다음 코드 예시는 get-role-policy의 사용 방법을 보여 줍니다.

AWS CLI

IAM 역할에 연결된 정책에 대한 정보 가져오기

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 생성](#)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [GetRolePolicy](#)를 참조하세요.

get-role

다음 코드 예시는 `get-role`의 사용 방법을 보여 줍니다.

AWS CLI

IAM 역할 정보 가져오기

다음 `get-role` 명령은 이름이 `Test-Role`인 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.

- API 세부 정보는 AWS CLI 명령 참조의 GetRole을 참조하세요.

get-saml-provider

다음 코드 예시는 `get-saml-provider`의 사용 방법을 보여 줍니다.

AWS CLI

SAML 제공업체 메타문서 검색

이 예제는 ARM이 `arn:aws:iam::123456789012:saml-provider/SAMLADFS`인 SAML 2.0 제공업체에 대한 세부 정보를 검색합니다. 응답에는 AWS SAML 제공업체 엔터티를 생성하기 위해 ID 제공업체로부터 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.

- API 세부 정보는 AWS CLI 명령 참조의 GetSamlProvider를 참조하세요.

get-server-certificate

다음 코드 예시는 `get-server-certificate`의 사용 방법을 보여 줍니다.

AWS CLI

AWS 계정의 서버 인증서에 대한 세부 정보 가져오기

다음 `get-server-certificate` 명령은 AWS 계정의 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
                HzAdBgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wgZ8wDQYJKoZIhvcNAQE
                BBQADgY0AMIGJAoGBAMaK0dn+a4GmWIgWJ21uUSfwfEvySWtC2XADZ4nB+BLYgVI
                k60CpiwsZ3G93vUEIO3IyNoH/f0wYK8mh9TrDHudUZg3qX4waLG5M43q7Wgc/MbQ
                ITxOUSQv7c7ugFFDzQGBzZswY6786m86gjpEIbb3OhjZnzcvQAaRHhdlQWIMm2nr
                AgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCku4nUhVVxYUntneD9+h8Mg9q6q+auN
                KyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0FlkbFFBjvSfpJIlJ00zbhNYS5f6Guo
                EDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjS;TbNYiytVbZPQUQ5Yaxu2jXnimvw
                3rrszlaEWEG5vb25lQGFtsYXpvbiEXAMPLE=\n-----END CERTIFICATE-----"
        }
}
```

AWS 계정에서 사용 가능한 서버 인증서를 나열하려면 `list-server-certificates` 명령을 사용합니다.

자세한 내용은 AWS IAM 사용 설명서의 IAM에서 서버 인증서 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetServerCertificate를 참조하세요.

`get-service-last-accessed-details-with-entities`

다음 코드 예시는 `get-service-last-accessed-details-with-entities`의 사용 방법을 보여줍니다.

AWS CLI

서비스 세부 정보가 포함된 서비스 액세스 보고서 검색

다음 `get-service-last-accessed-details-with-entities` 예제는 지정된 서비스에 액세스한 IAM 사용자와 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 사용하여 AWS에서의 권한 재정의를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetServiceLastAccessedDetailsWithEntities를 참조하세요.

get-service-last-accessed-details

다음 코드 예시는 `get-service-last-accessed-details`의 사용 방법을 보여 줍니다.

AWS CLI

서비스 액세스 보고서 검색

다음 get-service-last-accessed-details 예제는 IAM 엔터티가 액세스한 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 사용하여 AWS에서의 권한 재정의를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetServiceLastAccessedDetails를 참조하세요.

get-service-linked-role-deletion-status

다음 코드 예시는 get-service-linked-role-deletion-status의 사용 방법을 보여 줍니다.

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.

- API 세부 정보는 AWS CLI 명령 참조의 GetServiceLinkedRoleDeletionStatus를 참조하세요.

get-ssh-public-key

다음 코드 예시는 get-ssh-public-key의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: SSH 인코딩된 양식으로 IAM 사용자에게 연결된 SSH 퍼블릭 키 검색

다음 get-ssh-public-key 명령은 IAM 사용자 sofia에서 지정된 SSH 퍼블릭 키를 검색합니다. 출력은 SSH 인코딩에 있습니다.

```
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"
    }
}
```

예시 2: PEM 인코딩된 형식으로 IAM 사용자에게 연결된 SSG 퍼블릭 키 검색

다음 `get-ssh-public-key` 명령은 IAM 사용자 `sofia`에서 지정된 SSH 퍼블릭 키를 검색합니다. 출력은 PEM 인코딩에 있습니다.

```
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 [CodeCommit에 SSH 키 및 SSH 사용](#) 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetSshPublicKey 섹션을 참조하세요.

`get-user-policy`

다음 코드 예시는 `get-user-policy`의 사용 방법을 보여 줍니다.

AWS CLI

IAM 사용자에 대한 정책 세부 정보 나열

다음 `get-user-policy` 명령은 이름이 `Bob`인 IAM 사용자에게 연결된 지정된 정책의 세부 정보를 나열합니다.

```
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 [IAM의 정책 및 권한](https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies.html)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetUserPolicy를 참조하세요.

`get-user`

다음 코드 예시는 `get-user`의 사용 방법을 보여 줍니다.

AWS CLI

IAM 사용자 정보 가져오기

다음 `get-user` 명령은 이름이 `Paulo`인 IAM 사용자에 대한 정보를 가져옵니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 GetUser를 참조하세요.

list-access-keys

다음 코드 예시는 list-access-keys의 사용 방법을 보여 줍니다.

AWS CLI

IAM 사용자의 액세스 키 ID 나열

다음 list-access-keys 명령은 이름이 Bob인 IAM 사용자의 액세스 키 ID를 나열합니다.

```
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 비밀 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 사용자의 액세스 키 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListAccessKeys를 참조하세요.

`list-account-aliases`

다음 코드 예시는 `list-account-aliases`의 사용 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 별칭을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListAccountAliases를 참조하세요.

`list-attached-group-policies`

다음 코드 예시는 `list-attached-group-policies`의 사용 방법을 보여 줍니다.

AWS CLI

지정된 그룹에 연결된 모든 관리형 정책 나열

이 예제는 AWS 계정의 이름이 `Admins`인 IAM 그룹에 연결된 관리형 정책의 이름과 ARN을 반환합니다.

```
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 IAM의 정책 및 권한을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListAttachedGroupPolicies를 참조하세요.

`list-attached-role-policies`

다음 코드 예시는 `list-attached-role-policies`의 사용 방법을 보여 줍니다.

AWS CLI

지정된 IAM 역할에 연결된 모든 관리형 정책 나열

이 명령은 AWS 계정의 이름이 `SecurityAuditRole`인 IAM 역할에 연결된 관리형 정책의 이름과 ARN을 반환합니다.

```
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 IAM의 정책 및 권한을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListAttachedRolePolicies를 참조하세요.

`list-attached-user-policies`

다음 코드 예시는 `list-attached-user-policies`의 사용 방법을 보여 줍니다.

AWS CLI

지정된 사용자에게 연결된 모든 관리형 정책 나열

이 명령은 AWS 계정의 이름이 Bob인 IAM 사용자에 대한 관리형 정책의 이름과 ARN을 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 IAM의 정책 및 권한을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListAttachedUserPolicies를 참조하세요.

list-entities-for-policy

다음 코드 예시는 list-entities-for-policy의 사용 방법을 보여 줍니다.

AWS CLI

지정된 관리형 정책이 연결된 모든 사용자, 그룹 및 역할 나열

이 예제는 arn:aws:iam::123456789012:policy/TestPolicy 정책이 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 IAM의 정책 및 권한을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListEntitiesForPolicy를 참조하세요.

list-group-policies

다음 코드 예시는 list-group-policies의 사용 방법을 보여 줍니다.

AWS CLI

지정된 그룹에 연결된 모든 인라인 정책 나열

다음 list-group-policies 명령은 현재 계정에서 Admins라는 IAM 그룹에 연결된 인라인 정책의 이름을 나열합니다.

```
aws iam list-group-policies \
    --group-name Admins
```

출력:

```
{
    "PolicyNames": [
        "AdminRoot",
        "ExamplePolicy"
    ]
}
```

자세한 내용은 AWS IAM 사용 설명서의 IAM 정책 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListGroupPolicies를 참조하세요.

list-groups-for-user

다음 코드 예시는 list-groups-for-user의 사용 방법을 보여 줍니다.

AWS CLI

IAM 사용자가 속한 그룹 나열

다음 list-groups-for-user 명령은 Bob이라는 IAM 사용자가 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListGroupsForUser를 참조하세요.

list-groups

다음 코드 예시는 `list-groups`의 사용 방법을 보여 줍니다.

AWS CLI

현재 계정의 IAM 그룹 나열

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.

- API 세부 정보는 AWS CLI 명령 참조의 ListGroups를 참조하세요.

list-instance-profile-tags

다음 코드 예시는 `list-instance-profile-tags`의 사용 방법을 보여 줍니다.

AWS CLI

인스턴스 프로파일에 연결된 태그 나열

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.

- API 세부 정보는 AWS CLI 명령 참조의 ListInstanceProfileTags 섹션을 참조하세요.

list-instance-profiles-for-role

다음 코드 예시는 `list-instance-profiles-for-role`의 사용 방법을 보여 줍니다.

AWS CLI

IAM 역할에 대한 인스턴스 프로파일 나열

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.

- API 세부 정보는 AWS CLI 명령 참조의 ListInstanceProfilesForRole을 참조하세요.

list-instance-profiles

다음 코드 예시는 `list-instance-profiles`의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListInstanceProfiles를 참조하세요.

list-mfa-device-tags

다음 코드 예시는 `list-mfa-device-tags`의 사용 방법을 보여 줍니다.

AWS CLI

MFA 디바이스에 연결된 태그를 나열하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 ListMfaDeviceTags 섹션을 참조하세요.

list-mfa-devices

다음 코드 예시는 `list-mfa-devices`의 사용 방법을 보여 줍니다.

AWS CLI

지정된 사용자의 모든 MFA 디바이스 나열

이 예제는 IAM 사용자 Bob에게 할당된 MFA 디바이스에 대한 세부 정보를 반환합니다.

```
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 AWS에서 멀티 팩터 인증(MFA) 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListMfaDevices를 참조하세요.

list-open-id-connect-provider-tags

다음 코드 예시는 `list-open-id-connect-provider-tags`의 사용 방법을 보여 줍니다.

AWS CLI

OpenID Connect(OIDC) 호환 자격 증명 공급자에 연결된 태그를 나열하는 방법

다음 `list-open-id-connect-provider-tags` 명령은 지정된 OIDC ID 제공업체와 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListOpenIdConnectProviderTags 섹션을 참조하세요.

list-open-id-connect-providers

다음 코드 예시는 `list-open-id-connect-providers`의 사용 방법을 보여 줍니다.

AWS CLI

AWS 계정의 OpenID Connect 제공업체에 대한 정보 나열

이 예제는 현재 AWS 계정에 정의된 모든 OpenID Connect 제공업체의 ARNS 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 IAM에서 OIDC(OpenID Connect) ID 제공업체 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListOpenIdConnectProviders를 참조하세요.

list-organizations-features

다음 코드 예시는 list-organizations-features의 사용 방법을 보여 줍니다.

AWS CLI

조직에 대해 활성화된 중앙 집중식 루트 액세스 기능을 나열하려면

다음 list-organizations-features 명령은 조직에 대해 활성화된 중앙 집중식 루트 액세스 기능을 나열합니다.

```
aws iam list-organizations-features
```

출력:

```
{
    "EnabledFeatures": [
        "RootCredentialsManagement",
        "RootSessions"
    ]
    "OrganizationId": "o-aa111bb222"
}
```

자세한 내용은 AWS IAM 사용 설명서의 멤버 계정에 대한 루트 액세스 중앙 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListOrganizationsFeatures를 참조하세요.

list-policies-granting-service-access

다음 코드 예시는 list-policies-granting-service-access의 사용 방법을 보여 줍니다.

AWS CLI

지정된 서비스에 대한 보안 주체 액세스 권한을 부여하는 정책을 나열하는 방법

다음 list-policies-granting-service-access 예시에서는 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 CodeCommit용 IAM 자격 증명: Git 자격 증명, SSH 키 및 AWS 액세스 키 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListPoliciesGrantingServiceAccess 섹션을 참조하세요.

list-policies

다음 코드 예시는 list-policies의 사용 방법을 보여 줍니다.

AWS CLI

AWS 계정에서 사용할 수 있는 관리형 정책 나열

이 예제는 현재 AWS 계정에서 사용할 수 있는 처음 2개의 관리형 정책 컬렉션을 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 IAM의 정책 및 권한을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListPolicies를 참조하세요.

list-policy-tags

다음 코드 예시는 `list-policy-tags`의 사용 방법을 보여 줍니다.

AWS CLI

관리형 정책에 연결된 태그를 나열하는 방법

다음 `list-policy-tags` 명령은 지정된 관리 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListPolicyTags 섹션을 참조하세요.

list-policy-versions

다음 코드 예시는 list-policy-versions의 사용 방법을 보여 줍니다.

AWS CLI

지정된 관리형 정책의 버전에 대한 정보 나열

이 예제는 ARN이 arn:aws:iam::123456789012:policy/MySamplePolicy인 정책의 사용 가능한 버전 목록을 반환합니다.

```
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 IAM의 정책 및 권한을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListPolicyVersions를 참조하세요.

list-role-policies

다음 코드 예시는 list-role-policies의 사용 방법을 보여 줍니다.

AWS CLI

IAM 역할에 연결된 정책 나열

다음 `list-role-policies` 명령은 지정된 IAM 역할의 권한 정책 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListRolePolicies를 참조하세요.

list-role-tags

다음 코드 예시는 `list-role-tags`의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListRoleTags를 참조하세요.

list-roles

다음 코드 예시는 `list-roles`의 사용 방법을 보여 줍니다.

AWS CLI

현재 계정의 IAM 역할 나열

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.

- API 세부 정보는 AWS CLI 명령 참조의 ListRoles를 참조하세요.

list-saml-provider-tags

다음 코드 예시는 `list-saml-provider-tags`의 사용 방법을 보여 줍니다.

AWS CLI

SAML 공급자에 연결된 태그를 나열하는 방법

다음 `list-saml-provider-tags` 명령은 지정된 SAML 제공업체와 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListSamlProviderTags 섹션을 참조하세요.

`list-saml-providers`

다음 코드 예시는 `list-saml-providers`의 사용 방법을 보여 줍니다.

AWS CLI

AWS 계정의 SAML 공급자 나열

이 예제는 현재 AWS 계정에서 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListSAMLProviders를 참조하세요.

list-server-certificate-tags

다음 코드 예시는 `list-server-certificate-tags`의 사용 방법을 보여 줍니다.

AWS CLI

서버 인증서에 연결된 태그 나열

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.

- API 세부 정보는 AWS CLI 명령 참조의 ListServerCertificateTags 섹션을 참조하세요.

list-server-certificates

다음 코드 예시는 `list-server-certificates`의 사용 방법을 보여 줍니다.

AWS CLI

AWS 계정의 서버 인증서 나열

다음 `list-server-certificates` 명령은 AWS 계정에 저장되어 사용 가능한 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 IAM에서 서버 인증서 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListServerCertificates를 참조하세요.

list-service-specific-credential

다음 코드 예시는 list-service-specific-credential의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 사용자의 서비스별 자격 증명 나열

다음 list-service-specific-credentials 예시에서는 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"
    }
}
```

예시 2: 지정된 서비스로 필터링된 사용자의 서비스별 자격 증명 나열

다음 list-service-specific-credentials 예시에서는 요청을 하는 사용자에게 할당된 서비스별 자격 증명을 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 Create Git credentials for HTTPS connections to CodeCommit 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListServiceSpecificCredential 섹션을 참조하세요.

`list-service-specific-credentials`

다음 코드 예시는 `list-service-specific-credentials`의 사용 방법을 보여 줍니다.

AWS CLI

자격 증명 목록을 검색하는 방법

다음 `list-service-specific-credentials` 예시에서는 이름이 `developer`인 사용자의 AWS CodeCommit 리포지토리에 대한 HTTPS 액세스를 위해 생성된 보안 인증 정보를 나열합니다.

```
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 [Create Git credentials for HTTPS connections to CodeCommit](#) 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ListServiceSpecificCredentials](#) 섹션을 참조하세요.

list-signing-certificates

다음 코드 예시는 `list-signing-certificates`의 사용 방법을 보여 줍니다.

AWS CLI

IAM 사용자의 서명 인증서 나열

다음 `list-signing-certificates` 명령은 Bob이라는 IAM 사용자의 서명 인증서를 나열합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 ListSigningCertificates를 참조하세요.

list-ssh-public-keys

다음 코드 예시는 list-ssh-public-keys의 사용 방법을 보여 줍니다.

AWS CLI

IAM 사용자에게 연결된 SSH 퍼블릭 키를 나열하는 방법

다음 list-ssh-public-keys 예시에서는 IAM 사용자 sofia에 연결된 SSH 퍼블릭 키를 나열합니다.

```
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 CodeCommit에 SSH 키 및 SSH 사용 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListSshPublicKeys 섹션을 참조하세요.

list-user-policies

다음 코드 예시는 list-user-policies의 사용 방법을 보여 줍니다.

AWS CLI

IAM 사용자에 대한 정책 나열

다음 list-user-policies 명령은 이름이 Bob인 IAM 사용자에게 연결된 정책을 나열합니다.

```
aws iam list-user-policies \
    --user-name Bob
```

출력:

```
{
    "PolicyNames": [
        "ExamplePolicy",
        "TestPolicy"
    ]
}
```

자세한 내용은 AWS IAM 사용 설명서의 AWS 계정에서 IAM 사용자 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListUserPolicies를 참조하세요.

list-user-tags

다음 코드 예시는 list-user-tags의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListUserTags를 참조하세요.

list-users

다음 코드 예시는 `list-users`의 사용 방법을 보여 줍니다.

AWS CLI

IAM 사용자 나열

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.

- API 세부 정보는 AWS CLI 명령 참조의 ListUsers를 참조하세요.

list-virtual-mfa-devices

다음 코드 예시는 list-virtual-mfa-devices의 사용 방법을 보여 줍니다.

AWS CLI

가상 MFA 디바이스 나열

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 멀티 팩터 인증(MFA) 디바이스 활성화를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListVirtualMfaDevices를 참조하세요.

put-group-policy

다음 코드 예시는 put-group-policy의 사용 방법을 보여 줍니다.

AWS CLI

그룹에 정책 추가

다음 put-group-policy 명령은 이름이 Admins인 IAM 그룹에 정책을 추가합니다.

```
aws iam put-group-policy \
    --group-name Admins \
    --policy-document file://AdminPolicy.json \
    --policy-name AdminRoot
```

이 명령은 출력을 생성하지 않습니다.

정책은 AdminPolicy.json 파일에서 JSON 문서로 정의됩니다. (파일 이름과 확장자는 중요하지 않습니다.)

자세한 내용은 AWS IAM 사용 설명서의 IAM 정책 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutGroupPolicy를 참조하세요.

put-role-permissions-boundary

다음 코드 예시는 put-role-permissions-boundary의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: IAM 역할에 사용자 지정 정책을 기반으로 권한 경계 적용

다음 put-role-permissions-boundary 예제는 intern-boundary라는 사용자 지정 정책을 지정된 IAM 역할에 대한 권한 경계로 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: IAM 역할에 AWS 관리형 정책을 기반으로 권한 경계를 적용하는 방법

다음 put-role-permissions-boundary 예제는 AWS 관리형PowerUserAccess 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 변경을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutRolePermissionsBoundary를 참조하세요.

put-role-policy

다음 코드 예시는 put-role-policy의 사용 방법을 보여 줍니다.

AWS CLI

IAM 역할에 권한 정책 연결

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 파일에서 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 변경을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutRolePolicy를 참조하세요.

put-user-permissions-boundary

다음 코드 예시는 put-user-permissions-boundary의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: IAM 사용자에게 사용자 지정 정책을 기반으로 권한 경계 적용

다음 put-user-permissions-boundary 예제는 intern-boundary라는 사용자 지정 정책을 지정된 IAM 사용자에 대한 권한 경계로 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: IAM 사용자에게 AWS 관리형 정책을 기반으로 권한 경계를 적용하는 방법

다음 `put-user-permissions-boundary` 예제는 `PowerUserAccess`라는 AWS 관리형 정책을 지정된 IAM 사용자에 대한 권한 경계로 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.

- API 세부 정보는 AWS CLI 명령 참조의 PutUserPermissionsBoundary를 참조하세요.

put-user-policy

다음 코드 예시는 `put-user-policy`의 사용 방법을 보여 줍니다.

AWS CLI

IAM 사용자에게 정책 연결

다음 `put-user-policy` 명령은 정책을 이름이 `Bob`인 IAM 사용자에게 연결합니다.

```
aws iam put-user-policy \
    --user-name Bob \
    --policy-name ExamplePolicy \
    --policy-document file://AdminPolicy.json
```

이 명령은 출력을 생성하지 않습니다.

정책은 AdminPolicy.json 파일에서 JSON 문서로 정의됩니다. (파일 이름과 확장자는 중요하지 않습니다.)

자세한 내용은 AWS IAM 사용 설명서의 IAM 자격 증명 권한 추가 및 제거를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutUserPolicy를 참조하세요.

remove-client-id-from-open-id-connect-provider

다음 코드 예시는 remove-client-id-from-open-id-connect-provider의 사용 방법을 보여 줍니다.

AWS CLI

지정된 IAM OpenID Connect 제공업체에 등록된 클라이언트 ID 목록에서 지정된 클라이언트 ID 제거

이 예제는 ARN이 arn:aws:iam::123456789012:oidc-provider/example.oidcprovider.com인 IAM OIDC 제공업체와 연결된 클라이언트 ID 목록에서 클라이언트 ID My-TestApp-3을 제거합니다.

```
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 IAM에서 OIDC(OpenID Connect) ID 제공업체 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RemoveClientIdFromOpenIdConnectProvider를 참조하세요.

remove-role-from-instance-profile

다음 코드 예시는 remove-role-from-instance-profile의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 RemoveRoleFromInstanceProfile을 참조하세요.

remove-user-from-group

다음 코드 예시는 `remove-user-from-group`의 사용 방법을 보여 줍니다.

AWS CLI

IAM 그룹에서 사용자 제거

다음 `remove-user-from-group` 명령은 이름이 `Admins`인 IAM 그룹에서 이름이 `Bob`인 사용자를 제거합니다.

```
aws iam remove-user-from-group \
    --user-name Bob \
    --group-name Admins
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 IAM 사용자 그룹에서 사용자 추가 및 제거를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RemoveUserFromGroup을 참조하세요.

reset-service-specific-credential

다음 코드 예시는 `reset-service-specific-credential`의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 요청을 하는 사용자에게 연결된 서비스별 보안 인증의 암호 재설정

다음 `reset-service-specific-credential` 예시에서는 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"
    }
}
```

예시 2: 지정된 사용자에게 연결된 서비스별 자격 증명의 암호 재설정

다음 `reset-service-specific-credential` 예시에서는 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 Create Git credentials for HTTPS connections to CodeCommit 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ResetServiceSpecificCredential 섹션을 참조하세요.

resync-mfa-device

다음 코드 예시는 resync-mfa-device의 사용 방법을 보여 줍니다.

AWS CLI

MFA 디바이스 동기화

다음 resync-mfa-device 예제는 IAM 사용자 Bob과 연결되어 있고 ARN이 arn:aws:iam::123456789012:mfa/BobsMFADevice인 MFA 디바이스를 두 개의 인증 코드를 제공한 인증 프로그램과 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 AWS에서 멀티 팩터 인증(MFA) 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ResyncMfaDevice를 참조하세요.

set-default-policy-version

다음 코드 예시는 set-default-policy-version의 사용 방법을 보여 줍니다.

AWS CLI

지정된 정책의 지정된 버전을 정책의 기본 버전으로 설정

이 예제는 ARN이 arn:aws:iam::123456789012:policy/MyPolicy인 정책의 v2 버전을 기본 활성 버전으로 설정합니다.

```
aws iam set-default-policy-version \
    --policy-arn arn:aws:iam::123456789012:policy/MyPolicy \
    --version-id v2
```

자세한 내용은 AWS IAM 사용 설명서의 IAM의 정책 및 권한을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 SetDefaultPolicyVersion을 참조하세요.

set-security-token-service-preferences

다음 코드 예시는 set-security-token-service-preferences의 사용 방법을 보여 줍니다.

AWS CLI

글로벌 엔드포인트 토큰 버전을 설정하는 방법

다음 set-security-token-service-preferences 예시에서는 글로벌 엔드포인트에 대해 인증할 때 버전 2 토큰을 사용하도록 Amazon STS를 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 AWS STS 관리 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 SetSecurityTokenServicePreferences 섹션을 참조하세요.

simulate-custom-policy

다음 코드 예시는 simulate-custom-policy의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: IAM 사용자 또는 역할과 연결된 모든 IAM 정책의 효과 시뮬레이션

다음 simulate-custom-policy는 정책을 제공하고 변수 값을 정의하고 API 호출을 시뮬레이션하여 허용 또는 거부 여부를 확인하는 방법을 보여줍니다. 다음 예시는 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": []
        }
    ]
}
```

예시 2: 정책에서 금지하는 명령 시뮬레이션

다음 simulate-custom-policy 예시에서는 정책에서 금지하는 명령을 시뮬레이션한 결과를 보여줍니다. 이 예시에서는 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 시뮬레이터로 IAM 정책 테스트 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 SimulateCustomPolicy 섹션을 참조하세요.

simulate-principal-policy

다음 코드 예시는 simulate-principal-policy의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 임의 IAM 정책의 효과 시뮬레이션

다음 simulate-principal-policy는 API 작업을 호출하고 해당 사용자와 연결된 정책이 작업을 허용 또는 거부할지 여부를 결정하는 사용자를 시뮬레이션하는 방법을 보여줍니다. 다음 예시에서는 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": []
            }
        ]
}
```

예시 2: 금지된 명령의 효과를 시뮬레이션

다음 simulate-custom-policy 예시는 사용자의 정책 중 하나에 의해 금지된 명령을 시뮬레이션한 결과를 보여줍니다. 다음 예시에서는 사용자에게 특정 날짜 및 시간 이후에만 DynamoDB 데이터베이스에 대한 액세스를 허용하는 정책이 있습니다. 시뮬레이션에는 사용자가 정책 조건에서 허용하는 값보다 빠른 aws:CurrentTime 값으로 데이터베이스에 액세스하려고 시도하는 것이 있습니다.

```
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 시뮬레이터로 IAM 정책 테스트 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 SimulatePrincipalPolicy 섹션을 참조하세요.

tag-instance-profile

다음 코드 예시는 `tag-instance-profile`의 사용 방법을 보여 줍니다.

AWS CLI

인스턴스 프로파일에 태그 추가

다음 `tag-instance-profile` 명령은 지정된 인스턴스 프로파일에 Department 이름이 포함된 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.

- API 세부 정보는 AWS CLI 명령 참조의 TagInstanceProfile 섹션을 참조하세요.

tag-mfa-device

다음 코드 예시는 `tag-mfa-device`의 사용 방법을 보여 줍니다.

AWS CLI

MFA 디바이스에 태그를 추가하는 방법

다음 `tag-mfa-device` 명령은 지정된 MFA 디바이스에 Department 이름이 포함된 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.

- API 세부 정보는 AWS CLI 명령 참조의 TagMfaDevice 섹션을 참조하세요.

`tag-open-id-connect-provider`

다음 코드 예시는 `tag-open-id-connect-provider`의 사용 방법을 보여 줍니다.

AWS CLI

OpenID Connect(OIDC) 호환 자격 증명 공급자에 태그를 추가하는 방법

다음 `tag-open-id-connect-provider` 명령은 지정된 OIDC ID 제공업체에 Department 이름이 포함된 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.

- API 세부 정보는 AWS CLI 명령 참조의 TagOpenIdConnectProvider 섹션을 참조하세요.

`tag-policy`

다음 코드 예시는 `tag-policy`의 사용 방법을 보여 줍니다.

AWS CLI

고객 관리형 정책에 태그를 추가하는 방법

다음 `tag-policy` 명령은 지정된 고객 관리 정책에 부서 이름이 포함된 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.

- API 세부 정보는 AWS CLI 명령 참조의 TagPolicy 섹션을 참조하세요.

tag-role

다음 코드 예시는 `tag-role`의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 TagRole을 참조하세요.

tag-saml-provider

다음 코드 예시는 `tag-saml-provider`의 사용 방법을 보여 줍니다.

AWS CLI

SAML 공급자에 태그를 추가하는 방법

다음 `tag-saml-provider` 명령은 지정된 SAML 제공업체에 Department 이름이 포함된 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.

- API 세부 정보는 AWS CLI 명령 참조의 TagSamlProvider 섹션을 참조하세요.

tag-server-certificate

다음 코드 예시는 `tag-server-certificate`의 사용 방법을 보여 줍니다.

AWS CLI

서버 인증서에 태그 추가

다음 `tag-saml-provider` 명령은 지정된 서버 증명서에 Department 이름이 포함된 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.

- API 세부 정보는 AWS CLI 명령 참조의 TagServerCertificate 섹션을 참조하세요.

tag-user

다음 코드 예시는 `tag-user`의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 TagUser를 참조하세요.

untag-instance-profile

다음 코드 예시는 untag-instance-profile의 사용 방법을 보여 줍니다.

AWS CLI

인스턴스 프로파일에서 태그 제거

다음 untag-instance-profile 명령은 지정된 인스턴스 프로파일에서 키 이름이 'Department'인 태그를 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.

- API 세부 정보는 AWS CLI 명령 참조의 UntagInstanceProfile 섹션을 참조하세요.

untag-mfa-device

다음 코드 예시는 untag-mfa-device의 사용 방법을 보여 줍니다.

AWS CLI

MFA 디바이스에서 태그를 제거하는 방법

다음 untag-mfa-device 명령은 지정된 MFA 디바이스에서 키 이름이 'Department'인 모든 태그를 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.

- API 세부 정보는 AWS CLI 명령 참조의 UntagMfaDevice 섹션을 참조하세요.

untag-open-id-connect-provider

다음 코드 예시는 untag-open-id-connect-provider의 사용 방법을 보여 줍니다.

AWS CLI

OIDC ID 제공업체에서 태그 제거

다음 untag-open-id-connect-provider 명령은 지정된 OIDC ID 제공업체에서 키 이름이 'Department'인 모든 태그를 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.

- API 세부 정보는 AWS CLI 명령 참조의 UntagOpenIdConnectProvider 섹션을 참조하세요.

untag-policy

다음 코드 예시는 untag-policy의 사용 방법을 보여 줍니다.

AWS CLI

고객 관리형 정책에서 태그를 제거하는 방법

다음 untag-policy 명령은 지정된 고객 관리형 정책에서 키 이름이 'Department'인 모든 태그를 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 지정](#)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [UntagPolicy](#) 섹션을 참조하세요.

`untag-role`

다음 코드 예시는 `untag-role`의 사용 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 지정](#)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [UntagRole](#)을 참조하세요.

`untag-saml-provider`

다음 코드 예시는 `untag-saml-provider`의 사용 방법을 보여 줍니다.

AWS CLI

SAML 공급자에서 태그를 제거하는 방법

다음 `untag-saml-provider` 명령은 지정된 인스턴스 프로파일에서 키 이름이 'Department'인 태그를 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.

- API 세부 정보는 AWS CLI 명령 참조의 UntagSamlProvider 섹션을 참조하세요.

untag-server-certificate

다음 코드 예시는 untag-server-certificate의 사용 방법을 보여 줍니다.

AWS CLI

서버 인증서에서 태그 제거

다음 untag-server-certificate 명령은 지정된 서버 인증서에서 키 이름이 'Department'인 모든 태그를 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.

- API 세부 정보는 AWS CLI 명령 참조의 UntagServerCertificate 섹션을 참조하세요.

untag-user

다음 코드 예시는 untag-user의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 UntagUser를 참조하세요.

update-access-key

다음 코드 예시는 update-access-key의 사용 방법을 보여 줍니다.

AWS CLI

IAM 사용자의 액세스 키를 활성화 또는 비활성화

다음 update-access-key 명령은 이름이 Bob인 IAM 사용자의 지정된 액세스 키(액세스 키 ID 및 비밀 액세스 키)를 비활성화합니다.

```
aws iam update-access-key \
    --access-key-id AKIAIOSFODNN7EXAMPLE \
    --status Inactive \
    --user-name Bob
```

이 명령은 출력을 생성하지 않습니다.

키를 비활성화하면 프로그래밍 방식으로 AWS에 액세스하는 데 키를 사용할 수 없습니다. 하지만 키는 계속 사용할 수 있고 다시 활성화할 수 있습니다.

자세한 내용은 AWS IAM 사용 설명서의 IAM 사용자의 액세스 키 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateAccessKey를 참조하세요.

update-account-password-policy

다음 코드 예시는 update-account-password-policy의 사용 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 정책 변경은 해당 계정의 IAM 사용자에 대해 새로 생성되는 모든 암호에 영향을 줍니다. 암호 정책 변경은 기존 암호에 영향을 주지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 IAM 사용자의 계정 암호 정책 설정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateAccountPasswordPolicy를 참조하세요.

update-assume-role-policy

다음 코드 예시는 update-assume-role-policy의 사용 방법을 보여 줍니다.

AWS CLI

IAM 역할의 신뢰 정책 업데이트

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 파일에 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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateAssumeRolePolicy를 참조하세요.

update-group

다음 코드 예시는 update-group의 사용 방법을 보여 줍니다.

AWS CLI

IAM 그룹 이름 바꾸기

다음 `update-group` 명령은 IAM 그룹의 이름을 `Test`에서 `Test-1`로 변경합니다.

```
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 변경을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateGroup을 참조하세요.

`update-login-profile`

다음 코드 예시는 `update-login-profile`의 사용 방법을 보여 줍니다.

AWS CLI

IAM 사용자의 암호 업데이트

다음 `update-login-profile` 명령은 `Bob`이라는 IAM 사용자에 대한 새 암호를 생성합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateLoginProfile을 참조하세요.

update-open-id-connect-provider-thumbprint

다음 코드 예시는 update-open-id-connect-provider-thumbprint의 사용 방법을 보여 줍니다.

AWS CLI

기존 서버 인증서 지문 목록을 새 목록으로 바꾸기

이 예제는 ARN이 arn:aws:iam::123456789012:oidc-provider/example.oidcprovider.com인 OIDC 제공업체에 대한 인증서 지문 목록을 업데이트하여 새 지문을 사용합니다.

```
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 [IAM에서 OIDC(OpenID Connect) ID 제공업체 생성](#)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [UpdateOpenIdConnectProviderThumbprint](#)를 참조하세요.

update-role-description

다음 코드 예시는 update-role-description의 사용 방법을 보여 줍니다.

AWS CLI

IAM 역할의 설명 변경

다음 update-role 명령은 IAM 역할에 대한 설명을 production-role에서 Main production role로 변경합니다.

```
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 변경을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateRoleDescription을 참조하세요.

update-role

다음 코드 예시는 update-role의 사용 방법을 보여 줍니다.

AWS CLI

IAM 역할의 설명 또는 세션 기간 변경

다음 update-role 명령은 IAM 역할 production-role의 설명을 Main production role로 변경하고 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 변경을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateRole을 참조하세요.

update-saml-provider

다음 코드 예시는 update-saml-provider의 사용 방법을 보여 줍니다.

AWS CLI

기존 SAML 제공업체에 대한 메타데이터 문서 업데이트

이 예제는 ARN이 arn:aws:iam::123456789012:saml-provider/SAMLADFS인 IAM의 SAML 제공업체를 SAMLMetaData.xml 파일의 새 SAML 메타데이터 문서로 업데이트합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateSamlProvider를 참조하세요.

update-server-certificate

다음 코드 예시는 update-server-certificate의 사용 방법을 보여 줍니다.

AWS CLI

AWS 계정에 있는 서버 인증서의 경로 또는 이름 변경

다음 update-server-certificate 명령은 인증서의 이름을 myServerCertificate에서 myUpdatedServerCertificate로 변경합니다. 또한 Amazon CloudFront 서비스에서 액세스

할 수 있도록 경로를 /cloudfront/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 IAM에서 서버 인증서 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateServerCertificate를 참조하세요.

update-service-specific-credential

다음 코드 예시는 update-service-specific-credential의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 요청 사용자의 서비스별 자격 증명 상태 업데이트

다음 update-service-specific-credential 예시에서는 Inactive에 요청하는 사용자의 지정된 자격 증명 상태를 변경합니다.

```
aws iam update-service-specific-credential \
    --service-specific-credential-id ACCAEXAMPLE123EXAMPLE \
    --status Inactive
```

이 명령은 출력을 생성하지 않습니다.

예시 2: 지정된 사용자의 서비스별 자격 증명의 상태 업데이트

다음 update-service-specific-credential 예시에서는 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 Create Git credentials for HTTPS connections to CodeCommit 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateServiceSpecificCredential 섹션을 참조하세요.

update-signing-certificate

다음 코드 예시는 `update-signing-certificate`의 사용 방법을 보여 줍니다.

AWS CLI

IAM 사용자의 서명 인증서 활성화 또는 비활성

다음 `update-signing-certificate` 명령은 Bob이라는 IAM 사용자에 대해 지정된 서명 인증서를 비활성화합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateSigningCertificate를 참조하세요.

update-ssh-public-key

다음 코드 예시는 `update-ssh-public-key`의 사용 방법을 보여 줍니다.

AWS CLI

SSH 퍼블릭 키의 상태를 변경하는 방법

다음 `update-ssh-public-key` 명령은 지정된 퍼블릭 키의 상태를 `Inactive`로 변경합니다.

```
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 CodeCommit에 SSH 키 및 SSH 사용 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateSshPublicKey 섹션을 참조하세요.

update-user

다음 코드 예시는 update-user의 사용 방법을 보여 줍니다.

AWS CLI

IAM 사용자의 이름 변경

다음 update-user 명령은 IAM 사용자의 이름을 Bob에서 Robert로 변경합니다.

```
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 변경을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateUser를 참조하세요.

upload-server-certificate

다음 코드 예시는 upload-server-certificate의 사용 방법을 보여 줍니다.

AWS CLI

AWS 계정에 서버 인증서 업로드

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 myServerCertificate 이름 아래에서 사용할 수 있습니다. 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 IAM 사용 설명서의 서버 인증서 생성, 업로드 및 삭제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UploadServerCertificate를 참조하세요.

upload-signing-certificate

다음 코드 예시는 upload-signing-certificate의 사용 방법을 보여 줍니다.

AWS CLI

IAM 사용자의 서명 인증서 업로드

다음 upload-signing-certificate 명령은 Bob이라는 IAM 사용자의 서명 인증서를 업로드합니다.

```
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 PEM 형식의 certificate.pem이라는 파일에 있습니다.

자세한 내용은 IAM 사용 설명서의 사용자 서명 인증서 생성 및 업로드를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UploadSigningCertificate를 참조하세요.

`upload-ssh-public-key`

다음 코드 예시는 `upload-ssh-public-key`의 사용 방법을 보여 줍니다.

AWS CLI

SSH 퍼블릭 키를 업로드하고 사용자와 연결하는 방법

다음 `upload-ssh-public-key` 명령은 `sshkey.pub` 파일에 있는 퍼블릭 키를 업로드하여 사용자 `sofia`에 연결합니다.

```
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 CodeCommit의 퍼블릭 키와 프라이빗 키 설정 또는 SSH 및 Windows: Git 및 CodeCommit에 대한 퍼블릭 키와 프라이빗 키 설정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UploadSshPublicKey 섹션을 참조하세요.

AWS CLI를 사용하는 IAM Access Analyzer 예제

다음 코드 예제에서는 IAM Access Analyzer와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`apply-archive-rule`

다음 코드 예시는 `apply-archive-rule`의 사용 방법을 보여 줍니다.

AWS CLI

아카이브 규칙 기준을 충족하는 기존 결과에 아카이브 규칙을 적용하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 [ApplyArchiveRule]을 참조하세요.

cancel-policy-generation

다음 코드 예시는 cancel-policy-generation의 사용 방법을 보여 줍니다.

AWS CLI

요청된 정책 생성을 취소하는 방법

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 생성]을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [CancelPolicyGeneration]을 참조하세요.

check-access-not-granted

다음 코드 예시는 check-access-not-granted의 사용 방법을 보여 줍니다.

AWS CLI

지정된 액세스가 정책에서 허용되지 않는지 확인하는 방법

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:::amzn-s3-demo-bucket",
                "arn:aws:s3:::amzn-s3-demo-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 Analyzer API를 사용하여 액세스 미리 보기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CheckAccessNotGranted를 참조하세요.

check-no-new-access

다음 코드 예시는 check-no-new-access의 사용 방법을 보여 줍니다.

AWS CLI

기존 정책과 비교하여 업데이트된 정책에 새 액세스가 허용되는지 확인하는 방법

다음 check-no-new-access 예제에서는 기존 정책과 비교하여 업데이트된 정책에 새 액세스가 허용되는지 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.json`의 콘텐츠:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket",
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
            ]
        }
    ]
}
```

`new-policy.json`의 콘텐츠:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:GetObjectAcl",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket",
                "arn:aws:s3:::amzn-s3-demo-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 Analyzer API를 사용하여 액세스 미리 보기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CheckNoNewAccess를 참조하세요.

check-no-public-access

다음 코드 예시는 check-no-public-access의 사용 방법을 보여 줍니다.

AWS CLI

지정된 리소스 유형에 대해 리소스 정책이 퍼블릭 액세스 권한을 부여할 수 있는지 확인하는 방법

다음 check-no-public-access 예제에서는 지정된 리소스 유형에 대해 리소스 정책이 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 Analyzer API를 사용하여 액세스 미리 보기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CheckNoPublicAccess를 참조하세요.

`create-access-preview`

다음 코드 예시는 `create-access-preview`의 사용 방법을 보여 줍니다.

AWS CLI

리소스 권한을 배포하기 전에 리소스에 대한 IAM Access Analyzer 결과를 미리 볼 수 있는 액세스 미리 보기를 생성하는 방법

다음 `create-access-preview` 예제에서는 AWS 계정에서 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:::amzn-s3-demo-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:::amzn-s3-demo-
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 Analyzer API를 사용하여 액세스 미리 보기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateAccessPreview를 참조하세요.

create-analyzer

다음 코드 예시는 `create-analyzer`의 사용 방법을 보여 줍니다.

AWS CLI

분석기를 생성하는 방법

다음 `create-analyzer` 예제에서는 AWS 계정에서 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 시작을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateAnalyzer를 참조하세요.

create-archive-rule

다음 코드 예시는 create-archive-rule의 사용 방법을 보여 줍니다.

AWS CLI

지정된 분석기에 대한 아카이브 규칙을 생성하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateArchiveRule을 참조하세요.

delete-analyzer

다음 코드 예시는 delete-analyzer의 사용 방법을 보여 줍니다.

AWS CLI

지정된 분석기를 삭제하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteAnalyzer를 참조하세요.

`delete-archive-rule`

다음 코드 예시는 `delete-archive-rule`의 사용 방법을 보여 줍니다.

AWS CLI

지정된 아카이브 규칙을 삭제하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteArchiveRule을 참조하세요.

`get-access-preview`

다음 코드 예시는 `get-access-preview`의 사용 방법을 보여 줍니다.

AWS CLI

지정된 분석기의 액세스 미리 보기에 대한 정보를 검색하는 방법

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:::amzn-s3-demo-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:::amzn-
s3-demo-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 Analyzer API를 사용하여 액세스 미리 보기]를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetAccessPreview를 참조하세요.

get-analyzed-resource

다음 코드 예시는 get-analyzed-resource의 사용 방법을 보여 줍니다.

AWS CLI

분석된 리소스에 대한 정보를 검색하는 방법

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:::amzn-s3-demo-bucket
```

출력:

```
{
    "resource": {
        "analyzedAt": "2024-02-15T18:01:53.002000+00:00",
        "isPublic": false,
        "resourceArn": "arn:aws:s3:::amzn-s3-demo-bucket",
        "resourceOwnerAccount": "111122223333",
        "resourceType": "AWS::S3::Bucket"
    }
}
```

자세한 내용은 AWS IAM 사용 설명서의 AWS Identity and Access Management Access Analyzer 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetAnalyzedResource를 참조하세요.

get-analyzer

다음 코드 예시는 get-analyzer의 사용 방법을 보여 줍니다.

AWS CLI

지정된 분석기에 대한 정보를 검색하는 방법

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 사용](#)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [GetAnalyzer](#)를 참조하세요.

`get-archive-rule`

다음 코드 예시는 `get-archive-rule`의 사용 방법을 보여 줍니다.

AWS CLI

아카이브 규칙에 대한 정보를 검색하는 방법

다음 `get-archive-rule` 예제에서는 AWS 계정에서 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.

- API 세부 정보는 AWS CLI 명령 참조의 GetArchiveRule을 참조하세요.

get-finding-v2

다음 코드 예시는 `get-finding-v2`의 사용 방법을 보여 줍니다.

AWS CLI

지정된 결과에 대한 정보를 검색하는 방법

다음 `get-finding-v2` 예제에서는 AWS 계정에서 지정된 결과에 대한 정보를 검색합니다.

```
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 세부 정보는 AWS CLI 명령 참조의 GetFindingV2를 참조하세요.

get-finding

다음 코드 예시는 get-finding의 사용 방법을 보여 줍니다.

AWS CLI

지정된 결과에 대한 정보를 검색하는 방법

다음 get-finding 예제에서는 AWS 계정에서 지정된 결과에 대한 정보를 검색합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 GetFinding을 참조하세요.

get-generated-policy

다음 코드 예시는 `get-generated-policy`의 사용 방법을 보여 줍니다.

AWS CLI

`StartPolicyGeneration` API를 사용하여 생성된 정책을 검색하는 방법

다음 `get-generated-policy` 예제에서는 AWS 계정에서 StartPolicyGeneration API를 사용하여 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.

- API 세부 정보는 AWS CLI 명령 참조의 GetGeneratedPolicy를 참조하세요.

list-access-preview-findings

다음 코드 예시는 list-access-preview-findings의 사용 방법을 보여 줍니다.

AWS CLI

지정된 액세스 미리 보기를 통해 생성된 액세스 미리 보기 결과의 목록을 검색하는 방법

다음 list-access-preview-findings 예제에서는 AWS 계정에서 지정된 액세스 미리 보기를 통해 생성된 액세스 미리 보기 결과의 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:::amzn-s3-demo-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 Analyzer API를 사용하여 액세스 미리 보기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListAccessPreviewFindings를 참조하세요.

`list-access-previews`

다음 코드 예시는 `list-access-previews`의 사용 방법을 보여 줍니다.

AWS CLI

지정된 분석기에 대한 액세스 미리 보기의 목록을 검색하는 방법

다음 `list-access-previews` 예제에서는 AWS 계정에서 지정된 분석기에 대한 액세스 미리 보기의 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 Analyzer API를 사용하여 액세스 미리 보기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListAccessPreviews를 참조하세요.

`list-analyzed-resources`

다음 코드 예시는 `list-analyzed-resources`의 사용 방법을 보여 줍니다.

AWS CLI

사용 가능한 위젯을 나열하는 방법

다음 `list-analyzed-resources` 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListAnalyzedResources를 참조하세요.

list-analyzers

다음 코드 예시는 `list-analyzers`의 사용 방법을 보여 줍니다.

AWS CLI

분석기 목록을 검색하는 방법

다음 `list-analyzers` 예제에서는 AWS 계정에서 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:::amzn-s3-demo-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.

- API 세부 정보는 AWS CLI 명령 참조의 ListAnalyzers를 참조하세요.

list-archive-rules

다음 코드 예시는 `list-archive-rules`의 사용 방법을 보여 줍니다.

AWS CLI

지정된 분석기에 대해 생성된 아카이브 규칙의 목록을 검색하는 방법

다음 `list-archive-rules` 예제에서는 AWS 계정에서 지정된 분석기에 대해 생성된 아카이브 규칙의 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 사용](#)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ListArchiveRules](#)를 참조하세요.

list-findings-v2

다음 코드 예시는 `list-findings-v2`의 사용 방법을 보여 줍니다.

AWS CLI

지정된 분석기를 통해 생성된 결과의 목록을 검색하는 방법

다음 `list-findings-v2` 예제에서는 AWS 계정에서 지정된 분석기를 통해 생성된 결과의 목록을 검색합니다. 이 예제에서는 이름에 `Cognito`가 들어 있는 IAM 역할만 포함하도록 결과를 필터링합니다.

```
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 세부 정보는 AWS CLI 명령 참조의 ListFindingsV2를 참조하세요.

list-findings

다음 코드 예시는 `list-findings`의 사용 방법을 보여 줍니다.

AWS CLI

지정된 분석기를 통해 생성된 결과의 목록을 검색하는 방법

다음 `list-findings` 예제에서는 AWS 계정에서 지정된 분석기를 통해 생성된 결과의 목록을 검색합니다. 이 예제에서는 이름에 `Cognito`가 들어 있는 IAM 역할만 포함하도록 결과를 필터링합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 ListFindings를 참조하세요.

`list-policy-generations`

다음 코드 예시는 `list-policy-generations`의 사용 방법을 보여 줍니다.

AWS CLI

지난 7일 동안 요청된 모든 정책 생성을 나열하는 방법

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 Access Analyzer 정책 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListPolicyGenerations를 참조하세요.

`list-tags-for-resource`

다음 코드 예시는 `list-tags-for-resource`의 사용 방법을 보여 줍니다.

AWS CLI

지정된 리소스에 적용된 태그의 목록을 검색하는 방법

다음 `list-tags-for-resource` 예제에서는 AWS 계정에서 지정된 리소스에 적용된 태그의 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 Access Analyzer 정책 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource를 참조하세요.

`start-policy-generation`

다음 코드 예시는 `start-policy-generation`의 사용 방법을 보여 줍니다.

AWS CLI

정책 생성 요청을 시작하는 방법

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.json`의 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 Access Analyzer 정책 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartPolicyGeneration을 참조하세요.

`start-resource-scan`

다음 코드 예시는 `start-resource-scan`의 사용 방법을 보여 줍니다.

AWS CLI

지정된 리소스에 적용된 정책의 스캔을 즉시 시작하는 방법

다음 `start-resource-scan` 예제에서는 AWS 계정에서 지정된 리소스에 적용된 정책의 스캔을 즉시 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 Access Analyzer 정책 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartResourceScan을 참조하세요.

`tag-resource`

다음 코드 예시는 `tag-resource`의 사용 방법을 보여 줍니다.

AWS CLI

지정된 리소스에 태그를 추가하는 방법

다음 `tag-resource` 예제에서는 AWS 계정에서 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.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

`untag-resource`

다음 코드 예시는 `untag-resource`의 사용 방법을 보여 줍니다.

AWS CLI

지정된 리소스에서 태그를 제거하는 방법

다음 `untag-resource` 예제에서는 AWS 계정에서 지정된 리소스로부터 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.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

update-archive-rule

다음 코드 예시는 update-archive-rule의 사용 방법을 보여 줍니다.

AWS CLI

지정된 아카이브 규칙에 대한 기준 및 값을 업데이트하는 방법

다음 update-archive-rule 예제에서는 AWS 계정에서 지정된 아카이브 규칙에 대한 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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateArchiveRule을 참조하세요.

update-findings

다음 코드 예시는 update-findings의 사용 방법을 보여 줍니다.

AWS CLI

지정된 결과의 상태를 업데이트하는 방법

다음 update-findings 예제에서는 AWS 계정에서 지정된 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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateFindings를 참조하세요.

validate-policy

다음 코드 예시는 validate-policy의 사용 방법을 보여 줍니다.

AWS CLI

정책 검증을 요청하고 결과 목록을 반환하는 방법

다음 validate-policy 예제에서는 정책의 검증을 요청하고 결과 목록을 반환합니다. 이 예제의 정책은 웹 ID 페더레이션에 사용되는 Amazon Cognito 역할에 대한 역할 신뢰 정책입니다. 잘못된 수임 역할 작업인 sts:AssumeRole이 사용되므로, 신뢰 정책에서 생성된 결과는 빈 Sid 요소 값 및 불일치 정책 보안 주체와 관련됩니다. Cognito와 함께 사용해야 하는 올바른 역할 수임 작업은 sts:AssumeRoleWithWebIdentity입니다.

```
aws accessanalyzer validate-policy \
    --policy-document file://myfile.json \
    --policy-type RESOURCE_POLICY
```

myfile.json의 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 검사를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ValidatePolicy를 참조하세요.

AWS CLI를 사용한 Image Builder 예시

다음 코드 예시에서는 Image Builder에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`create-component`

다음 코드 예시에서는 `create-component` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

구성 요소 생성

다음 `create-component` 예시에서는 JSON 문서 파일을 사용하고 Amazon S3 버킷에 업로드된 YAML 형식의 구성 요소 문서를 참조하는 구성 요소를 생성합니다.

```
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 Setting Up and Managing an EC2 Image Builder Image Pipeline Using the AWS CLI를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateComponent 섹션을 참조하세요.

`create-distribution-configuration`

다음 코드 예시에서는 `create-distribution-configuration` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

배포 구성 생성

다음 `create-distribution-configuration` 예시에서는 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 Setting Up and Managing an EC2 Image Builder Image Pipeline Using the AWS CLI를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateDistributionConfiguration 섹션을 참조하세요.

`create-image-pipeline`

다음 코드 예시에서는 `create-image-pipeline` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

이미지 파이프라인 생성

다음 `create-image-pipeline` 예시에서는 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 Setting Up and Managing an EC2 Image Builder Image Pipeline Using the AWS CLI를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateImagePipeline 섹션을 참조하세요.

`create-image-recipe`

다음 코드 예시에서는 `create-image-recipe` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

레시피 생성

다음 `create-image-recipe` 예시에서는 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.json`의 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 Setting Up and Managing an EC2 Image Builder Image Pipeline Using the AWS CLI를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateImageRecipe 섹션을 참조하세요.

`create-image`

다음 코드 예시에서는 `create-image` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

이미지 생성

다음 `create-image` 예시에서는 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 Setting Up and Managing an EC2 Image Builder Image Pipeline Using the AWS CLI를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateImage 섹션을 참조하세요.

`create-infrastructure-configuration`

다음 코드 예시에서는 `create-infrastructure-configuration` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

인프라 구성 생성

다음 `create-infrastructure-configuration` 예시에서는 JSON 파일을 사용하여 인프라 구성을 생성합니다.

```
aws imagebuilder create-infrastructure-configuration \
    --cli-input-json file://create-infrastructure-configuration.json
```

`create-infrastructure-configuration.json`의 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 Setting Up and Managing an EC2 Image Builder Image Pipeline Using the AWS CLI를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateInfrastructureConfiguration 섹션을 참조하세요.

`delete-component`

다음 코드 예시에서는 `delete-component` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

구성 요소 삭제

다음 `delete-component` 예시에서는 ARN을 지정하여 구성 요소 빌드 버전을 삭제합니다.

```
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 [Setting Up and Managing an EC2 Image Builder Image Pipeline Using the AWS CLI](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteComponent](#) 섹션을 참조하세요.

`delete-image-pipeline`

다음 코드 예시에서는 `delete-image-pipeline` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

이미지 파이프라인 삭제

다음 `delete-image-pipeline` 예시에서는 ARN을 지정하여 이미지 파이프라인을 삭제합니다.

```
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 Setting Up and Managing an EC2 Image Builder Image Pipeline Using the AWS CLI를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteImagePipeline 섹션을 참조하세요.

`delete-image-recipe`

다음 코드 예시에서는 `delete-image-recipe` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

이미지 레시피를 삭제하는 방법

다음 `delete-image-recipe` 예시에서는 ARN을 지정하여 이미지 레시피를 삭제합니다.

```
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 Setting Up and Managing an EC2 Image Builder Image Pipeline Using the AWS CLI를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteImageRecipe 섹션을 참조하세요.

`delete-image`

다음 코드 예시에서는 `delete-image` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

이미지 삭제

다음 `delete-image` 예시에서는 ARN을 지정하여 이미지 빌드 버전을 삭제합니다.

```
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 Setting Up and Managing an EC2 Image Builder Image Pipeline Using the AWS CLI를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteImage 섹션을 참조하세요.

`delete-infrastructure-configuration`

다음 코드 예시에서는 `delete-infrastructure-configuration` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

인프라 구성 삭제

다음 `delete-infrastructure-configuration` 예시에서는 ARN을 지정하여 이미지 파이프라인을 삭제합니다.

```
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 Setting Up and Managing an EC2 Image Builder Image Pipeline Using the AWS CLI를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteInfrastructureConfiguration 섹션을 참조하세요.

get-component-policy

다음 코드 예시에서는 get-component-policy 코드를 사용하는 방법을 보여줍니다.

AWS CLI

구성 요소 정책 세부 정보 가져오기

다음 get-component-policy 예시에서는 ARN을 지정하여 구성 요소 정책의 세부 정보를 나열합니다.

```
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 AWS CLI를 하여 EC2 Image Builder 이미지 파이프라인 설정 및 관리<https://docs.aws.amazon.com/imagebuilder/latest/userguide/managing-image-builder-cli.html>`__를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetComponentPolicy 섹션을 참조하세요.

get-component

다음 코드 예시에서는 `get-component` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

구성 요소 세부 정보 가져오기

다음 `get-component` 예시에서는 ARN을 지정하여 구성 요소의 세부 정보를 나열합니다.

```
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 Setting Up and Managing an EC2 Image Builder Image Pipeline Using the AWS CLI를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetComponent 섹션을 참조하세요.

get-distribution-configuration

다음 코드 예시에서는 get-distribution-configuration 코드를 사용하는 방법을 보여줍니다.

AWS CLI

배포 구성의 세부 정보를 가져오는 방법

다음 get-distribution-configuration 예시에서는 ARN을 지정하여 배포 구성의 세부 정보를 표시합니다.

```
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 Setting Up and Managing an EC2 Image Builder Image Pipeline Using the AWS CLI를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetDistributionConfiguration 섹션을 참조하세요.

get-image-pipeline

다음 코드 예시에서는 get-image-pipeline 코드를 사용하는 방법을 보여줍니다.

AWS CLI

이미지 파이프라인 세부 정보 가져오기

다음 get-image-pipeline 예시에서는 이미지의 ARN을 지정하여 이미지 파이프라인의 세부 정보를 나열합니다.

```
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 Setting Up and Managing an EC2 Image Builder Image Pipeline Using the AWS CLI를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetImagePipeline 섹션을 참조하세요.

get-image-policy

다음 코드 예시에서는 `get-image-policy` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

이미지 정책 세부 정보 가져오기

다음 `get-image-policy` 예시에서는 이미지의 ARN을 지정하여 이미지 정책의 세부 정보를 나열합니다.

```
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 Setting Up and Managing an EC2 Image Builder Image Pipeline Using the AWS CLI를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetImagePolicy 섹션을 참조하세요.

get-image-recipe-policy

다음 코드 예시에서는 `get-image-recipe-policy` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

이미지 레시피 정책 세부 정보 가져오기

다음 `get-image-recipe-policy` 예시에서는 이미지의 ARN을 지정하여 이미지 레시피 정책의 세부 정보를 나열합니다.

```
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 Setting Up and Managing an EC2 Image Builder Image Pipeline Using the AWS CLI를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetImageRecipePolicy 섹션을 참조하세요.

get-image

다음 코드 예시에서는 get-image 코드를 사용하는 방법을 보여줍니다.

AWS CLI

이미지 세부 정보를 가져오는 방법

다음 get-image 예시에서는 이미지의 ARN을 지정하여 이미지의 세부 정보를 나열합니다.

```
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 Setting Up and Managing an EC2 Image Builder Image Pipeline Using the AWS CLI를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetImage 섹션을 참조하세요.

get-infrastructure-configuration

다음 코드 예시에서는 get-infrastructure-configuration 코드를 사용하는 방법을 보여줍니다.

AWS CLI

인프라 구성 세부 정보 가져오기

다음 get-infrastructure-configuration 예시에서는 ARN을 지정하여 인프라 구성의 세부 정보를 나열합니다.

```
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 Setting Up and Managing an EC2 Image Builder Image Pipeline Using the AWS CLI를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetInfrastructureConfiguration 섹션을 참조하세요.

`import-component`

다음 코드 예시에서는 `import-component` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

구성 요소 가져오기

다음 `import-component` 예시에서는 JSON 파일을 사용하여 기존 스크립트를 가져옵니다.

```
aws imagebuilder import-component \
    --cli-input-json file://import-component.json
```

`import-component.json`의 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 Setting Up and Managing an EC2 Image Builder Image Pipeline Using the AWS CLI를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ImportComponent 섹션을 참조하세요.

`list-component-build-versions`

다음 코드 예시에서는 `list-component-build-versions` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

구성 요소 빌드 버전 나열

다음 `list-component-build-versions` 예시에서는 특정 의미 버전을 가진 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 Setting Up and Managing an EC2 Image Builder Image Pipeline Using the AWS CLI를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListComponentBuildVersions 섹션을 참조하세요.

`list-components`

다음 코드 예시에서는 `list-components` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

모든 구성 요소 시맨틱 버전을 나열하는 방법

다음 `list-components` 예시에는 액세스 권한이 있는 모든 구성 요소 의미 체계 버전이 나열됩니다. 본인이 소유한 구성 요소를 나열할지, Amazon이 소유한 구성 요소를 나열할지 또는 다른 계정에서 공유한 구성 요소를 나열할지 원하는 대로 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 Setting Up and Managing an EC2 Image Builder Image Pipeline Using the AWS CLI를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListComponents 섹션을 참조하세요.

list-distribution-configurations

다음 코드 예시에서는 `list-distribution-configurations` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

배포 나열

다음 `list-distribution-configurations` 예시에서는 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 Setting Up and Managing an EC2 Image Builder Image Pipeline Using the AWS CLI를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListDistributionConfigurations 섹션을 참조하세요.

`list-image-build-versions`

다음 코드 예시에서는 `list-image-build-versions` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

이미지 빌드 버전 나열

다음 `list-image-build-versions` 예시에서는 의미 체계 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 [Setting Up and Managing an EC2 Image Builder Image Pipeline Using the AWS CLI]를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ListImageBuildVersions] 섹션을 참조하세요.

list-image-pipeline-images

다음 코드 예시에서는 `list-image-pipeline-images` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

이미지 파이프라인 이미지를 나열하는 방법

다음 `list-image-pipeline-images` 예시에서는 특정 이미지 파이프라인에서 생성된 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 [Setting Up and Managing an EC2 Image Builder Image Pipeline Using the AWS CLI](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ListImagePipelineImages](#) 섹션을 참조하세요.

`list-image-recipes`

다음 코드 예시에서는 `list-image-recipes` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

이미지 레시피 나열

다음 `list-image-recipes` 예시에서는 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 Setting Up and Managing an EC2 Image Builder Image Pipeline Using the AWS CLI를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListImageRecipes 섹션을 참조하세요.

list-images

다음 코드 예시에서는 `list-images` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

이미지를 나열하는 방법

다음 `list-images` 예시에서는 액세스할 수 있는 모든 의미 체계 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 Setting Up and Managing an EC2 Image Builder Image Pipeline Using the AWS CLI를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListImages를 참조하세요.

`list-infrastructure-configurations`

다음 코드 예시에서는 `list-infrastructure-configurations` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

인프라 구성 나열

다음 `list-infrastructure-configurations` 예시에서는 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 Setting Up and Managing an EC2 Image Builder Image Pipeline Using the AWS CLI를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListInfrastructureConfigurations 섹션을 참조하세요.

list-tags-for-resource

다음 코드 예시에서는 `list-tags-for-resource` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

특정 리소스에 대한 태그 나열

다음 `list-tags-for-resource` 예시에서는 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 Setting Up and Managing an EC2 Image Builder Image Pipeline Using the AWS CLI를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource를 참조하세요.

put-component-policy

다음 코드 예시에서는 `put-component-policy` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

구성 요소에 리소스 정책을 적용하는 방법

다음 put-component-policy 명령은 빌드 구성 요소에 리소스 정책을 적용하여 빌드 구성 요소의 계정 간 공유를 활성화합니다. RAM CLI 명령 create-resource-share를 사용하는 것이 좋습니다. 리소스가 공유되는 모든 보안 주체에게 리소스가 표시되도록 하려면 EC2 Image Builder CLI 명령 put-component-policy를 사용하는 경우 RAM CLI 명령 promote-resource-share-create-from-policy도 사용해야 합니다.

```
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 Setting Up and Managing an EC2 Image Builder Image Pipeline Using the AWS CLI를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutComponentPolicy 섹션을 참조하세요.

put-image-policy

다음 코드 예시에서는 put-image-policy 코드를 사용하는 방법을 보여줍니다.

AWS CLI

이미지에 리소스 정책을 적용하는 방법

다음 put-image-policy 명령은 이미지에 리소스 정책을 적용하여 이미지의 계정 간 공유를 활성화합니다. RAM CLI 명령 create-resource-share를 사용하는 것이 좋습니다. 리소스가 공유되는 모든 보안 주체에게 리소스가 표시되도록 하려면 EC2 Image Builder CLI 명령 put-image-policy를 사용하는 경우, RAM CLI 명령 promote-resource-share-create-from-policy도 함께 사용해야 합니다.

```
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 Setting Up and Managing an EC2 Image Builder Image Pipeline Using the AWS CLI를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutImagePolicy 섹션을 참조하세요.

put-image-recipe-policy

다음 코드 예시에서는 `put-image-recipe-policy` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

이미지 레시피에 리소스 정책을 적용하는 방법

다음 `put-image-recipe-policy` 명령은 이미지 레시피에 리소스 정책을 적용하여 계정 간 이미지 레시피를 공유할 수 있도록 합니다. RAM CLI 명령 `create-resource-share`을 사용하는 것이 좋습니다. 리소스가 공유되는 모든 보안 주체에게 리소스가 표시되도록 하려면 EC2 Image Builder CLI 명령 `put-image-recipe-policy`를 사용하는 경우 RAM CLI 명령 `promote-resource-share-create-from-policy`도 사용해야 합니다.

```
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 Setting Up and Managing an EC2 Image Builder Image Pipeline Using the AWS CLI를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutImageRecipePolicy 섹션을 참조하세요.

`start-image-pipeline-execution`

다음 코드 예시에서는 `start-image-pipeline-execution` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

이미지 파이프라인을 수동으로 시작하는 방법

다음 `start-image-pipeline-execution` 예시에서는 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 Setting Up and Managing an EC2 Image Builder Image Pipeline Using the AWS CLI를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartImagePipelineExecution 섹션을 참조하세요.

tag-resource

다음 코드 예시에서는 tag-resource 코드를 사용하는 방법을 보여줍니다.

AWS CLI

리소스에 태그를 지정하는 방법

다음 tag-resource 예시에서는 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 Setting Up and Managing an EC2 Image Builder Image Pipeline Using the AWS CLI를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

untag-resource

다음 코드 예시에서는 untag-resource의 사용 방법을 보여줍니다.

AWS CLI

리소스에서 태그 제거

다음 `untag-resource` 예시에서는 JSON 파일을 사용하여 리소스에서 태그를 제거합니다.

```
aws imagebuilder untag-resource \
    --cli-input-json file://tag-resource.json
```

`untag-resource.json`의 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 Setting Up and Managing an EC2 Image Builder Image Pipeline Using the AWS CLI를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

`update-distribution-configuration`

다음 코드 예시에서는 `update-distribution-configuration` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

배포 구성 업데이트

다음 `update-distribution-configuration` 예시에서는 JSON 파일을 사용하여 배포 구성을 업데이트합니다.

```
aws imagebuilder update-distribution-configuration \
    --cli-input-json file://update-distribution-configuration.json
```

`update-distribution-configuration.json`의 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 Setting Up and Managing an EC2 Image Builder Image Pipeline Using the AWS CLI를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateDistributionConfiguration 섹션을 참조하세요.

`update-image-pipeline`

다음 코드 예시에서는 `update-image-pipeline` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

이미지 파이프라인 업데이트

다음 `update-image-pipeline` 예시에서는 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 Setting Up and Managing an EC2 Image Builder Image Pipeline Using the AWS CLI를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateImagePipeline 섹션을 참조하세요.

update-infrastructure-configuration

다음 코드 예시에서는 update-infrastructure-configuration 코드를 사용하는 방법을 보여줍니다.

AWS CLI

인프라 구성 업데이트

다음 `update-infrastructure-configuration` 예시에서는 JSON 파일을 사용하여 인프라 구성을 업데이트합니다.

```
aws imagebuilder update-infrastructure-configuration \
    --cli-input-json file:/update-infrastructure-configuration.json
```

`update-infrastructure-configuration.json`의 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 Setting Up and Managing an EC2 Image Builder Image Pipeline Using the AWS CLI를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateInfrastructureConfiguration 섹션을 참조하세요.

AWS CLI를 사용한 Incident Manager 예시

다음 코드 예시에서는 Incident Manager와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`create-replication-set`

다음 코드 예시에서는 `create-replication-set`의 사용 방법을 보여줍니다.

AWS CLI

복제 세트 생성

다음 `create-replication-set` 예시에서는 Incident Manager가 Amazon Web Services 계정의 데이터를 복제하고 암호화하는 데 사용하는 복제 세트를 생성합니다. 이 예시에서는 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 사용자 안내서의 Incident Manager 복제 세트 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateReplicationSet를 참조하세요.

create-response-plan

다음 코드 예시에서는 `create-response-plan`의 사용 방법을 보여줍니다.

AWS CLI

응답 계획 생성

다음 `create-response-plan` 예시에서는 지정된 세부 정보로 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 사용자 안내서의 인시던트 준비를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateResponsePlan을 참조합니다.

create-timeline-event

다음 코드 예시에서는 create-timeline-event의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 사용자 지정 타임라인 이벤트 생성

다음 create-timeline-event 예시에서는 지정된 인시던트에 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"
}
```

예시 2: 인시던트 메모를 사용하여 타임라인 이벤트 생성

다음 create-timeline-event 예시에서는 '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 사용자 안내서의 인시던트 세부 정보를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateTimelineEvent를 참조하세요.

delete-incident-record

다음 코드 예시에서는 delete-incident-record의 사용 방법을 보여줍니다.

AWS CLI

인시던트 레코드 삭제

다음 delete-incident-record 예시에서는 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 사용자 안내서의 인시던트 추적을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteIncidentRecord를 참조하세요.

delete-replication-set

다음 코드 예시에서는 delete-replication-set의 사용 방법을 보여줍니다.

AWS CLI

복제 세트 삭제

다음 delete-replication-set 예시에서는 Amazon Web Services 계정에서 복제 세트를 삭제합니다. 복제 세트를 삭제하면 모든 Incident Manager 데이터도 삭제됩니다. 이 작업은 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 사용자 안내서의 Incident Manager 복제 세트 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteReplicationSet를 참조하세요.

delete-resource-policy

다음 코드 예시에서는 `delete-resource-policy`의 사용 방법을 보여줍니다.

AWS CLI

리소스 정책 삭제

다음 `delete-resource-policy` 예시에서는 응답 계획에서 리소스 정책을 삭제합니다. 이렇게 하면 응답 계획이 공유된 위탁자 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 사용자 안내서의 공유 연락처 및 응답 계획 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteResourcePolicy를 참조하세요.

delete-response-plan

다음 코드 예시에서는 `delete-response-plan`의 사용 방법을 보여줍니다.

AWS CLI

응답 계획 삭제

다음 `delete-response-plan` 예시에서는 지정된 응답 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 사용자 안내서의 인시던트 준비를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteResponsePlan을 참조합니다.

delete-timeline-event

다음 코드 예시에서는 `delete-timeline-event`의 사용 방법을 보여줍니다.

AWS CLI

타임라인 이벤트 삭제

다음 `delete-timeline-event` 예시에서는 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 사용자 안내서의 인시던트 세부 정보를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteTimelineEvent를 참조하세요.

get-incident-record

다음 코드 예시에서는 `get-incident-record`의 사용 방법을 보여줍니다.

AWS CLI

인시던트 레코드 가져오기

다음 `get-incident-record` 예시에서는 지정된 인시던트 레코드의 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 사용자 안내서의 [인시던트 세부 정보]를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [GetIncidentRecord]를 참조하세요.

get-replication-set

다음 코드 예시에서는 `get-replication-set`의 사용 방법을 보여줍니다.

AWS CLI

복제 세트 가져오기

다음 `get-replication-set` 예시에서는 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 사용자 안내서의 Incident Manager 복제 세트 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetReplicationSet를 참조하세요.

get-resource-policies

다음 코드 예시에서는 get-resource-policies의 사용 방법을 보여줍니다.

AWS CLI

응답 계획의 리소스 정책 나열

다음 command-name 예시에서는 지정된 응답 계획에 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 사용자 안내서의 공유 연락처 및 응답 계획 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetResourcePolicies를 참조하세요.

get-response-plan

다음 코드 예시에서는 get-response-plan의 사용 방법을 보여줍니다.

AWS CLI

응답 계획의 세부 정보 가져오기

다음 command-name 예시에서는 AWS 계정의 지정된 응답 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 사용자 안내서의 인시던트 준비를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetResponsePlan을 참조합니다.

get-timeline-event

다음 코드 예시에서는 get-timeline-event의 사용 방법을 보여줍니다.

AWS CLI

타임라인 이벤트의 세부 정보 가져오기

다음 get-timeline-event 예시에서는 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 사용자 안내서의 인시던트 세부 정보를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetTimelineEvent를 참조하세요.

`list-incident-records`

다음 코드 예시에서는 `list-incident-records`의 사용 방법을 보여줍니다.

AWS CLI

인시던트 레코드 나열

다음 `command-name` 예시에서는 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 사용자 안내서의 인시던트 목록을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListIncidentRecords를 참조하세요.

`list-related-items`

다음 코드 예시에서는 `list-related-items`의 사용 방법을 보여줍니다.

AWS CLI

관련 항목 나열

다음 `list-related-items` 예시에서는 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 사용자 안내서의 인시던트 세부 정보를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListRelatedItems를 참조하세요.

list-replication-sets

다음 코드 예시에서는 `list-replication-sets`의 사용 방법을 보여줍니다.

AWS CLI

복제 세트 나열

다음 `list-replication-set` 예시에서는 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 사용자 안내서의 Incident Manager 복제 세트 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListReplicationSets를 참조하세요.

list-response-plans

다음 코드 예시에서는 `list-response-plans`의 사용 방법을 보여줍니다.

AWS CLI

사용 가능한 응답 계획 나열

다음 `list-response-plans` 예시에서는 Amazon Web Services 계정의 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 사용자 안내서의 인시던트 준비를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListResponsePlans를 참조하세요.

`list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`의 사용 방법을 보여줍니다.

AWS CLI

응답 계획의 태그 나열

다음 `list-tags-for-resource` 예시에서는 지정된 응답 계획에 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 사용자 안내서의 [태그 지정](#)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ListTagsForResource](#)를 참조하세요.

`list-timeline-events`

다음 코드 예시에서는 `list-timeline-events`의 사용 방법을 보여줍니다.

AWS CLI

인시던트의 타임라인 이벤트 나열

다음 `command-name` 예시에서는 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 사용자 안내서의 인시던트 세부 정보를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTimelineEvents를 참조하세요.

put-resource-policy

다음 코드 예시에서는 put-resource-policy의 사용 방법을 보여줍니다.

AWS CLI

응답 계획 및 인시던트 공유

다음 command-name 예시에서는 지정된 위탁자와 응답 계획 및 관련 인시던트를 공유하는 리소스 정책을 Example-Response-Plan에 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::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/*\"]}]}"
```

출력:

```
{
    "policyId": "be8b57191f0371f1c6827341aa3f0a03"
}
```

자세한 내용은 Incident Manager 사용자 안내서의 공유 연락처 및 응답 계획 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutResourcePolicy를 참조하세요.

start-incident

다음 코드 예시에서는 start-incident의 사용 방법을 보여줍니다.

AWS CLI

인시던트 시작

다음 start-incident 예시에서는 지정된 응답 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 사용자 안내서의 인시던트 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartIncident를 참조하세요.

`tag-resource`

다음 코드 예시에서는 `tag-resource`의 사용 방법을 보여줍니다.

AWS CLI

응답 계획에 태그 지정

다음 `tag-resource` 예시에서는 지정된 응답 계획에 제공된 태그 키와 값 페어로 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 사용자 안내서의 태그 지정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

`untag-resource`

다음 코드 예시에서는 `untag-resource`의 사용 방법을 보여줍니다.

AWS CLI

리소스에서 태그 제거

다음 `untag-resource` 예시에서는 지정된 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 사용자 안내서의 태그 지정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

update-deletion-protection

다음 코드 예시에서는 `update-deletion-protection`의 사용 방법을 보여줍니다.

AWS CLI

복제 세트 삭제 보호 업데이트

다음 `update-deletion-protection` 예시에서는 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 사용자 안내서의 Incident Manager 복제 세트 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateDeletionProtection을 참조하세요.

update-incident-record

다음 코드 예시에서는 `update-incident-record`의 사용 방법을 보여줍니다.

AWS CLI

인시던트 레코드 업데이트

다음 command-name 예시에서는 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 사용자 안내서의 인시던트 세부 정보를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateIncidentRecord를 참조하세요.

update-related-items

다음 코드 예시에서는 update-related-items의 사용 방법을 보여줍니다.

AWS CLI

인시던트 관련 항목 업데이트

다음 update-related-item 예시에서는 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 사용자 안내서의 인시던트 세부 정보를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateRelatedItems를 참조하세요.

update-replication-set

다음 코드 예시에서는 update-replication-set의 사용 방법을 보여줍니다.

AWS CLI

복제 세트 업데이트

다음 command-name 예시에서는 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 사용자 안내서의 Incident Manager 복제 세트 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateReplicationSet를 참조하세요.

update-response-plan

다음 코드 예시에서는 update-response-plan의 사용 방법을 보여줍니다.

AWS CLI

응답 계획 업데이트

다음 update-response-plan 예시에서는 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 사용자 안내서의 인시던트 준비를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateResponsePlan을 참조하세요.

update-timeline-event

다음 코드 예시에서는 update-timeline-event의 사용 방법을 보여줍니다.

AWS CLI

타임라인 이벤트 업데이트

다음 update-timeline-event 예시에서는 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 사용자 안내서의 인시던트 세부 정보를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateTimelineEvent를 참조하세요.

AWS CLI를 사용한 Incident Manager 연락처 예시

다음 코드 예시에서는 Incident Manager 연락처와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

accept-page

다음 코드 예시에서는 accept-page의 사용 방법을 보여줍니다.

AWS CLI

상담 중 호출 수락

다음 accept-page 예시에서는 문의 채널로 전송된 수락 코드를 사용하여 호출을 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 사용자 안내서의 연락처를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AcceptPage를 참조하세요.

activate-contact-channel

다음 코드 예시에서는 activate-contact-channel의 사용 방법을 보여줍니다.

AWS CLI

연락처의 문의 채널 활성화

다음 activate-contact-channel 예시에서는 문의 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 사용자 안내서의 연락처를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ActivateContactChannel을 참조하세요.

command-name

다음 코드 예시에서는 command-name의 사용 방법을 보여줍니다.

AWS CLI

연락처 삭제

다음 command-name 예시에서는 연락처를 삭제합니다. 이 연락처는 더 이상 해당 연락처와 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 사용자 안내서의 연락처를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CommandName을 참조하세요.

create-contact-channel

다음 코드 예시에서는 create-contact-channel의 사용 방법을 보여줍니다.

AWS CLI

문의 채널 생성

연락처 Akua Mansa를 위한 SMS 유형의 문의 채널을 생성합니다. 문의 채널은 SMS, EMAIL 또는 VOICE 유형으로 생성할 수 있습니다.

```
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 사용자 안내서의 연락처를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateContactChannel을 참조하세요.

create-contact

다음 코드 예시에서는 create-contact의 사용 방법을 보여줍니다.

AWS CLI

연락처 생성

다음 create-contact 예시에서는 계획이 비어 있는 환경에 연락처를 생성합니다. 계획은 문의 채널을 생성한 후 업데이트할 수 있습니다. create-contact-channel 명령을 이 명령의 출력 ARN과 함께 사용합니다. 이 연락처의 문의 채널을 생성한 후 update-contact를 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 사용자 안내서의 연락처를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateContact를 참조하세요.

deactivate-contact-channel

다음 코드 예시에서는 deactivate-contact-channel의 사용 방법을 보여줍니다.

AWS CLI

문의 채널 비활성화

다음 deactivate-contact-channel 예시에서는 문의 채널을 비활성화합니다. 문의 채널을 비활성화하면 인시던트 발생 시 해당 문의 채널이 더 이상 호출되지 않습니다. 언제든지 activate-contact-channel 명령을 사용하여 문의 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 사용자 안내서의 연락처를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DeactivateContactChannel]을 참조하세요.

delete-contact-channel

다음 코드 예시에서는 delete-contact-channel의 사용 방법을 보여줍니다.

AWS CLI

문의 채널 삭제

다음 delete-contact-channel 예시에서는 문의 채널을 삭제합니다. 문의 채널을 삭제하면 인시던트 발생 시 해당 문의 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 사용자 안내서의 연락처를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteContactChannel을 참조하세요.

delete-contact

다음 코드 예시에서는 delete-contact의 사용 방법을 보여줍니다.

AWS CLI

연락처 삭제

다음 delete-contact 예시에서는 연락처를 삭제합니다. 이 연락처는 더 이상 해당 연락처와 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 사용자 안내서의 연락처를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteContact를 참조하세요.

describe-engagement

다음 코드 예시에서는 describe-engagement의 사용 방법을 보여줍니다.

AWS CLI

상담의 세부 정보 설명

다음 describe-engagement 예시에서는 연락처 또는 에스컬레이션 계획의 상담 세부 정보를 나열합니다. 주제와 콘텐츠가 문의 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 사용자 안내서의 연락처를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeEngagement를 참조하세요.

describe-page

다음 코드 예시에서는 describe-page의 사용 방법을 보여줍니다.

AWS CLI

문의 채널로 전송된 호출의 세부 정보 나열

다음 describe-page 예시에서는 문의 채널로 전송된 호출의 세부 정보를 나열합니다. 호출에는 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 사용자 안내서의 연락처를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribePage를 참조하세요.

get-contact-channel

다음 코드 예시에서는 get-contact-channel의 사용 방법을 보여줍니다.

AWS CLI

문의 채널의 세부 정보 나열

다음 get-contact-channel 예시에서는 문의 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 사용자 안내서의 연락처를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetContactChannel을 참조하세요.

get-contact-policy

다음 코드 예시에서는 get-contact-policy의 사용 방법을 보여줍니다.

AWS CLI

연락처의 리소스 정책 나열

다음 get-contact-policy 예시에서는 지정된 연락처에 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 사용자 안내서의 공유 연락처 및 응답 계획 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetContactPolicy를 참조하세요.

get-contact

다음 코드 예시에서는 get-contact의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 연락처 계획 설명

다음 get-contact 예시에서는 연락처를 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
                        }
                    }
                ]
            }
        ]
    }
}
```

예시 2: 에스컬레이션 계획 설명

다음 `get-contact` 예시에서는 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 사용자 안내서의 연락처를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetContact를 참조하세요.

list-contact-channels

다음 코드 예시에서는 list-contact-channels의 사용 방법을 보여줍니다.

AWS CLI

연락처의 문의 채널 나열

다음 list-contact-channels 예시에서는 지정된 연락처의 사용 가능한 문의 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 사용자 안내서의 연락처를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListContactChannels를 참조하세요.

list-contacts

다음 코드 예시에서는 `list-contacts`의 사용 방법을 보여줍니다.

AWS CLI

모든 에스컬레이션 계획 및 연락처 나열

다음 `list-contacts` 예시에서는 계정의 연락처 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 사용자 안내서의 연락처를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListContacts을 참조하세요.

list-engagements

다음 코드 예시에서는 `list-engagements`의 사용 방법을 보여줍니다.

AWS CLI

모든 상담 나열

다음 `list-engagements` 예시에서는 에스컬레이션 계획 및 연락처의 상담을 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 사용자 안내서의 연락처를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListEngagements를 참조하세요.

list-page-receipts

다음 코드 예시에서는 list-page-receipts의 사용 방법을 보여줍니다.

AWS CLI

호출 수신 나열

다음 command-name 예시에서는 연락처에서 호출을 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 사용자 안내서의 연락처를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListPageReceipts를 참조하세요.

list-pages-by-contact

다음 코드 예시에서는 list-pages-by-contact의 사용 방법을 보여줍니다.

AWS CLI

연락처별 호출 나열

다음 list-pages-by-contact 예시에서는 지정된 연락처의 모든 호출을 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 사용자 안내서의 연락처를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListPagesByContact를 참조하세요.

`list-pages-by-engagement`

다음 코드 예시에서는 `list-pages-by-engagement`의 사용 방법을 보여줍니다.

AWS CLI

상담이 시작된 문의 채널로 전송된 호출 나열

다음 `list-pages-by-engagement` 예시에서는 정의된 상담 계획에 참여하는 동안 발생한 호출을 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 사용자 안내서의 연락처를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListPagesByEngagement를 참조하세요.

list-tags-for-resource

다음 코드 예시에서는 list-tags-for-resource의 사용 방법을 보여줍니다.

AWS CLI

연락처의 태그 나열

다음 list-tags-for-resource 예시에서는 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 사용자 안내서의 태그 지정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource를 참조하세요.

put-contact-policy

다음 코드 예시에서는 put-contact-policy의 사용 방법을 보여줍니다.

AWS CLI

연락처 및 상담 공유

다음 put-contact-policy 예시에서는 연락처 및 관련 상담을 위탁자와 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 사용자 안내서의 공유 연락처 및 응답 계획 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutContactPolicy를 참조하세요.

send-activation-code

다음 코드 예시에서는 send-activation-code의 사용 방법을 보여줍니다.

AWS CLI

활성화 코드 전송

다음 send-activation-code 예시에서는 활성화 코드와 메시지를 지정된 문의 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 사용자 안내서의 연락처를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 SendActivationCode를 참조하세요.

`start-engagement`

다음 코드 예시에서는 `start-engagement`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 연락처의 문의 채널 호출

다음 `start-engagement`는 연락처의 문의 채널을 호출합니다. sender, subject, public-subject 및 public-content에는 필드가 없습니다. Incident Manager는 제공된 VOICE 또는 EMAIL 문의 채널로 subject 및 content를 전송합니다. Incident Manager는 제공된 SMS 문의 채널로 public-subject 및 public-content를 전송합니다. sender는 상담을 시작한 사람을 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 사용자 안내서의 연락처를 참조하세요.

예시 2: 제공된 에스컬레이션 계획에서 연락처 호출

다음 `start-engagement`은 에스컬레이션 계획을 통해 연락처와 상담합니다. 각 연락처는 상담 계획에 따라 호출됩니다.

```
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 사용자 안내서의 연락처를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartEngagement를 참조하세요.

`stop-engagement`

다음 코드 예시에서는 `stop-engagement`의 사용 방법을 보여줍니다.

AWS CLI

상담 중지

다음 `stop-engagement` 예시에서는 연락처 및 문의 채널을 더 이상 호출하지 못하도록 상담을 중지합니다.

```
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 사용자 안내서의 연락처를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StopEngagement를 참조하세요.

tag-resource

다음 코드 예시에서는 tag-resource의 사용 방법을 보여줍니다.

AWS CLI

연락처에 태그 지정

다음 tag-resource 예시에서는 지정된 연락처에 제공된 태그 키 값 페어로 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 사용자 안내서의 태그 지정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

untag-resource

다음 코드 예시에서는 untag-resource의 사용 방법을 보여줍니다.

AWS CLI

연락처에서 태그 제거

다음 untag-resource 예시에서는 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 사용자 안내서의 태그 지정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

update-contact-channel

다음 코드 예시에서는 update-contact-channel의 사용 방법을 보여줍니다.

AWS CLI

문의 채널 업데이트

다음 update-contact-channel 예시에서는 문의 채널의 이름과 전달 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 사용자 안내서의 연락처를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateContactChannel을 참조하세요.

update-contact

다음 코드 예시에서는 update-contact의 사용 방법을 보여줍니다.

AWS CLI

연락처의 상담 계획 업데이트

다음 update-contact 예시에서는 세 가지 유형의 문의 채널을 포함하도록 연락처 Akua의 상담 계획을 업데이트합니다. 이는 Akua의 문의 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 사용자 안내서의 연락처를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateContact를 참조하세요.

AWS CLI를 사용한 Amazon Inspector 예시

다음 코드 예시에서는 Amazon Inspector에서 AWS Command Line Interface를 사용하여 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`add-attributes-to-findings`

다음 코드 예시에서는 `add-attributes-to-findings` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

결과에 속성을 추가하는 방법

다음 `add-attribute-to-finding` 명령은 `arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-8l1VIE0D/run/0-Z02cjjug/finding/0-T8yM9mEU`의 ARN을 사용하여 `Example`의 키와 값이 `example`인 속성을 결과에 할당합니다.

```
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 사용 설명서의 Amazon Inspector Findings를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AddAttributesToFindings 섹션을 참조하세요.

associate-member

다음 코드 예시는 associate-member의 사용 방법을 보여 줍니다.

AWS CLI

예: AWS 계정을 Amazon Inspector 위임된 관리자와 연결

다음 associate-member 예제에서는 AWS 계정을 Amazon Inspector 위임된 관리자와 연결합니다.

```
aws inspector2 associate-member \
    --account-id 123456789012
```

출력:

```
{
    "accountId": "123456789012"
}
```

자세한 내용은 Amazon Inspector 사용 설명서의 AWS Organizations로 Amazon Inspector에서 여러 계정 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AssociateMember 섹션을 참조하세요.

create-assessment-target

다음 코드 예시는 create-assessment-target의 사용 방법을 보여 줍니다.

AWS CLI

평가 대상 생성

다음 create-assessment-target 명령은 ARN이 arn:aws:inspector:us-west-2:123456789012:resourcegroup/0-AB6DMKnv인 리소스 그룹을 사용하여 ExampleAssessmentTarget이라는 평가 대상을 생성합니다.

```
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 사용 설명서의 Amazon Inspector Assessment Targets를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateAssessmentTarget 섹션을 참조하세요.

create-assessment-template

다음 코드 예시에서는 `create-assessment-template` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

평가 템플릿 생성

다음 `create-assessment-template` 명령은 ARN이 `arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX`인 평가 대상에 대해 `ExampleAssessmentTemplate`라는 평가 템플릿을 생성합니다.

```
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 설명서의 Amazon Inspector Assessment Templates and Assessment Runs를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateAssessmentTemplate 섹션을 참조하세요.

create-filter

다음 코드 예시에서는 create-filter 코드를 사용하는 방법을 보여줍니다.

AWS CLI

필터 생성

다음 create-filter 예시에서는 ECR 인스턴스 유형 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 Filtering Amazon Inspector findings를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateFilter 섹션을 참조하세요.

create-findings-report

다음 코드 예시에서는 create-findings-report 코드를 사용하는 방법을 보여줍니다.

AWS CLI

조사 결과 보고서를 생성하는 방법

다음 create-findings-report 예시에서는 조사 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 설명서]의 Managing findings in Amazon Inspector를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [CreateFindingsReport] 섹션을 참조하세요.

create-resource-group

다음 코드 예시에서는 create-resource-group 코드를 사용하는 방법을 보여줍니다.

AWS CLI

리소스 그룹 생성

다음 create-resource-group 명령은 Name의 태그 키와 example의 값을 사용하여 리소스 그룹을 생성합니다.

```
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 사용 설명서의 Amazon Inspector Assessment Targets를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [CreateResourceGroup](#) 섹션을 참조하세요.

`create-sbom-export`

다음 코드 예시에서는 `create-sbom-export` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

소프트웨어 재료표(SBOM) 보고서를 생성하는 방법

다음 `create-sbom-export` 예시에서는 소프트웨어 자재 명세서(SBOM) 보고서를 생성합니다.

```
aws inspector2 create-sbom-export \
    --report-format SPDX_2_3 \
    --resource-filter-criera
 '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 [Exporting SBOMs with Amazon Inspector](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [CreateSbomExport](#) 섹션을 참조하세요.

`delete-assessment-run`

다음 코드 예시에서는 `delete-assessment-run` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

평가 실행 삭제

다음 delete-assessment-run 명령은 arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-it5r2S4T/run/0-11LMTAVe의 ARN을 사용한 평가 실행을 삭제합니다.

```
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 설명서의 Amazon Inspector Assessment Templates and Assessment Runs를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteAssessmentRun 섹션을 참조하세요.

delete-assessment-target

다음 코드 예시에서는 delete-assessment-target 코드를 사용하는 방법을 보여줍니다.

AWS CLI

평가 대상 삭제

다음 delete-assessment-target 명령은 ARN이 arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq인 평가 대상을 삭제합니다.

```
aws inspector delete-assessment-target --assessment-target-arn arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq
```

자세한 내용은 Amazon Inspector 사용 설명서의 Amazon Inspector Assessment Targets를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteAssessmentTarget 섹션을 참조하세요.

delete-assessment-template

다음 코드 예시에서는 delete-assessment-template 코드를 사용하는 방법을 보여줍니다.

AWS CLI

평가 템플릿 삭제

다음 delete-assessment-template 명령은 ARN이 arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-it5r2S4T인 평가 템플릿을 삭제합니다.

```
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 설명서의 Amazon Inspector Assessment Templates and Assessment Runs를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteAssessmentTemplate 섹션을 참조하세요.

delete-filter

다음 코드 예시에서는 `delete-filter` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

필터 세트 삭제

다음 `delete-filter` 예시에서는 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 Filtering Amazon Inspector findings를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteFilter 섹션을 참조하세요.

describe-assessment-runs

다음 코드 예시에서는 `describe-assessment-runs` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

평가 실행을 설명하는 방법

다음 describe-assessment-run 명령은 arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-4r1V2mAw/run/0-MKkpXXPE의 ARN을 사용한 평가 실행을 설명합니다.

```
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 설명서의 Amazon Inspector Assessment Templates and Assessment Runs를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeAssessmentRuns 섹션을 참조하세요.

describe-assessment-targets

다음 코드 예시에서는 describe-assessment-targets 코드를 사용하는 방법을 보여줍니다.

AWS CLI

평가 대상을 설명하는 방법

다음 describe-assessment-targets 명령은 ARN이 arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq인 평가 대상을 설명합니다.

```
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 사용 설명서의 Amazon Inspector Assessment Targets를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeAssessmentTargets 섹션을 참조하세요.

describe-assessment-templates

다음 코드 예시에서는 `describe-assessment-templates` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

평가 템플릿을 설명하는 방법

다음 `describe-assessment-templates` 명령은 ARN이 `arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-4r1V2mAw`인 평가 템플릿을 설명합니다.

```
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 설명서의 Amazon Inspector Assessment Templates and Assessment Runs를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeAssessmentTemplates](#) 섹션을 참조하세요.

`describe-cross-account-access-role`

다음 코드 예시에서는 `describe-cross-account-access-role` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

교차 계정 액세스 역할을 설명하는 방법

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 사용 설명서의 Setting up Amazon Inspector를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeCrossAccountAccessRole 섹션을 참조하세요.

describe-findings

다음 코드 예시에서는 describe-findings 코드를 사용하는 방법을 보여줍니다.

AWS CLI

조사 결과를 설명하는 방법

다음 describe-findings 명령은 ARN이 arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-4r1V2mAw/run/0-MKkpXXPE/finding/0-HwPnsDm4인 조사 결과를 설명합니다.

```
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 사용 설명서의 Amazon Inspector Findings를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeFindings 섹션을 참조하세요.

`describe-resource-groups`

다음 코드 예시에서는 `describe-resource-groups` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

리소스 그룹을 설명하는 방법

다음 `describe-resource-groups` 명령은 ARN이 `arn:aws:inspector:us-west-2:123456789012:resourcegroup/0-PyGXopAI`인 리소스 그룹을 설명합니다.

```
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 사용 설명서의 Amazon Inspector Assessment Targets를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeResourceGroups 섹션을 참조하세요.

describe-rules-packages

다음 코드 예시에서는 describe-rules-packages 코드를 사용하는 방법을 보여줍니다.

AWS CLI

규칙 패키지를 설명하는 방법

다음 describe-rules-packages 명령은 ARN이 arn:aws:inspector:us-west-2:758058086616:rulespackage/0-9hgA516p인 규칙 패키지를 설명합니다.

```
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 설명서의 Amazon Inspector Rules Packages and Rules를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeRulesPackages 섹션을 참조하세요.

`disassociate-member`

다음 코드 예시는 `disassociate-member`의 사용 방법을 보여 줍니다.

AWS CLI

예: Amazon Inspector 위임된 관리자에서 멤버 계정의 연결을 해제

다음 `disassociate-member` 예제에서는 Amazon Inspector 위임된 관리자로부터 AWS 계정의 연결을 해제합니다.

```
aws inspector2 disassociate-member \
    --account-id 123456789012
```

출력:

```
{
```

```
    "accountId": "123456789012"
}
```

자세한 내용은 Amazon Inspector 사용 설명서의 AWS Organizations로 Amazon Inspector에서 여러 계정 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisassociateMember 섹션을 참조하세요.

get-configuration

다음 코드 예시는 get-configuration의 사용 방법을 보여 줍니다.

AWS CLI

Inspector 스캔에 대한 설정 구성을 가져오는 방법

다음 get-configuration 예시에서는 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 Automated resource scanning with Amazon Inspector를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [GetConfiguration](https://docs.aws.amazon.com/goto/aws-cli/inspector2-2020-06-08/GetConfiguration) 섹션을 참조하세요.

get-member

다음 코드 예시는 get-member의 사용 방법을 보여 줍니다.

AWS CLI

예제: 조직의 멤버 정보 가져오기

aws Inspector2 get-member --account-id 123456789012

출력:

```
{
        "member": {
        "accountId": "123456789012",
        "delegatedAdminAccountId": "123456789012",
        "relationshipStatus": "ENABLED",
        "updatedAt": "2023-09-11T09:57:20.520000-07:00"
    }
}
```

자세한 내용은 Amazon Inspector 사용 설명서의 AWS Organizations로 Amazon Inspector에서 여러 계정 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetMember 섹션을 참조하세요.

get-telemetry-metadata

다음 코드 예시는 get-telemetry-metadata의 사용 방법을 보여 줍니다.

AWS CLI

원격 측정 메타데이터를 가져오는 방법

다음 get-telemetry-metadata 명령은 arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-4r1V2mAw/run/0-MKkpXXPE의 ARN을 사용하여 평가 실행을 위해 수집된 데이터에 대한 정보를 생성합니다.

```
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"
            }
        ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetTelemetryMetadata 섹션을 참조하세요.

list-account-permissions

다음 코드 예시에서는 list-account-permissions 코드를 사용하는 방법을 보여줍니다.

AWS CLI

계정 권한을 나열하는 방법

다음 list-account-permissions 예시에서는 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 Identity and Access Management for Amazon Inspector를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListAccountPermissions 섹션을 참조하세요.

`list-assessment-run-agents`

다음 코드 예시에서는 `list-assessment-run-agents` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

평가 실행 에이전트를 나열하는 방법

다음 `list-assessment-run-agents` 명령은 지정된 ARN으로 실행되는 평가의 에이전트를 나열합니다.

```
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 Agents를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListAssessmentRunAgents 섹션을 참조하세요.

list-assessment-runs

다음 코드 예시에서는 list-assessment-runs 코드를 사용하는 방법을 보여줍니다.

AWS CLI

평가 실행을 나열하는 방법

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 Assessment Templates and Assessment Runs를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListAssessmentRuns 섹션을 참조하세요.

list-assessment-targets

다음 코드 예시에서는 list-assessment-targets 코드를 사용하는 방법을 보여줍니다.

AWS CLI

평가 대상을 나열하는 방법

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 사용 설명서의 Amazon Inspector Assessment Targets를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListAssessmentTargets 섹션을 참조하세요.

list-assessment-templates

다음 코드 예시에서는 `list-assessment-templates` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

평가 템플릿을 나열하는 방법

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 설명서의 Amazon Inspector Assessment Templates and Assessment Runs를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListAssessmentTemplates 섹션을 참조하세요.

list-coverage-statistics

다음 코드 예시에서는 `list-coverage-statistics` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

예시 1: 그룹별로 적용 범위 통계를 나열하는 방법

다음 `list-coverage-statistics` 예시에서는 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 [Assessing Amazon Inspector coverage of your AWS environment](#)를 참조하세요.

예시 2: 리소스 유형별로 적용 범위 통계를 나열하는 방법

다음 list-coverage-statistics 예시에서는 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 [Assessing Amazon Inspector coverage of your AWS environment](#)를 참조하세요.

예시 3: ECR 리포지토리 이름별로 적용 범위 통계를 나열하는 방법

다음 list-coverage-statistics 예시에서는 AWS 환경의 적용 범위 통계를 ECR 리포지토리 이름별로 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 [Assessing Amazon Inspector coverage of your AWS environment](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ListCoverageStatistics](#) 섹션을 참조하세요.

list-coverage

다음 코드 예시에서는 `list-coverage` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

예시 1: 환경에 대한 적용 범위 세부 정보를 나열하는 방법

다음 `list-coverage` 예시에서는 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"
        }
```

```
    ]
}
```

예시 2: Lambda 함수 리소스 유형에 대한 적용 범위 세부 정보를 나열하는 방법

다음 `list-coverage` 예시에서는 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [ListCoverage](#) 섹션을 참조하세요.

`list-delegated-admin-accounts`

다음 코드 예시에서는 `list-delegated-admin-accounts` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

조직의 위임된 관리자 계정에 대한 정보를 나열하는 방법

다음 `list-delegated-admin-accounts` 예시에서는 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 Designating a delegated administrator for Amazon Inspector를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListDelegatedAdminAccounts 섹션을 참조하세요.

list-event-subscriptions

다음 코드 예시에서는 `list-event-subscriptions` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

이벤트 구독을 나열하는 방법

다음 `list-event-subscriptions` 명령은 ARN이 `arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-7sbz2Kz0`인 평가 템플릿의 모든 이벤트 구독을 나열합니다.

```
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 설명서의 Amazon Inspector Assessment Templates and Assessment Runs를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ListEventSubscriptions](https://docs.aws.amazon.com/goto/aws-cli/inspector-2016-02-16/ListEventSubscriptions) 섹션을 참조하세요.

list-filters

다음 코드 예시에서는 `list-filters` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Amazon Inspector를 활성화하는 데 사용한 계정과 연결된 필터를 나열하는 방법

다음 `list-filters` 예시에서는 Amazon Inspector를 활성화하는 데 사용한 계정과 관련된 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 Filtering Amazon Inspector findings를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListFilters 섹션을 참조하세요.

list-findings

다음 코드 예시에서는 `list-findings` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

조사 결과를 나열하는 방법

다음 `list-findings` 명령은 생성된 모든 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 사용 설명서의 Amazon Inspector Findings를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListFindings 섹션을 참조하세요.

list-members

다음 코드 예시는 `list-members`의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 조직의 Amazon Inspector 위임된 관리자와 연결된 모든 멤버 계정을 나열

aws Inspector2 list-members --only-associated

출력:

```
{
        {
             "members": [
```

```
        {
            "accountId": "123456789012",
            "delegatedAdminAccountId": "123456789012",
            "relationshipStatus": "ENABLED",
            "updatedAt": "2023-09-11T09:57:20.520000-07:00"
        },
        {
            "accountId": "123456789012",
            "delegatedAdminAccountId": "123456789012",
            "relationshipStatus": "ENABLED",
            "updatedAt": "2024-08-12T10:13:01.472000-07:00"
        },
        {
            "accountId": "625032911453",
            "delegatedAdminAccountId": "123456789012",
            "relationshipStatus": "ENABLED",
            "updatedAt": "2023-09-11T09:57:20.438000-07:00"
        },
        {
           "accountId": "715411239211",
           "delegatedAdminAccountId": "123456789012",
           "relationshipStatus": "ENABLED",
           "updatedAt": "2024-04-24T09:14:57.471000-07:00"
        }
    ]
}
```

자세한 내용은 Amazon Inspector 사용 설명서의 AWS Organizations로 Amazon Inspector에서 여러 계정 관리를 참조하세요.

예제 2: 조직의 Amazon Inspector 위임된 관리자와 연결 및 연결 해제된 모든 멤버 계정을 나열

aws Inspector2 list-members --no-only-associated

출력:

```
{
        {
           "members": [
        {
           "accountId": "123456789012",
           "delegatedAdminAccountId": "123456789012",
           "relationshipStatus": "REMOVED",
```

```
            "updatedAt": "2024-05-15T11:34:53.326000-07:00"
        },
        {
            "accountId": "123456789012",
            "delegatedAdminAccountId": "123456789012",
            "relationshipStatus": "ENABLED",
            "updatedAt": "2023-09-11T09:57:20.520000-07:00"
        },
        {
            "accountId": "123456789012",
            "delegatedAdminAccountId": "123456789012",
            "relationshipStatus": "ENABLED",
            "updatedAt": "2024-08-12T10:13:01.472000-07:00"
        },
        {
            "accountId": "123456789012",
            "delegatedAdminAccountId": "123456789012",
            "relationshipStatus": "ENABLED",
            "updatedAt": "2023-09-11T09:57:20.438000-07:00"
        },
        {
            "accountId": "123456789012",
            "delegatedAdminAccountId": "123456789012",
            "relationshipStatus": "ENABLED",
            "updatedAt": "2024-04-24T09:14:57.471000-07:00"
        }
    ]
}
```

자세한 내용은 Amazon Inspector 사용 설명서의 AWS Organizations로 Amazon Inspector에서 여러 계정 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListMembers를 참조하세요.

`list-rules-packages`

다음 코드 예시는 `list-rules-packages`의 사용 방법을 보여 줍니다.

AWS CLI

규칙 패키지를 나열하는 방법

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 설명서의 Amazon Inspector Rules Packages and Rules를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListRulesPackages 섹션을 참조하세요.

list-tags-for-resource

다음 코드 예시에서는 `list-tags-for-resource` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

리소스 태그 나열

다음 `list-tags-for-resource` 명령은 ARN이 `arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-gcwFliYu`인 평가 템플릿과 연결된 모든 태그를 나열합니다.

```
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 설명서의 Amazon Inspector Assessment Templates and Assessment Runs를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource를 참조하세요.

`list-usage-totals`

다음 코드 예시에서는 `list-usage-totals` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

지난 30일 동안의 사용량 합계를 나열하는 방법

다음 `list-usage-totals` 예시에서는 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 Monitoring usage and cost in Amazon Inspector를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListUsageTotals 섹션을 참조하세요.

preview-agents

다음 코드 예시에서는 `preview-agents` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

에이전트를 미리 보는 방법

다음 `preview-agents` 명령은 ARN이 `arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq`인 평가 대상의 일부인 EC2 인스턴스에 설치된 에이전트를 미리 봅니다.

```
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 사용 설명서의 Amazon Inspector Assessment Targets를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PreviewAgents 섹션을 참조하세요.

register-cross-account-access-role

다음 코드 예시에서는 `register-cross-account-access-role` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

교차 계정 액세스 역할을 등록하는 방법

다음 `register-cross-account-access-role` 명령은 preview-agents 명령을 호출할 때 평가 실행이 시작될 때 Amazon Inspector가 EC2 인스턴스를 나열하는 데 사용하는 `arn:aws:iam::123456789012:role/inspector`의 ARN으로 IAM 역할을 등록합니다.

```
aws inspector register-cross-account-access-role --role-arn arn:aws:iam::123456789012:role/inspector
```

자세한 내용은 Amazon Inspector 사용 설명서의 Setting up Amazon Inspector를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RegisterCrossAccountAccessRole 섹션을 참조하세요.

remove-attributes-from-findings

다음 코드 예시에서는 `remove-attributes-from-findings` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

조사 결과에서 속성을 제거하는 방법

다음 remove-attributes-from-finding 명령은 ARN이 arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-8l1VIE0D/run/0-Z02cjjug/finding/0-T8yM9mEU인 조사 결과에서 키가 Example, 값이 example인 속성을 제거합니다.

```
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 사용 설명서의 Amazon Inspector Findings를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RemoveAttributesFromFindings 섹션을 참조하세요.

set-tags-for-resource

다음 코드 예시에서는 set-tags-for-resource 코드를 사용하는 방법을 보여줍니다.

AWS CLI

리소스의 태그 설정

다음 set-tags-for-resource 명령은 키가 Example, 값이 example인 태그를 ARN이 arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-7sbz2Kz0인 평가 템플릿으로 설정합니다.

```
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 설명서의 Amazon Inspector Assessment Templates and Assessment Runs를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [SetTagsForResource](https://docs.aws.amazon.com/goto/aws-cli/inspector-2016-02-16/SetTagsForResource) 섹션을 참조하세요.

start-assessment-run

다음 코드 예시에서는 start-assessment-run 코드를 사용하는 방법을 보여줍니다.

AWS CLI

평가 실행을 시작하는 방법

다음 start-assessment-run 명령은 ARN이 arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-it5r2S4T인 평가 템플릿을 사용하여 examplerun이라는 평가 실행을 시작합니다.

```
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 설명서의 Amazon Inspector Assessment Templates and Assessment Runs를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [StartAssessmentRun](https://docs.aws.amazon.com/goto/aws-cli/inspector-2016-02-16/StartAssessmentRun) 섹션을 참조하세요.

stop-assessment-run

다음 코드 예시에서는 stop-assessment-run 코드를 사용하는 방법을 보여줍니다.

AWS CLI

평가 실행을 중지하는 방법

다음 stop-assessment-run 명령은 arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-it5r2S4T/run/0-jOoroxyY의 ARN으로 평가 실행을 중지합니다.

```
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 설명서의 Amazon Inspector Assessment Templates and Assessment Runs를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StopAssessmentRun 섹션을 참조하세요.

`subscribe-to-event`

다음 코드 예시에서는 `subscribe-to-event` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

이벤트를 구독하는 방법

다음 예시에서는 ARN이 `arn:aws:sns:us-west-2:123456789012:exampletopic`인 주제에 `ASSESSMENT_RUN_COMPLETED` 이벤트에 대한 Amazon SNS 알림을 전송하는 프로세스를 구현합니다.

```
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 설명서의 Amazon Inspector Assessment Templates and Assessment Runs를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 SubscribeToEvent 섹션을 참조하세요.

`unsubscribe-from-event`

다음 코드 예시에서는 `unsubscribe-from-event` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

이벤트 구독을 취소하는 방법

다음 unsubscribe-from-event 명령은 arn:aws:sns:us-west-2:123456789012:exampletopic의 ARN을 사용하여 ASSESSMENT_RUN_COMPLETED 이벤트에 대한 Amazon SNS 알림을 주제로 보내는 프로세스를 비활성화합니다.

```
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 설명서의 Amazon Inspector Assessment Templates and Assessment Runs를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UnsubscribeFromEvent 섹션을 참조하세요.

update-assessment-target

다음 코드 예시에서는 update-assessment-target 코드를 사용하는 방법을 보여줍니다.

AWS CLI

평가 대상 업데이트

다음 update-assessment-target 명령은 평가 대상을 arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX의 ARN과 Example의 이름으로, 리소스 그룹을 arn:aws:inspector:us-west-2:123456789012:resourcegroup/0-yNbgL5Pt의 ARN으로 업데이트합니다.

```
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 사용 설명서의 Amazon Inspector Assessment Targets를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateAssessmentTarget 섹션을 참조하세요.

update-filter

다음 코드 예시에서는 update-filter 코드를 사용하는 방법을 보여줍니다.

AWS CLI

필터 업데이트

다음 update-filter 예시에서는 필터를 업데이트하여 ECR 인스턴스 조사 결과 대신 Lambda 조사 결과를 생략합니다.

```
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 Managing findings in Amazon Inspector를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateFilter 섹션을 참조하세요.

AWS CLI를 사용한 AWS IoT 예시

다음 코드 예시에서는 AWS IoT에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

accept-certificate-transfer

다음 코드 예시에서는 `accept-certificate-transfer`의 사용 방법을 보여줍니다.

AWS CLI

다른 AWS 계정에서 전송된 디바이스 인증서 수락

다음 `accept-certificate-transfer` 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 AcceptCertificateTransfer를 참조하세요.

add-thing-to-billing-group

다음 코드 예시에서는 `add-thing-to-billing-group`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 결제 그룹에 이름으로 사물 추가

다음 add-thing-to-billing-group 예시에서는 GroupOne이라는 결제 그룹에 MyLightBulb라는 사물을 추가합니다.

```
aws iot add-thing-to-billing-group \
    --billing-group-name GroupOne \
    --thing-name MyLightBulb
```

이 명령은 출력을 생성하지 않습니다.

예시 2: 결제 그룹에 ARN으로 사물 추가

다음 add-thing-to-billing-group 예시에서는 지정된 ARN이 있는 결제 그룹에 지정된 ARN이 있는 사물을 추가합니다. 여러 AWS 리전 또는 계정을 사용하는 경우 ARN을 지정하는 것이 도움이 됩니다. 이렇게 하면 올바른 리전 및 계정에 확실히 추가할 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 AWS IoT 개발자 안내서의 결제 그룹을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AddThingToBillingGroup을 참조하세요.

add-thing-to-thing-group

다음 코드 예시에서는 add-thing-to-thing-group의 사용 방법을 보여줍니다.

AWS CLI

그룹에 사물 추가

다음 add-thing-to-thing-group 예시에서는 지정된 사물 그룹에 지정된 사물을 추가합니다.

```
aws iot add-thing-to-thing-group \
    --thing-name MyLightBulb \
```

```
    --thing-group-name LightBulbs
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 사물 그룹을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AddThingToThingGroup을 참조하세요.

associate-targets-with-job

다음 코드 예시에서는 associate-targets-with-job의 사용 방법을 보여줍니다.

AWS CLI

사물 그룹을 연속 작업에 연결

다음 associate-targets-with-job 예시에서는 지정된 사물 그룹을 지정된 연속 작업에 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.

- API 세부 정보는 AWS CLI 명령 참조의 AssociateTargetsWithJob을 참조하세요.

attach-policy

다음 코드 예시에서는 attach-policy의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 정책을 사물 그룹에 연결

다음 attach-policy 예시에서는 지정된 정책을 ARN으로 식별된 사물 그룹에 연결합니다.

```
aws iot attach-policy \
    --target "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs" \
    --policy-name "UpdateDeviceCertPolicy"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 사물 그룹을 참조하세요.

예시 2: 정책을 인증서에 연결

다음 attach-policy 예시에서는 UpdateDeviceCertPolicy 정책을 인증서로 지정된 위탁자에 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 AWS IoT 정책 연결을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AttachPolicy를 참조하세요.

attach-security-profile

다음 코드 예시에서는 attach-security-profile의 사용 방법을 보여줍니다.

AWS CLI

모든 미등록 디바이스에 보안 프로필 연결

다음 attach-security-profile 예시에서는 Testprofile이라는 AWS IoT Device Defender 보안 프로필을 이 AWS 계정의 us-west-2 리전에 있는 모든 미등록 디바이스에 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 명령 탐지를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AttachSecurityProfile을 참조하세요.

attach-thing-principal

다음 코드 예시에서는 attach-thing-principal의 사용 방법을 보여줍니다.

AWS CLI

사물에 인증서 연결

다음 attach-thing-principal 예시에서는 MyTemperatureSensor 사물에 인증서를 연결합니다. 인증서는 ARN으로 식별됩니다. AWS IoT 콘솔에서 인증서의 ARN을 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.

- API 세부 정보는 AWS CLI 명령 참조의 AttachThingPrincipal을 참조하세요.

cancel-audit-mitigation-actions-task

다음 코드 예시에서는 cancel-audit-mitigation-actions-task의 사용 방법을 보여줍니다.

AWS CLI

감사 완화 조치 작업 취소

다음 cancel-audit-mitigations-action-task 예시에서는 지정된 작업에 대한 완화 조치의 적용을 취소합니다. 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 조치 명령)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CancelAuditMitigationActionsTask를 참조하세요.

cancel-audit-task

다음 코드 예시에서는 `cancel-audit-task`의 사용 방법을 보여줍니다.

AWS CLI

감사 작업 취소

다음 `cancel-audit-task` 예시에서는 지정된 작업 ID를 사용해 감사 작업을 취소합니다. 완료된 작업은 취소할 수 없습니다.

```
aws iot cancel-audit-task \
    --task-id a3aea009955e501a31b764abe1bebd3d
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 명령 감사를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CancelAuditTask를 참조하세요.

cancel-certificate-transfer

다음 코드 예시에서는 `cancel-certificate-transfer`의 사용 방법을 보여줍니다.

AWS CLI

다른 AWS 계정으로 인증서 전송 취소

다음 `cancel-certificate-transfer` 예시에서는 지정된 인증서의 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.

- API 세부 정보는 AWS CLI 명령 참조의 CancelCertificateTransfer를 참조하세요.

cancel-job-execution

다음 코드 예시에서는 `cancel-job-execution`의 사용 방법을 보여줍니다.

AWS CLI

디바이스의 작업 실행 취소

다음 `cancel-job-execution` 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 CancelJobExecution을 참조하세요.

cancel-job

다음 코드 예시에서는 `cancel-job`의 사용 방법을 보여줍니다.

AWS CLI

작업 취소

다음 `cancel-job` 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 CancelJob을 참조하세요.

`clear-default-authorizer`

다음 코드 예시에서는 `clear-default-authorizer`의 사용 방법을 보여줍니다.

AWS CLI

기본 권한 부여자 삭제

다음 `clear-default-authorizer` 예시에서는 현재 구성된 기본 사용자 지정 권한 부여자를 삭제합니다. 이 명령을 실행하면 기본 권한 부여자가 존재하지 않게 됩니다. 사용자 지정 권한 부여자를 사용하는 경우, HTTP 요청 헤더에서 이름으로 지정해야 합니다.

```
aws iot clear-default-authorizer
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT API 참조의 ClearDefaultAuthorizer를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ClearDefaultAuthorizer를 참조하세요.

`confirm-topic-rule-destination`

다음 코드 예시에서는 `confirm-topic-rule-destination`의 사용 방법을 보여줍니다.

AWS CLI

주제 규칙 대상 확인

다음 `confirm-topic-rule-destination` 예시에서는 HTTP 엔드포인트에서 수신된 확인 토큰을 사용하여 주제 규칙 대상을 확인합니다.

```
aws iot confirm-topic-rule-destination \
    --confirmation-token "AYADeIcmtq-
ZkxfpiWIQqHWM5ucAXwABABVhd3MtY3J5cHRvLXB1YmxpYy1rZXkAREFxY1E0UmlGeDg0V21BZWZ1VjZtZWFRVUJJUkt
aywpPqg8YEsa1lD4B40aJ2s1wEHKMybiF1RoOZzYisI0IvslzQY5UmCkqq3tV-3f7-
nKfosgIAAAAADAAAEAAAAAAAAAAAAAAAAAAi9RMgy-
V19V9m6Iw2xfbw_____wAAAAEAAAAAAAAAAAAAAAEAAAB1hw4SokgUcxiJ3gTO6n50NLJVpzyQR1UmPIj5sShqXEQGcO
iufgrzTePl8RZYOWr0O6Aj9DiVzJZx-1iD6Pu-
G6PUw1kaO7Knzs2B4AD0qfrHUF4pYRTvyUgBnMGUCMQC8ZRmhKqntd_c6Kgrow3bMUDBvNqo2qZr8Z8Jm2rzgseROlAn
PIetJ803Z4IlIlF8xXlcdPGP-PV1dOXFemyL8g"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 [주제 규칙 대상 확인](#)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ConfirmTopicRuleDestination](#)을 참조하세요.

`create-audit-suppression`

다음 코드 예시에서는 `create-audit-suppression`의 사용 방법을 보여줍니다.

AWS CLI

감사 결과 억제 생성

다음 `create-audit-suppression` 예시에서는 과도한 허용을 알리는 플래그가 지정된 'virtualMachinePolicy'라는 정책에 대한 감사 결과 억제를 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 AWS IoT 개발자 안내서의 [감사 결과 억제](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [CreateAuditSuppression](#)을 참조하세요.

`create-authorizer`

다음 코드 예시에서는 `create-authorizer`의 사용 방법을 보여줍니다.

AWS CLI

사용자 지정 권한 부여자 생성

다음 `create-authorizer` 예시에서는 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 AWS IoT API 참조의 CreateAuthorizer를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateAuthorizer를 참조하세요.

`create-billing-group`

다음 코드 예시에서는 `create-billing-group`의 사용 방법을 보여줍니다.

AWS CLI

결제 그룹 생성

다음 `create-billing-group` 예시에서는 `GroupOne`이라는 간단한 결제 그룹을 생성합니다.

```
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 AWS IoT 개발자 안내서의 [결제 그룹]을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [CreateBillingGroup]을 참조합니다.

`create-certificate-from-csr`

다음 코드 예시에서는 `create-certificate-from-csr`의 사용 방법을 보여줍니다.

AWS CLI

인증서 서명 요청(CSR)에서 디바이스 인증서 생성

다음 `create-certificate-from-csr` 예시에서는 CSR에서 디바이스 인증서를 생성합니다. CSR을 생성하려면 `openssl` 명령을 사용합니다.

```
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 AWS IoT API 참조의 CreateCertificateFromCSR을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateCertificateFromCsr을 참조하세요.

`create-custom-metric`

다음 코드 예시에서는 `create-custom-metric`의 사용 방법을 보여줍니다.

AWS CLI

디바이스가 Device Defender에 게시한 사용자 지정 지표 생성

다음 `create-custom-metric` 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateCustomMetric을 참조하세요.

`create-dimension`

다음 코드 예시에서는 `create-dimension`의 사용 방법을 보여줍니다.

AWS CLI

측정기준 생성

다음 create-dimension은 TopicFilterForAuthMessages라는 단일 주제 필터를 사용하여 측정기준을 생성합니다.

```
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 명령 탐지를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateDimension을 참조하세요.

create-domain-configuration

다음 코드 예시에서는 create-domain-configuration의 사용 방법을 보여줍니다.

AWS CLI

도메인 구성 생성

다음 create-domain-configuration 예시에서는 서비스 유형이 DATA인 AWS 관리형 도메인 구성을 생성합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateDomainConfiguration을 참조하세요.

`create-dynamic-thing-group`

다음 코드 예시에서는 `create-dynamic-thing-group`의 사용 방법을 보여줍니다.

AWS CLI

동적 사물 그룹 생성

다음 `create-dynamic-thing-group` 예시에서는 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 AWS IoT 개발자 안내서의 동적 사물 그룹을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateDynamicThingGroup을 참조하세요.

`create-job`

다음 코드 예시에서는 `create-job`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 작업 생성

다음 `create-job` 예시에서는 JSON 문서를 `MyRaspberryPi` 디바이스로 전송하는 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"
}
```

예시 2: 연속 작업 생성

다음 `create-job` 예시에서는 대상으로 지정된 사물이 작업을 완료한 후 계속 실행되는 작업을 생성합니다. 이 예시에서 대상은 사물 그룹이므로 새 디바이스가 그룹에 추가되면 연속 작업이 그러한 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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateJob을 참조하세요.

`create-keys-and-certificate`

다음 코드 예시에서는 `create-keys-and-certificate`의 사용 방법을 보여줍니다.

AWS CLI

RSA 키 쌍 생성 및 X.509 인증서 발급

다음 `create-keys-and-certificate`는 2048비트 RSA 키 쌍을 생성한 후 발급된 퍼블릭 키를 사용해 X.509 인증서를 발급합니다. 이때에만 AWS IoT에서 이 인증서의 프라이빗 키가 제공되므로, 이 키를 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 AWS IoT 디바이스 인증서 생성 및 등록을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateKeysAndCertificate를 참조하세요.

`create-mitigation-action`

다음 코드 예시에서는 `create-mitigation-action`의 사용 방법을 보여줍니다.

AWS CLI

완화 조치 생성

다음 `create-mitigation-action` 예시에서는 적용 시 `QuarantineGroup1`이라는 사물 그룹으로 사물을 이동하는 `AddThingsToQuarantineGroup1Action`이라는 완화 조치를 정의합니다. 이 작업은 동적 사물 그룹을 재정의합니다.

```
aws iot create-mitigation-action --cli-input-json file::params.json
```

`params.json`의 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 조치 명령)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateMitigationAction을 참조하세요.

`create-ota-update`

다음 코드 예시에서는 `create-ota-update`의 사용 방법을 보여줍니다.

AWS CLI

Amazon FreeRTOS와 함께 사용할 OTA 업데이트 생성

다음 `create-ota-update` 예시에서는 사물의 대상 그룹 1개 또는 여러 그룹에서 AWS IoT OTAUpdate를 생성합니다. 이는 단일 디바이스 또는 디바이스 그룹에 새 펌웨어 이미지를 배포할 수 있는 Amazon FreeRTOS 무선 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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateOtaUpdate를 참조하세요.

create-policy-version

다음 코드 예시에서는 `create-policy-version`의 사용 방법을 보여줍니다.

AWS CLI

새 버전으로 정책 업데이트

다음 `create-policy-version` 예시에서는 정책 정의를 업데이트하여 새 정책 버전을 생성합니다. 또한 이 예시에서는 새 버전을 기본값으로 설정합니다.

```
aws iot create-policy-version \
    --policy-name UpdateDeviceCertPolicy \
    --policy-document file://policy.json \
    --set-as-default
```

`policy.json`의 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 AWS IoT 정책을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreatePolicyVersion을 참조하세요.

`create-policy`

다음 코드 예시에서는 `create-policy`의 사용 방법을 보여줍니다.

AWS CLI

AWS IoT 정책 생성

다음 `create-policy` 예시에서는 TemperatureSensorPolicy라는 AWS IoT 정책을 생성합니다. `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 AWS IoT 정책을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreatePolicy를 참조하세요.

`create-provisioning-claim`

다음 코드 예시에서는 `create-provisioning-claim`의 사용 방법을 보여줍니다.

AWS CLI

프로비저닝 클레임 생성

다음 `create-provisioning-claim` 예시에서는 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 AWS IoT Core 개발자 안내서의 신뢰할 수 있는 사용자에 의한 프로비저닝을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateProvisioningClaim을 참조하세요.

`create-provisioning-template-version`

다음 코드 예시에서는 `create-provisioning-template-version`의 사용 방법을 보여줍니다.

AWS CLI

프로비저닝 템플릿 버전 생성

다음 예시에서는 지정된 프로비저닝 템플릿의 버전을 생성합니다. 새 버전의 본문은 파일 `template.json`에 제공됩니다.

```
aws iot create-provisioning-template-version \
    --template-name widget-template \
    --template-body file://template.json
```

template.json의 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 AWS IoT 보안 터널링을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateProvisioningTemplateVersion을 참조하세요.

create-provisioning-template

다음 코드 예시에서는 create-provisioning-template의 사용 방법을 보여줍니다.

AWS CLI

프로비저닝 템플릿 생성

다음 create-provisioning-template 예시에서는 파일 template.json에 정의된 프로비저닝 템플릿을 생성합니다.

```
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.json의 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 AWS IoT 보안 터널링을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateProvisioningTemplate을 참조하세요.

create-role-alias

다음 코드 예시에서는 create-role-alias의 사용 방법을 보여줍니다.

AWS CLI

역할 별칭 생성

다음 create-role-alias 예시에서는 지정된 역할에 대해 LightBulbRole이라는 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 AWS IoT API 참조의 CreateRoleAlias를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateRoleAlias를 참조하세요.

create-scheduled-audit

다음 코드 예시에서는 create-scheduled-audit의 사용 방법을 보여줍니다.

AWS CLI

예약된 감사 생성

다음 create-scheduled-audit 예시에서는 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 명령 감사를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateScheduledAudit을 참조하세요.

create-security-profile

다음 코드 예시에서는 create-security-profile의 사용 방법을 보여줍니다.

AWS CLI

보안 프로필 생성

다음 create-security-profile 예시에서는 셀룰러 대역폭이 임계값을 초과하는지 또는 5분 내에 10회를 초과하는 권한 부여 실패가 발생하는지 확인하는 보안 프로필을 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 명령 탐지를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateSecurityProfile을 참조하세요.

`create-stream`

다음 코드 예시에서는 `create-stream`의 사용 방법을 보여줍니다.

AWS CLI

1개 이상의 대용량 파일을 MQTT를 통해 청크 단위로 전송하는 스트림 생성

다음 `create-stream` 예시에서는 1개 이상의 대용량 파일을 MQTT를 통해 청크 단위로 전송하는 스트림을 생성합니다. 스트림은 S3 같은 소스에서 데이터 바이트를 MQTT 메시지로서 청크 또는 블록 단위로 묶어서 전송합니다. 스트림 1개에 다수의 파일을 연결할 수 있습니다.

```
aws iot create-stream \
    --cli-input-json file://create-stream.json
```

`create-stream.json`의 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 AWS IoT API 참조의 CreateStream을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateStream을 참조하세요.

create-thing-group

다음 코드 예시에서는 create-thing-group의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 사물 그룹 생성

다음 create-thing-group 예시에서는 1개의 설명과 2개의 속성이 있는 LightBulbs라는 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"
```

```
}
```

예시 2: 상위 그룹에 포함된 사물 그룹 생성

다음 `create-thing-group`은 `LightBulbs`라는 상위 사물 그룹이 있는 `HalogenBulbs`라는 사물 그룹을 생성합니다.

```
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 AWS IoT 개발자 안내서의 [사물 그룹]을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [CreateThingGroup]을 참조하세요.

`create-thing-type`

다음 코드 예시에서는 `create-thing-type`의 사용 방법을 보여줍니다.

AWS CLI

사물 유형 정의

다음 `create-thing-type` 예시에서는 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 AWS IoT 개발자 안내서의 [사물 유형]을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [CreateThingType]을 참조하세요.

create-thing

다음 코드 예시에서는 `create-thing`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 레지스트리에서 사물 레코드 생성

다음 `create-thing` 예시에서는 AWS IoT 사물 레지스트리에서 디바이스의 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 "
}
```

예시 2: 사물 유형과 연결된 사물 정의

다음 `create-thing` 예시에서는 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 AWS IoT 개발자 안내서의 레지스트리를 사용하여 사물을 관리하는 방법 및 사물 유형을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateThing을 참조하세요.

create-topic-rule-destination

다음 코드 예시에서는 `create-topic-rule-destination`의 사용 방법을 보여줍니다.

AWS CLI

주제 규칙 대상 생성

다음 `create-topic-rule-destination` 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateTopicRuleDestination을 참조하세요.

create-topic-rule

다음 코드 예시에서는 create-topic-rule의 사용 방법을 보여줍니다.

AWS CLI

Amazon SNS 알림을 보내는 규칙 생성

다음 create-topic-rule 예시에서는 디바이스 섀도우에서 찾을 수 있는 토양 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 예시에서는 다음 JSON 코드를 plant-rule.json이라는 파일에 저장해야 합니다.

```
{
    "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 AWS IoT 규칙 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateTopicRule을 참조하세요.

`delete-account-audit-configuration`

다음 코드 예시에서는 `delete-account-audit-configuration`의 사용 방법을 보여줍니다.

AWS CLI

AWS 계정에 대한 모든 감사 확인 비활성화

다음 `delete-account-audit-configuration` 예시에서는 이 계정의 AWS IoT Device Defender 기본 설정을 복원하여 모든 감사 확인을 비활성화하고 구성 데이터를 삭제합니다. 또한 이 계정에서 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 명령 감사를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteAccountAuditConfiguration을 참조하세요.

`delete-audit-suppression`

다음 코드 예시에서는 `delete-audit-suppression`의 사용 방법을 보여줍니다.

AWS CLI

감사 결과 억제 삭제

다음 `delete-audit-suppression` 예시에서는 DEVICE_CERTIFICATE_EXPIRING_CHECK에 대한 감사 결과 억제를 삭제합니다.

```
aws iot delete-audit-suppression \
    --check-name DEVICE_CERTIFICATE_EXPIRING_CHECK \
    --resource-identifier deviceCertificateId="c7691e<shortened>"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 감사 결과 억제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteAuditSuppression을 참조하세요.

delete-authorizer

다음 코드 예시에서는 delete-authorizer의 사용 방법을 보여줍니다.

AWS CLI

사용자 지정 권한 부여자 삭제

다음 delete-authorizer 예시에서는 CustomAuthorizer라는 권한 부여자를 삭제합니다. 사용자 지정 권한 부여자는 삭제하기 전에 INACTIVE 상태에 있어야 합니다.

```
aws iot delete-authorizer \
    --authorizer-name CustomAuthorizer
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 DeleteAuthorizer를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteAuthorizer를 참조하세요.

delete-billing-group

다음 코드 예시에서는 delete-billing-group의 사용 방법을 보여줍니다.

AWS CLI

결제 그룹 삭제

다음 delete-billing-group 예시에서는 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 AWS IoT 개발자 안내서의 결제 그룹을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteBillingGroup을 참조하세요.

delete-ca-certificate

다음 코드 예시에서는 delete-ca-certificate의 사용 방법을 보여줍니다.

AWS CLI

CA 인증서 삭제

다음 `delete-ca-certificate` 예시에서는 지정된 인증서 ID로 CA 인증서를 삭제합니다.

```
aws iot delete-ca-certificate \
    --certificate-
id f4efed62c0142f16af278166f61962501165c4f0536295207426460058cd1467
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT API 참조의 DeleteCACertificate를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteCaCertificate를 참조하세요.

delete-certificate

다음 코드 예시에서는 `delete-certificate`의 사용 방법을 보여줍니다.

AWS CLI

디바이스 인증서 삭제

다음 `delete-certificate` 에시에서는 지정된 ID로 디바이스 인증서를 삭제합니다.

```
aws iot delete-certificate \
    --certificate-
id c0c57bbc8baaf4631a9a0345c957657f5e710473e3ddbee1428d216d54d53ac9
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT API 참조의 DeleteCertificate를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteCertificate를 참조하세요.

delete-custom-metric

다음 코드 예시에서는 `delete-custom-metric`의 사용 방법을 보여줍니다.

AWS CLI

사용자 지정 지표 삭제

다음 delete-custom-metric 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteCustomMetric을 참조하세요.

delete-dimension

다음 코드 예시에서는 delete-dimension의 사용 방법을 보여줍니다.

AWS CLI

측정기준 삭제

다음 delete-dimension 예시에서는 TopicFilterForAuthMessages라는 측정기준을 삭제합니다.

```
aws iot delete-dimension \
    --name TopicFilterForAuthMessages
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 명령 탐지를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteDimension을 참조하세요.

delete-domain-configuration

다음 코드 예시에서는 delete-domain-configuration의 사용 방법을 보여줍니다.

AWS CLI

도메인 구성 삭제

다음 `delete-domain-configuration` 예시에서는 AWS 계정에서 `additionalDataDomain`이라는 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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteDomainConfiguration을 참조하세요.

delete-dynamic-thing-group

다음 코드 예시에서는 `delete-dynamic-thing-group`의 사용 방법을 보여줍니다.

AWS CLI

동적 사물 그룹 삭제

다음 `delete-dynamic-thing-group` 예시에서는 지정된 동적 사물 그룹을 삭제합니다.

```
aws iot delete-dynamic-thing-group \
    --thing-group-name "RoomTooWarm"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 동적 사물 그룹을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteDynamicThingGroup을 참조하세요.

delete-job-execution

다음 코드 예시에서는 `delete-job-execution`의 사용 방법을 보여줍니다.

AWS CLI

작업 실행 삭제

다음 `delete-job-execution` 예시에서는 디바이스에서 지정된 작업의 실행을 삭제합니다. `describe-job-execution`을 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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteJobExecution을 참조하세요.

delete-job

다음 코드 예시에서는 delete-job의 사용 방법을 보여줍니다.

AWS CLI

작업 삭제

다음 delete-job 예시에서는 지정된 작업을 삭제합니다. --force 옵션을 지정하면 상태가 IN_PROGRESS인 경우에도 작업이 삭제됩니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteJob을 참조하세요.

delete-mitigation-action

다음 코드 예시에서는 delete-mitigation-action의 사용 방법을 보여줍니다.

AWS CLI

완화 조치 삭제

다음 delete-mitigation-action 예시에서는 지정된 완화 조치를 삭제합니다.

```
aws iot delete-mitigation-action \
```

```
    --action-name AddThingsToQuarantineGroup1Action
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 DeleteMitigationAction(완화 조치 명령)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteMitigationAction을 참조하세요.

delete-ota-update

다음 코드 예시에서는 delete-ota-update의 사용 방법을 보여줍니다.

AWS CLI

OTA 업데이트 삭제

다음 delete-ota-update 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteOtaUpdate를 참조하세요.

delete-policy-version

다음 코드 예시에서는 delete-policy-version의 사용 방법을 보여줍니다.

AWS CLI

정책 버전 삭제

다음 delete-policy-version 예시에서는 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 AWS IoT 정책을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeletePolicyVersion을 참조하세요.

delete-policy

다음 코드 예시에서는 delete-policy의 사용 방법을 보여줍니다.

AWS CLI

정책 삭제

다음 delete-policy 예시에서는 AWS 계정에서 지정된 정책을 삭제합니다.

```
aws iot delete-policy --policy-name UpdateDeviceCertPolicy
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 AWS IoT 정책을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeletePolicy를 참조하세요.

delete-provisioning-template-version

다음 코드 예시에서는 delete-provisioning-template-version의 사용 방법을 보여줍니다.

AWS CLI

프로비저닝 템플릿 버전 삭제

다음 delete-provisioning-template-version 예시에서는 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 AWS IoT 보안 터널링을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteProvisioningTemplateVersion을 참조하세요.

delete-provisioning-template

다음 코드 예시에서는 delete-provisioning-template의 사용 방법을 보여줍니다.

AWS CLI

프로비저닝 템플릿 삭제

다음 delete-provisioning-template 예시에서는 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 AWS IoT 보안 터널링을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteProvisioningTemplate을 참조하세요.

delete-registration-code

다음 코드 예시에서는 delete-registration-code의 사용 방법을 보여줍니다.

AWS CLI

등록 코드 삭제

다음 delete-registration-code 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteRegistrationCode를 참조하세요.

delete-role-alias

다음 코드 예시에서는 delete-role-alias의 사용 방법을 보여줍니다.

AWS CLI

AWS IoT 역할 별칭 삭제

다음 delete-role-alias 예시에서는 LightBulbRole이라는 AWS IoT 역할 별칭을 삭제합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteRoleAlias를 참조하세요.

delete-scheduled-audit

다음 코드 예시에서는 delete-scheduled-audit의 사용 방법을 보여줍니다.

AWS CLI

예정된 감사 삭제

다음 delete-scheduled-audit 예시에서는 AWSIoTDeviceDefenderDailyAudit이라는 AWS IoT Device Defender 예약 감사를 삭제합니다.

```
aws iot delete-scheduled-audit \
    --scheduled-audit-name AWSIoTDeviceDefenderDailyAudit
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 명령 감사를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteScheduledAudit을 참조하세요.

delete-security-profile

다음 코드 예시에서는 delete-security-profile의 사용 방법을 보여줍니다.

AWS CLI

보안 프로필 삭제

다음 delete-security-profile 예시에서는 PossibleIssue라는 보안 프로필을 삭제합니다.

```
aws iot delete-security-profile \
    --security-profile-name PossibleIssue
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 명령 탐지를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteSecurityProfile을 참조하세요.

delete-stream

다음 코드 예시에서는 delete-stream의 사용 방법을 보여줍니다.

AWS CLI

데이터 스트림 삭제

다음 delete-stream 예시에서는 지정된 스트림을 삭제합니다.

```
aws iot delete-stream \
    --stream-id stream12345
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT API 참조의 DeleteStream을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteStream을 참조하세요.

delete-thing-group

다음 코드 예시에서는 delete-thing-group의 사용 방법을 보여줍니다.

AWS CLI

사물 그룹 삭제

다음 delete-thing-group 예시에서는 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 AWS IoT 개발자 안내서의 사물 그룹을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteThingGroup을 참조하세요.

`delete-thing-type`

다음 코드 예시에서는 `delete-thing-type`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 사물 유형 삭제

다음 `delete-thing-type` 예시에서는 더 이상 사용되지 않는 사물 유형을 삭제합니다.

```
aws iot delete-thing-type \
    --thing-type-name "obsoleteThingType"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 사물 유형을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteThingType을 참조하세요.

`delete-thing`

다음 코드 예시에서는 `delete-thing`의 사용 방법을 보여줍니다.

AWS CLI

사물의 세부 정보 표시

다음 `delete-thing` 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteThing을 참조하세요.

delete-topic-rule-destination

다음 코드 예시에서는 delete-topic-rule-destination의 사용 방법을 보여줍니다.

AWS CLI

주제 규칙 대상 삭제

다음 delete-topic-rule-destination 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteTopicRuleDestination을 참조하세요.

delete-topic-rule

다음 코드 예시에서는 delete-topic-rule의 사용 방법을 보여줍니다.

AWS CLI

규칙 삭제

다음 delete-topic-rule 예시에서는 지정된 규칙을 삭제합니다.

```
aws iot delete-topic-rule \
    --rule-name "LowMoistureRule"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 규칙 삭제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteTopicRule을 참조하세요.

delete-v2-logging-level

다음 코드 예시에서는 delete-v2-logging-level의 사용 방법을 보여줍니다.

AWS CLI

사물 그룹의 로깅 수준 삭제

다음 delete-v2-logging-level 예시에서는 지정된 사물 그룹의 로깅 수준을 삭제합니다.

```
aws iot delete-v2-logging-level \
    --target-type THING_GROUP \
    --target-name LightBulbs
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteV2LoggingLevel을 참조하세요.

deprecate-thing-type

다음 코드 예시에서는 deprecate-thing-type의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 사물 유형 사용 중단

다음 deprecate-thing-type 예시에서는 사용자가 새 사물을 연결할 수 없도록 사물 유형을 사용 중지합니다.

```
aws iot deprecate-thing-type \
    --thing-type-name "obsoleteThingType"
```

이 명령은 출력을 생성하지 않습니다.

예시 2: 사물 유형의 사용 중지 되돌리기

다음 deprecate-thing-type 예시에서는 사물 유형의 사용 중단을 되돌려 사용자가 새 사물을 다시 연결할 수 있도록 합니다.

```
aws iot deprecate-thing-type \
    --thing-type-name "obsoleteThingType" \
    --undo-deprecate
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 사물 유형을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DeprecateThingType](https://docs.aws.amazon.com/goto/aws-cli/iot-2015-05-28/DeprecateThingType)을 참조하세요.

describe-account-audit-configuration

다음 코드 예시에서는 describe-account-audit-configuration의 사용 방법을 보여줍니다.

AWS CLI

현재 감사 구성 설정 보기

다음 describe-account-audit-configuration 예시에서는 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 명령 감사를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeAccountAuditConfiguration을 참조하세요.

describe-audit-finding

다음 코드 예시에서는 describe-audit-finding의 사용 방법을 보여줍니다.

AWS CLI

감사 결과의 세부 정보 나열

다음 describe-audit-finding 예시에서는 지정된 AWS IoT Device Defender 감사 결과의 세부 정보를 나열합니다. 감사는 여러 결과를 생성할 수 있습니다. list-audit-findings 명령을 사용하여 감사 결과 목록을 가져와 findingId를 얻습니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeAuditFinding을 참조하세요.

describe-audit-mitigation-actions-task

다음 코드 예시에서는 describe-audit-mitigation-actions-task의 사용 방법을 보여줍니다.

AWS CLI

감사 완화 조치 작업의 세부 정보 표시

다음 describe-audit-mitigation-actions-task 예시에서는 ResetPolicyVersionAction이 결과에 적용된 지정된 작업의 세부 정보를 보여줍니다. 결과에는 작업이 시작 및 종료된 시간, 대상 조사 결과 수(및 결과), 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 조치 명령)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeAuditMitigationActionsTask를 참조하세요.

`describe-audit-suppression`

다음 코드 예시에서는 `describe-audit-suppression`의 사용 방법을 보여줍니다.

AWS CLI

감사 결과 억제의 세부 정보 가져오기

다음 `describe-audit-suppression` 예시에서는 감사 결과 억제의 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 AWS IoT 개발자 안내서의 감사 결과 억제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeAuditSuppression을 참조하세요.

describe-audit-task

다음 코드 예시에서는 describe-audit-task의 사용 방법을 보여줍니다.

AWS CLI

감사 인스턴스 정보 가져오기

다음 describe-audit-task 예시에서는 AWS IoT Device Defender 감사의 인스턴스 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 명령 감사를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeAuditTask를 참조하세요.

describe-authorizer

다음 코드 예시에서는 describe-authorizer의 사용 방법을 보여줍니다.

AWS CLI

사용자 지정 권한 부여자의 정보 가져오기

다음 describe-authorizer 예시에서는 지정된 사용자 지정 권한 부여자의 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 AWS IoT API 참조의 DescribeAuthorizer를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeAuthorizer를 참조하세요.

describe-billing-group

다음 코드 예시에서는 describe-billing-group의 사용 방법을 보여줍니다.

AWS CLI

결제 그룹 정보 가져오기

다음 describe-billing-group 예시에서는 지정된 결제 그룹의 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 AWS IoT 개발자 안내서의 결제 그룹을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeBillingGroup을 참조하세요.

describe-ca-certificate

다음 코드 예시에서는 describe-ca-certificate의 사용 방법을 보여줍니다.

AWS CLI

CA 인증서의 세부 정보 가져오기

다음 describe-ca-certificate 예시에서는 지정된 CA 인증서의 세부 정보를 표시합니다.

```
aws iot describe-ca-certificate \
    --certificate-
id f4efed62c0142f16af278166f61962501165c4f0536295207426460058cd1467
```

출력:


```
{
    "certificateDescription": {
        "certificateArn": "arn:aws:iot:us-west-2:123456789012:cacert/
f4efed62c0142f16af278166f61962501165c4f0536295207426460058cd1467",
        "certificateId":
 "f4efed62c0142f16af278166f61962501165c4f0536295207426460058cd1467",
        "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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeCaCertificate를 참조하세요.

describe-certificate

다음 코드 예시에서는 describe-certificate의 사용 방법을 보여줍니다.

AWS CLI

인증서 정보 가져오기

다음 describe-certificate 예시에서는 지정된 인증서의 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 AWS IoT API 참조의 DescribeCertificate를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeCertificate를 참조하세요.

describe-custom-metric

다음 코드 예시에서는 describe-custom-metric의 사용 방법을 보여줍니다.

AWS CLI

Device Defender 사용자 지정 지표의 정보 가져오기

다음 describe-custom-metric 예시에서는 myCustomMetric이라는 사용자 지정 지표의 정보를 가져옵니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeCustomMetric을 참조하세요.

describe-default-authorizer

다음 코드 예시에서는 describe-default-authorizer의 사용 방법을 보여줍니다.

AWS CLI

기본 사용자 지정 권한 부여자의 정보 가져오기

다음 describe-default-authorizer 예시에서는 기본 사용자 지정 권한 부여자의 세부 정보를 표시합니다.

```
aws iot describe-default-authorizer
```

출력:

```
{
    "authorizerName": "CustomAuthorizer",
    "authorizerArn": "arn:aws:iot:us-west-2:123456789012:authorizer/
CustomAuthorizer"
}
```

자세한 내용은 AWS IoT API 참조의 DescribeDefaultAuthorizer를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDefaultAuthorizer를 참조하세요.

describe-dimension

다음 코드 예시에서는 describe-dimension의 사용 방법을 보여줍니다.

AWS CLI

측정기준 정보 가져오기

다음 describe-dimension 예시에서는 TopicFilterForAuthMessages라는 측정기준의 정보를 가져옵니다.

```
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 명령 탐지를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDimension을 참조하세요.

describe-domain-configuration

다음 코드 예시에서는 describe-domain-configuration의 사용 방법을 보여줍니다.

AWS CLI

도메인 구성 설명

다음 describe-domain-configuration 예시에서는 지정된 도메인 구성의 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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDomainConfiguration을 참조하세요.

describe-endpoint

다음 코드 예시에서는 describe-endpoint의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 현재 AWS 엔드포인트 가져오기

다음 describe-endpoint 예시에서는 모든 명령이 적용되는 기본 AWS 엔드포인트를 가져옵니다.

```
aws iot describe-endpoint
```

출력:

```
{
    "endpointAddress": "abc123defghijk.iot.us-west-2.amazonaws.com"
}
```

자세한 내용은 AWS IoT 개발자 안내서의 DescribeEndpoint를 참조하세요.

예시 2: ATS 엔드포인트 가져오기

다음 describe-endpoint 예시에서는 Amazon Trust Services(ATS) 엔드포인트를 가져옵니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeEndpoint를 참조하세요.

describe-event-configurations

다음 코드 예시에서는 describe-event-configurations의 사용 방법을 보여줍니다.

AWS CLI

게시되는 이벤트 유형 표시

다음 describe-event-configurations 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeEventConfigurations를 참조하세요.

describe-index

다음 코드 예시에서는 describe-index의 사용 방법을 보여줍니다.

AWS CLI

사물 인덱스의 현재 상태 가져오기

다음 describe-index 예시에서는 사물 인덱스의 현재 상태를 가져옵니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeIndex를 참조하세요.

describe-job-execution

다음 코드 예시에서는 describe-job-execution의 사용 방법을 보여줍니다.

AWS CLI

디바이스에서 작업의 실행 세부 정보 가져오기

다음 describe-job-execution 예시에서는 지정된 작업의 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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeJobExecution을 참조하세요.

describe-job

다음 코드 예시에서는 describe-job의 사용 방법을 보여줍니다.

AWS CLI

작업의 세부 상태 가져오기

다음 describe-job 예시에서는 ID가 example-job-01인 작업의 세부 상태를 가져옵니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeJob을 참조하세요.

describe-mitigation-action

다음 코드 예시에서는 describe-mitigation-action의 사용 방법을 보여줍니다.

AWS CLI

정의된 완화 조치의 세부 정보 보기

다음 describe-mitigation-action 예시에서는 지정된 완화 조치의 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 조치 명령)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeMitigationAction을 참조하세요.

describe-provisioning-template-version

다음 코드 예시에서는 describe-provisioning-template-version의 사용 방법을 보여줍니다.

AWS CLI

프로비저닝 템플릿 버전 설명

다음 describe-provisioning-template-version 예시에서는 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 AWS IoT Core 개발자 안내서의 플릿 프로비저닝을 사용하여 디바이스 인증서가 없는 디바이스 프로비저닝을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeProvisioningTemplateVersion을 참조하세요.

describe-provisioning-template

다음 코드 예시에서는 describe-provisioning-template의 사용 방법을 보여줍니다.

AWS CLI

프로비저닝 템플릿 설명

다음 describe-provisioning-template 예시에서는 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 AWS IoT Core 개발자 안내서의 플릿 프로비저닝을 사용하여 디바이스 인증서가 없는 디바이스 프로비저닝을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeProvisioningTemplate을 참조하세요.

`describe-role-alias`

다음 코드 예시에서는 `describe-role-alias`의 사용 방법을 보여줍니다.

AWS CLI

AWS IoT 역할 별칭 정보 가져오기

다음 `describe-role-alias` 예시에서는 지정된 역할 별칭의 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 AWS IoT API 참조의 DescribeRoleAlias를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeRoleAlias를 참조하세요.

describe-scheduled-audit

다음 코드 예시에서는 describe-scheduled-audit의 사용 방법을 보여줍니다.

AWS CLI

예정된 감사 정보 가져오기

다음 describe-scheduled-audit 예시에서는 AWSIoTDeviceDefenderDailyAudit이라는 AWS IOT Device Defender 예약 감사의 세부 정보를 가져옵니다.

```
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 명령 감사를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeScheduledAudit을 참조하세요.

describe-security-profile

다음 코드 예시에서는 `describe-security-profile`의 사용 방법을 보여줍니다.

AWS CLI

보안 프로필 정보 가져오기

다음 `describe-security-profile` 예시에서는 `PossibleIssue.`라는 AWS IoT Device Defender 보안 프로필의 정보를 가져옵니다.

```
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 명령 탐지를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeSecurityProfile을 참조하세요.

describe-stream

다음 코드 예시에서는 `describe-stream`의 사용 방법을 보여줍니다.

AWS CLI

스트림 정보 가져오기

다음 `describe-stream` 예시에서는 지정된 스트림의 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 AWS IoT API 참조의 DescribeStream을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeStream을 참조하세요.

describe-thing-group

다음 코드 예시에서는 describe-thing-group의 사용 방법을 보여줍니다.

AWS CLI

사물 그룹 정보 가져오기

다음 describe-thing-group 예시에서는 HalogenBulbs라는 사물 그룹의 정보를 가져옵니다.

```
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 AWS IoT 개발자 안내서의 사물 그룹을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeThingGroup을 참조하세요.

describe-thing-type

다음 코드 예시에서는 describe-thing-type의 사용 방법을 보여줍니다.

AWS CLI

사물 유형 정보 가져오기

다음 describe-thing-type 예시에서는 AWS 계정에 정의된 지정된 사물 유형의 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 AWS IoT 개발자 안내서의 사물 유형을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeThingType을 참조하세요.

describe-thing

다음 코드 예시에서는 describe-thing의 사용 방법을 보여줍니다.

AWS CLI

사물의 세부 정보 표시

다음 describe-thing 예시에서는 AWS 계정의 AWS IoT 레지스트리에 정의된 사물(디바이스)의 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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeThing을 참조하세요.

detach-policy

다음 코드 예시에서는 detach-policy의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 사물 그룹에서 AWS IoT 정책 분리

다음 detach-policy 예시에서는 지정된 정책을 사물 그룹에서 분리하고, 여기에서 확장하여 해당 그룹의 모든 사물과 하위 그룹에서도 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 AWS IoT 개발자 안내서의 사물 그룹을 참조하세요.

예시 2: 디바이스 인증서에서 AWS IoT 정책 분리

다음 detach-policy 예시에서는 ARN으로 식별되는 디바이스 인증서에서 TemperatureSensorPolicy 정책을 분리합니다.

```
aws iot detach-policy \
    --policy-name TemperatureSensorPolicy \
    --target arn:aws:iot:us-
west-2:123456789012:cert/488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 DetachPolicy를 참조하세요.

detach-security-profile

다음 코드 예시에서는 detach-security-profile의 사용 방법을 보여줍니다.

AWS CLI

대상에서 보안 프로필 연결 해제

다음 detach-security-profile 예시에서는 Testprofile이라는 AWS IoT Device Defender 보안 프로필과 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 명령 탐지를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DetachSecurityProfile을 참조하세요.

detach-thing-principal

다음 코드 예시에서는 detach-thing-principal의 사용 방법을 보여줍니다.

AWS CLI

사물에서 인증서 및 위탁자 분리

다음 detach-thing-principal 예시에서는 지정된 사물에서 위탁자를 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.

- API 세부 정보는 AWS CLI 명령 참조의 DetachThingPrincipal을 참조하세요.

disable-topic-rule

다음 코드 예시에서는 disable-topic-rule의 사용 방법을 보여줍니다.

AWS CLI

주제 규칙 비활성화

다음 `disable-topic-rule` 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 DisableTopicRule을 참조하세요.

enable-topic-rule

다음 코드 예시에서는 `enable-topic-rule`의 사용 방법을 보여줍니다.

AWS CLI

주제 규칙 활성화

다음 `enable-topic-rule` 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 EnableTopicRule을 참조하세요.

get-behavior-model-training-summaries

다음 코드 예시에서는 `get-behavior-model-training-summaries`의 사용 방법을 보여줍니다.

AWS CLI

Device Defender의 ML Detect Security Profile 교육 모델 상태 나열

다음 get-behavior-model-training-summaries 예시에서는 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(명령 탐지)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetBehaviorModelTrainingSummaries를 참조하세요.

get-cardinality

다음 코드 예시에서는 `get-cardinality`의 사용 방법을 보여줍니다.

AWS CLI

쿼리와 일치하는 고유 값의 개수(근사치)를 반환

다음 설정 스크립트를 사용하여 온도 센서 10개를 나타내는 10개의 사물을 생성할 수 있습니다. 각 새 사물에는 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` 예시에서는 설정 스크립트에서 생성한 10개의 센서를 쿼리하고 온도 센서가 비정상 온도 값을 보고하는 랙의 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.

- API 세부 정보는 AWS CLI 명령 참조의 GetCardinality를 참조하세요.

get-effective-policies

다음 코드 예시에서는 `get-effective-policies`의 사용 방법을 보여줍니다.

AWS CLI

사물에 영향을 미치는 정책 나열

다음 `get-effective-policies` 예시에서는 지정된 사물에 영향을 미치는 정책을 나열합니다. 여기에는 해당 사물이 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 AWS IoT 개발자 안내서의 사물에 대한 효과적인 정책 가져오기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetEffectivePolicies를 참조하세요.

get-indexing-configuration

다음 코드 예시에서는 get-indexing-configuration의 사용 방법을 보여줍니다.

AWS CLI

사물 인덱싱 구성 가져오기

다음 get-indexing-configuration 예시에서는 AWS IoT 플릿 인덱싱의 현재 구성 데이터를 가져옵니다.

```
aws iot get-indexing-configuration
```

출력:

```
{
    "thingIndexingConfiguration": {
        "thingIndexingMode": "OFF",
        "thingConnectivityIndexingMode": "OFF"
    },
    "thingGroupIndexingConfiguration": {
        "thingGroupIndexingMode": "OFF"
    }
}
```

자세한 내용은 AWS IoT 개발자 안내서의 사물 인덱싱 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetIndexingConfiguration을 참조하세요.

get-job-document

다음 코드 예시에서는 get-job-document의 사용 방법을 보여줍니다.

AWS CLI

작업 관련 문서 가져오기

다음 get-job-document 예시에서는 ID가 example-job-01인 작업에 관한 문서의 세부 정보를 표시합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 GetJobDocument를 참조하세요.

get-logging-options

다음 코드 예시에서는 get-logging-options의 사용 방법을 보여줍니다.

AWS CLI

로깅 옵션 가져오기

다음 get-logging-options 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 GetLoggingOptions를 참조하세요.

get-ota-update

다음 코드 예시에서는 get-ota-update의 사용 방법을 보여줍니다.

AWS CLI

OTA 업데이트 정보 가져오기

다음 get-ota-update 예시에서는 지정된 OTA 업데이트의 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.

- API 세부 정보는 AWS CLI 명령 참조의 GetOtaUpdate를 참조하세요.

get-percentiles

다음 코드 예시에서는 get-percentiles의 사용 방법을 보여줍니다.

AWS CLI

쿼리와 일치하는 집계 값을 백분위수 그룹으로 그룹화

다음 설정 스크립트를 사용하여 온도 센서 10개를 나타내는 10개의 사물을 생성할 수 있습니다. 각 새 사물에는 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` 예시에서는 설정 스크립트에서 생성한 센서 10개를 쿼리하고 지정된 각 백분위수 그룹의 값을 반환합니다. 백분위수 그룹 '10'에는 쿼리와 일치하는 값의 약 10%에서 발생하는 집계된 필드 값이 포함됩니다. 다음 출력에서 {"percent": 10.0, "value": 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.

- API 세부 정보는 AWS CLI 명령 참조의 GetPercentiles를 참조하세요.

get-policy-version

다음 코드 예시에서는 `get-policy-version`의 사용 방법을 보여줍니다.

AWS CLI

정책의 특정 버전 정보 가져오기

다음 `get-policy-version` 예시에서는 지정된 정책의 첫 번째 버전 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 AWS IoT 정책을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetPolicyVersion을 참조하세요.

get-policy

다음 코드 예시에서는 get-policy의 사용 방법을 보여줍니다.

AWS CLI

정책의 기본 버전 정보 가져오기

다음 get-policy 예시에서는 지정된 정책의 기본 버전 정보를 가져옵니다.

```
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 AWS IoT 정책을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetPolicy를 참조하세요.

get-registration-code

다음 코드 예시에서는 get-registration-code의 사용 방법을 보여줍니다.

AWS CLI

AWS 계정별 등록 코드 가져오기

다음 get-registration-code 예시에서는 AWS 계정별 등록 코드를 가져옵니다.

```
aws iot get-registration-code
```

출력:

```
{
    "registrationCode":
 "15c51ae5e36ba59ba77042df1115862076bea4bd15841c838fcb68d5010a614c"
}
```

자세한 내용은 AWS IoT 개발자 안내서의 자체 인증서 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetRegistrationCode를 참조하세요.

get-statistics

다음 코드 예시에서는 get-statistics의 사용 방법을 보여줍니다.

AWS CLI

디바이스 인덱스에서 집계 데이터 검색

다음 get-statistics 예시에서는 디바이스 섀도우에서 false로 설정된 connectivity.connected라는 속성이 있는 사물 수(즉, 연결되지 않은 디바이스 수)를 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.

- API 세부 정보는 AWS CLI 명령 참조의 GetStatistics를 참조하세요.

get-topic-rule-destination

다음 코드 예시에서는 get-topic-rule-destination의 사용 방법을 보여줍니다.

AWS CLI

주제 규칙 대상 가져오기

다음 `get-topic-rule-destination` 예시에서는 주제 규칙 대상의 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.

- API 세부 정보는 AWS CLI 명령 참조의 GetTopicRuleDestination을 참조하세요.

get-topic-rule

다음 코드 예시에서는 `get-topic-rule`의 사용 방법을 보여줍니다.

AWS CLI

규칙 정보 가져오기

다음 `get-topic-rule` 예시에서는 지정된 규칙의 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 AWS IoT 개발자 안내서의 규칙 보기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetTopicRule을 참조하세요.

get-v2-logging-options

다음 코드 예시에서는 get-v2-logging-options의 사용 방법을 보여줍니다.

AWS CLI

현재 로깅 옵션 나열

다음 get-v2-logging-options 예시에서는 AWS IoT의 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 세부 정보는 AWS CLI 명령 참조의 GetV2LoggingOptions를 참조하세요.

list-active-violations

다음 코드 예시에서는 `list-active-violations`의 사용 방법을 보여줍니다.

AWS CLI

활성 위반 나열

다음 `list-active-violations` 예시에서는 지정된 보안 프로파일의 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
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListActiveViolations를 참조하세요.

`list-attached-policies`

다음 코드 예시에서는 `list-attached-policies`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 그룹에 연결되어 있는 정책 나열

다음 `list-attached-policies` 예시에서는 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 AWS IoT 개발자 안내서의 [사물 그룹]을 참조하세요.

예시 2: 디바이스 인증서에 연결된 정책 나열

다음 `list-attached-policies` 예시에서는 디바이스 인증서에 연결된 AWS IoT 정책을 나열합니다. 인증서는 ARN으로 식별됩니다.

```
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 AWS IoT 개발자 안내서의 [사물 그룹]을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListAttachedPolicies를 참조하세요.

list-audit-findings

다음 코드 예시에서는 list-audit-findings의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 감사의 모든 결과 나열

다음 list-audit-findings 예시에서는 지정된 작업 ID로 AWS IoT Device Defender 감사의 모든 결과를 나열합니다.

```
aws iot list-audit-findings \
    --task-id a3aea009955e501a31b764abe1bebd3d
```

출력:

```
{
    "findings": []
}
```

예시 2: 감사 확인 유형의 결과 나열

다음 list-audit-findings 예시에서는 디바이스가 디바이스 인증서를 공유하는 2019년 6월 5일부터 2019년 6월 19일까지 실행된 AWS IoT Device Defender 감사의 결과를 보여줍니다. 확인 이름을 지정할 때는 시작 및 종료 시간을 입력해야 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 명령 감사를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListAuditFindings를 참조하세요.

list-audit-mitigation-actions-executions

다음 코드 예시에서는 `list-audit-mitigation-actions-executions`의 사용 방법을 보여줍니다.

AWS CLI

감사 완화 조치 실행의 세부 정보 나열

감사 완화 조치 작업은 AWS IoT Device Defender 감사의 하나 이상의 결과에 완화 조치를 적용합니다. 다음 `list-audit-mitigation-actions-executions` 예시에서는 `taskId`가 지정된 완화 조치 작업 및 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 조치 명령)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListAuditMitigationActionsExecutions를 참조하세요.

`list-audit-mitigation-actions-tasks`

다음 코드 예시에서는 `list-audit-mitigation-actions-tasks`의 사용 방법을 보여줍니다.

AWS CLI

감사 완화 조치 작업 나열

다음 `list-audit-mitigation-actions-tasks` 예시에서는 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 조치 명령)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListAuditMitigationActionsTasks를 참조하세요.

`list-audit-suppressions`

다음 코드 예시에서는 `list-audit-suppressions`의 사용 방법을 보여줍니다.

AWS CLI

모든 감사 결과 억제 나열

다음 `list-audit-suppressions` 예시에서는 모든 활성화된 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 AWS IoT 개발자 안내서의 감사 결과 억제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListAuditSuppressions를 참조하세요.

list-audit-tasks

다음 코드 예시에서는 `list-audit-tasks`의 사용 방법을 보여줍니다.

AWS CLI

모든 감사 결과 나열

다음 `list-audit-tasks` 예시에서는 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 명령 감사를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListAuditTasks를 참조하세요.

list-authorizers

다음 코드 예시에서는 list-authorizers의 사용 방법을 보여줍니다.

AWS CLI

사용자 지정 권한 부여자 나열

다음 list-authorizers 예시에서는 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 AWS IoT API 참조의 ListAuthorizers를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListAuthorizers를 참조하세요.

list-billing-groups

다음 코드 예시에서는 list-billing-groups의 사용 방법을 보여줍니다.

AWS CLI

AWS 계정 및 리전의 결제 그룹 나열

다음 `list-billing-groups` 예시에서는 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 AWS IoT 개발자 안내서의 결제 그룹을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListBillingGroups를 참조하세요.

`list-ca-certificates`

다음 코드 예시에서는 `list-ca-certificates`의 사용 방법을 보여줍니다.

AWS CLI

AWS 계정에 등록된 CA 인증서 나열

다음 `list-ca-certificates` 예시에서는 AWS 계정에 등록된 CA 인증서를 나열합니다.

```
aws iot list-ca-certificates
```

출력:

```
{
    "certificates": [
```

```
        {
            "certificateArn": "arn:aws:iot:us-west-2:123456789012:cacert/
f4efed62c0142f16af278166f61962501165c4f0536295207426460058cd1467",
            "certificateId":
 "f4efed62c0142f16af278166f61962501165c4f0536295207426460058cd1467",
            "status": "INACTIVE",
            "creationDate": 1569365372.053
        }
    ]
}
```

자세한 내용은 AWS IoT 개발자 안내서의 자체 인증서 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListCaCertificates를 참조하세요.

list-certificates-by-ca

다음 코드 예시에서는 `list-certificates-by-ca`의 사용 방법을 보여줍니다.

AWS CLI

CA 인증서로 서명된 모든 디바이스 인증서 나열

다음 `list-certificates-by-ca` 예시에서는 지정된 CA 인증서로 서명된 AWS 계정의 모든 디바이스 인증서를 나열합니다.

```
aws iot list-certificates-by-ca \
    --ca-certificate-
id f4efed62c0142f16af278166f61962501165c4f0536295207426460058cd1467
```

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.

- API 세부 정보는 AWS CLI 명령 참조의 ListCertificatesByCa를 참조하세요.

list-certificates

다음 코드 예시에서는 `list-certificates`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: AWS 계정에 등록된 인증서 나열

다음 `list-certificates` 예시에서는 계정에 등록된 모든 인증서를 나열합니다. 페이징 수가 기본 제한인 25보다 큰 경우, 이 명령의 `nextMarker` 응답 값을 사용하여 다음 명령에 제공하여 다음 결과 배치를 가져올 수 있습니다. 값 없이 `nextMarker`가 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
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListCertificates를 참조하세요.

list-custom-metrics

다음 코드 예시에서는 `list-custom-metrics`의 사용 방법을 보여줍니다.

AWS CLI

사용자 지정 지표 나열

다음 `list-custom-metrics` 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListCustomMetrics를 참조하세요.

`list-dimensions`

다음 코드 예시에서는 `list-dimensions`의 사용 방법을 보여줍니다.

AWS CLI

AWS 계정의 측정기준 나열

다음 `list-dimensions` 예시에서는 AWS 계정에 정의된 모든 AWS IoT Device Defender 측정기준을 나열합니다.

```
aws iot list-dimensions
```

출력:

```
{
    "dimensionNames": [
        "TopicFilterForAuthMessages",
        "TopicFilterForActivityMessages"
    ]
}
```

자세한 내용은 AWS IoT 개발자 안내서의 명령 탐지를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListDimensions를 참조하세요.

`list-domain-configurations`

다음 코드 예시에서는 `list-domain-configurations`의 사용 방법을 보여줍니다.

AWS CLI

도메인 구성 나열

다음 `list-domain-configurations` 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListDomainConfigurations를 참조하세요.

list-indices

다음 코드 예시에서는 `list-indices`의 사용 방법을 보여줍니다.

AWS CLI

구성된 검색 인덱스 나열

다음 `list-indices` 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListIndices를 참조하세요.

list-job-executions-for-job

다음 코드 예시에서는 `list-job-executions-for-job`의 사용 방법을 보여줍니다.

AWS CLI

AWS 계정의 작업 나열

다음 `list-job-executions-for-job` 예시에서는 AWS 계정의 jobId로 지정된 작업의 모든 실행을 나열합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 [ListJobExecutionsForJob](#)을 참조하세요.

list-job-executions-for-thing

다음 코드 예시에서는 `list-job-executions-for-thing`의 사용 방법을 보여줍니다.

AWS CLI

사물에 대해 실행된 작업 나열

다음 `list-job-executions-for-thing` 예시에서는 `MyRaspberryPi`라는 이름의 사물에 대해 실행된 모든 작업을 나열합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 [ListJobExecutionsForThing]을 참조하세요.

`list-jobs`

다음 코드 예시에서는 `list-jobs`의 사용 방법을 보여줍니다.

AWS CLI

AWS 계정의 작업 나열

다음 `list-jobs` 예시에서는 작업 상태를 기준으로 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListJobs를 참조하세요.

list-mitigation-actions

다음 코드 예시에서는 `list-mitigation-actions`의 사용 방법을 보여줍니다.

AWS CLI

정의된 모든 완화 조치 나열

다음 `list-mitigation-actions` 예시에서는 AWS 계정 및 리전에 정의된 모든 완화 조치를 나열합니다. 각 조치의 이름, ARN 및 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 조치 명령)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListMitigationActions를 참조하세요.

list-mitigations-actions

다음 코드 예시에서는 `list-mitigations-actions`의 사용 방법을 보여줍니다.

AWS CLI

정의된 모든 완화 조치 나열

다음 `list-mitigations-actions` 예시에서는 AWS 계정 및 리전에 정의된 모든 완화 조치를 나열합니다. 각 조치의 이름, ARN 및 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 조치 명령)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListMitigationsActions를 참조하세요.

list-ota-updates

다음 코드 예시에서는 `list-ota-updates`의 사용 방법을 보여줍니다.

AWS CLI

계정의 OTA 업데이트 나열

다음 `list-ota-updates` 예시에서는 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 [ListOTAUpdates](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ListOtaUpdates](#)를 참조하세요.

`list-outgoing-certificates`

다음 코드 예시에서는 `list-outgoing-certificates`의 사용 방법을 보여줍니다.

AWS CLI

다른 AWS 계정으로 전송되는 인증서 나열

다음 `list-outgoing-certificates` 예시에서는 `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 AWS IoT API 참조의 ListOutgoingCertificates를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListOutgoingCertificates를 참조하세요.

`list-policies`

다음 코드 예시에서는 `list-policies`의 사용 방법을 보여줍니다.

AWS CLI

AWS 계정에 정의된 정책 나열

다음 `list-policies` 예시에서는 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 AWS IoT 정책을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListPolicies를 참조하세요.

list-policy-versions

다음 코드 예시에서는 list-policy-versions의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 정책의 모든 버전 보기

다음 list-policy-versions 예시에서는 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 AWS IoT 정책을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListPolicyVersions를 참조하세요.

list-principal-things

다음 코드 예시에서는 `list-principal-things`의 사용 방법을 보여줍니다.

AWS CLI

위탁자에 연결된 사물 나열

다음 `list-principal-things` 예시에서는 ARN으로 지정된 위탁자에 연결된 사물을 나열합니다.

```
aws iot list-principal-things \
    --principal arn:aws:iot:us-
west-2:123456789012:cert/2e1eb273792174ec2b9bf4e9b37e6c6c692345499506002a35159767055278e8
```

출력:

```
{
    "things": [
        "DeskLamp",
        "TableLamp"
    ]
}
```

자세한 내용은 AWS IoT API 참조의 ListPrincipalThings를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListPrincipalThings를 참조하세요.

list-provisioning-template-versions

다음 코드 예시에서는 `list-provisioning-template-versions`의 사용 방법을 보여줍니다.

AWS CLI

프로비저닝 템플릿 버전 나열

다음 `list-provisioning-template-versions` 예시에서는 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 AWS IoT 보안 터널링을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListProvisioningTemplateVersions를 참조하세요.

`list-provisioning-templates`

다음 코드 예시에서는 `list-provisioning-templates`의 사용 방법을 보여줍니다.

AWS CLI

프로비저닝 템플릿 나열

다음 `list-provisioning-templates` 예시에서는 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 AWS IoT 보안 터널링을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListProvisioningTemplates를 참조하세요.

list-role-aliases

다음 코드 예시에서는 `list-role-aliases`의 사용 방법을 보여줍니다.

AWS CLI

AWS 계정의 AWS IoT 역할 별칭 나열

다음 `list-role-aliases` 예시에서는 AWS 계정의 AWS IoT 역할 별칭을 나열합니다.

```
aws iot list-role-aliases
```

출력:

```
{
    "roleAliases": [
        "ResidentAlias",
        "ElectricianAlias"
    ]
}
```

자세한 내용은 AWS IoT API 참조의 ListRoleAliases를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListRoleAliases를 참조하세요.

list-scheduled-audits

다음 코드 예시에서는 `list-scheduled-audits`의 사용 방법을 보여줍니다.

AWS CLI

AWS 계정의 예약된 감사 나열

다음 `list-scheduled-audits` 예시에서는 AWS 계정의 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 명령 감사를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListScheduledAudits를 참조하세요.

`list-security-profiles-for-target`

다음 코드 예시에서는 `list-security-profiles-for-target`의 사용 방법을 보여줍니다.

AWS CLI

대상에 연결된 Device Defender 보안 프로필 나열

다음 `list-security-profiles-for-target` 예시에서는 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 명령 탐지를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListSecurityProfilesForTarget을 참조하세요.

`list-security-profiles`

다음 코드 예시에서는 `list-security-profiles`의 사용 방법을 보여줍니다.

AWS CLI

AWS 계정의 보안 프로필 나열

다음 `list-security-profiles` 예시에서는 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 명령 탐지를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListSecurityProfiles를 참조하세요.

list-streams

다음 코드 예시에서는 `list-streams`의 사용 방법을 보여줍니다.

AWS CLI

계정의 스트림 나열

다음 `list-streams` 예시에서는 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 AWS IoT API 참조의 ListStreams를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListStreams를 참조하세요.

`list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`의 사용 방법을 보여줍니다.

AWS CLI

리소스에 연결된 태그 및 값 표시

다음 `list-tags-for-resource` 예시에서는 사물 그룹 `LightBulbs`에 연결된 태그 및 값을 표시합니다.

```
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 AWS IoT 리소스 태그 지정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource를 참조하세요.

`list-targets-for-policy`

다음 코드 예시에서는 `list-targets-for-policy`의 사용 방법을 보여줍니다.

AWS CLI

AWS IoT 정책에 연결된 위탁자 나열

다음 `list-targets-for-policy` 예시에서는 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 AWS IoT 개발자 안내서의 사물 그룹을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTargetsForPolicy를 참조하세요.

`list-targets-for-security-profile`

다음 코드 예시에서는 `list-targets-for-security-profile`의 사용 방법을 보여줍니다.

AWS CLI

보안 프로필이 적용되는 대상 나열

다음 `list-targets-for-security-profile` 예시에서는 `PossibleIssue`라는 AWS IoT Device Defender 보안 프로필이 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 명령 탐지를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTargetsForSecurityProfile을 참조하세요.

list-thing-groups-for-thing

다음 코드 예시에서는 `list-thing-groups-for-thing`의 사용 방법을 보여줍니다.

AWS CLI

사물이 속한 그룹 나열

다음 `list-thing-groups-for-thing` 예시에서는 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 AWS IoT 개발자 안내서의 사물 그룹을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListThingGroupsForThing을 참조하세요.

list-thing-groups

다음 코드 예시에서는 `list-thing-groups`의 사용 방법을 보여줍니다.

AWS CLI

AWS 계정에 정의된 사물 그룹 나열

다음 describe-thing-group 예시에서는 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 AWS IoT 개발자 안내서의 사물 그룹을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListThingGroups를 참조하세요.

list-thing-principals

다음 코드 예시에서는 list-thing-principals의 사용 방법을 보여줍니다.

AWS CLI

사물에 연결된 위탁자 나열

다음 list-thing-principals 예시에서는 지정된 사물에 연결된 위탁자(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 AWS IoT API 참조의 ListThingPrincipals를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListThingPrincipals를 참조하세요.

`list-thing-types`

다음 코드 예시에서는 `list-thing-types`의 사용 방법을 보여줍니다.

AWS CLI

정의된 사물 유형 나열

다음 `list-thing-types` 예시에서는 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 AWS IoT 개발자 안내서의 사물 유형을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListThingTypes를 참조하세요.

list-things-in-billing-group

다음 코드 예시에서는 `list-things-in-billing-group`의 사용 방법을 보여줍니다.

AWS CLI

결제 그룹의 사물 나열

다음 `list-things-in-billing-group` 예시에서는 지정된 결제 그룹에 있는 사물을 나열합니다.

```
aws iot list-things-in-billing-group \
    --billing-group-name GroupOne
```

출력:

```
{
    "things": [
        "MyOtherLightBulb",
        "MyLightBulb"
    ]
}
```

자세한 내용은 AWS IoT 개발자 안내서의 결제 그룹을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListThingsInBillingGroup을 참조하세요.

list-things-in-thing-group

다음 코드 예시에서는 `list-things-in-thing-group`의 사용 방법을 보여줍니다.

AWS CLI

그룹에 속한 사물 나열

다음 `list-things-in-thing-group` 예시에서는 지정된 사물 그룹에 속하는 사물을 나열합니다.

```
aws iot list-things-in-thing-group \
    --thing-group-name LightBulbs
```

출력:

```
{
    "things": [
        "MyLightBulb"
    ]
}
```

자세한 내용은 AWS IoT 개발자 안내서의 사물 그룹을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListThingsInThingGroup을 참조하세요.

list-things

다음 코드 예시에서는 `list-things`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 레지스트리의 모든 사물 나열

다음 `list-things` 예시에서는 AWS 계정의 AWS IoT 레지스트리에 정의된 사물(디바이스)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
        }
    ]
}
```


예시 2: 특정 속성을 가진 정의된 사물 나열

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.

- API 세부 정보는 AWS CLI 명령 참조의 ListThings를 참조하세요.

list-topic-rule-destinations

다음 코드 예시에서는 `list-topic-rule-destinations`의 사용 방법을 보여줍니다.

AWS CLI

주제 규칙 대상 나열

다음 `list-topic-rule-destinations` 예시에서는 현재 AWS 리전에 정의된 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListTopicRuleDestinations를 참조하세요.

list-topic-rules

다음 코드 예시에서는 list-topic-rules의 사용 방법을 보여줍니다.

AWS CLI

규칙 나열

다음 list-topic-rules 예시에서는 정의된 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 AWS IoT 개발자 안내서의 규칙 보기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTopicRules를 참조하세요.

list-v2-logging-levels

다음 코드 예시에서는 `list-v2-logging-levels`의 사용 방법을 보여줍니다.

AWS CLI

로깅 수준 나열

다음 `list-v2-logging-levels` 예시에서는 구성된 로깅 수준을 나열합니다. 로깅 수준이 설정되지 않은 경우, 이 명령을 실행하면 `NotConfiguredException`이 발생합니다.

```
aws iot list-v2-logging-levels
```

출력:

```
{
    "logTargetConfigurations": [
        {
            "logTarget": {
                "targetType": "DEFAULT"
            },
            "logLevel": "ERROR"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListV2LoggingLevels를 참조하세요.

`list-violation-events`

다음 코드 예시에서는 `list-violation-events`의 사용 방법을 보여줍니다.

AWS CLI

특정 기간 동안 발생한 보안 프로필 위반 나열

다음 `list-violation-events` 예시에서는 현재 AWS 계정 및 AWS 리전의 모든 AWS IoT Device Defender 보안 프로필에 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 명령 탐지를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListViolationEvents를 참조하세요.

register-ca-certificate

다음 코드 예시에서는 `register-ca-certificate`의 사용 방법을 보여줍니다.

AWS CLI

인증 기관(CA) 인증서 등록

다음 `register-ca-certificate` 예시에서는 CA 인증서를 등록합니다. 명령은 CA 인증서와 CA 인증서에 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/
f4efed62c0142f16af278166f61962501165c4f0536295207426460058cd1467",
    "certificateId":
 "f4efed62c0142f16af278166f61962501165c4f0536295207426460058cd1467"
 }
```

자세한 내용은 AWS IoT API 참조의 RegisterCACertificate를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RegisterCaCertificate를 참조하세요.

register-certificate

다음 코드 예시에서는 `register-certificate`의 사용 방법을 보여줍니다.

AWS CLI

자체 서명된 디바이스 인증서 등록

다음 `register-certificate` 예시에서는 `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 AWS IoT API 참조의 RegisterCertificate를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RegisterCertificate를 참조하세요.

register-thing

다음 코드 예시에서는 register-thing의 사용 방법을 보여줍니다.

AWS CLI

사물 등록

다음 register-thing 예시에서는 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
kRn4amsUT21Ojg/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 AWS IoT Core 개발자 안내서의 신뢰할 수 있는 사용자에 의한 프로비저닝을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RegisterThing을 참조하세요.

reject-certificate-transfer

다음 코드 예시에서는 `reject-certificate-transfer`의 사용 방법을 보여줍니다.

AWS CLI

인증서 전송 거부

다음 `reject-certificate-transfer` 예시에서는 다른 AWS 계정으로부터의 지정된 디바이스 인증서 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 전송]을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [RejectCertificateTransfer]를 참조하세요.

`remove-thing-from-billing-group`

다음 코드 예시에서는 `remove-thing-from-billing-group`의 사용 방법을 보여줍니다.

AWS CLI

결제 그룹에서 사물 제거

다음 `remove-thing-from-billing-group` 예시에서는 지정된 사물을 결제 그룹에서 제거합니다.

```
aws iot remove-thing-from-billing-group \
    --billing-group-name GroupOne \
    --thing-name MyOtherLightBulb
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 [결제 그룹]을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [RemoveThingFromBillingGroup]을 참조하세요.

`remove-thing-from-thing-group`

다음 코드 예시에서는 `remove-thing-from-thing-group`의 사용 방법을 보여줍니다.

AWS CLI

사물 그룹에서 사물 제거

다음 `remove-thing-from-thing-group` 예시에서는 지정된 사물을 사물 그룹에서 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.

- API 세부 정보는 AWS CLI 명령 참조의 RemoveThingFromThingGroup을 참조하세요.

`replace-topic-rule`

다음 코드 예시에서는 `replace-topic-rule`의 사용 방법을 보여줍니다.

AWS CLI

주제의 규칙 정의 업데이트

다음 `replace-topic-rule` 예시에서는 지정된 규칙을 업데이트하여 토양 수분 수준의 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 AWS IoT 규칙 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ReplaceTopicRule을 참조하세요.

`search-index`

다음 코드 예시에서는 `search-index`의 사용 방법을 보여줍니다.

AWS CLI

사물 인덱스 쿼리

다음 `search-index` 예시에서는 `LightBulb`의 유형이 있는 사물의 `AWS_Things` 인덱스를 쿼리합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 SearchIndex를 참조하세요.

set-default-authorizer

다음 코드 예시에서는 `set-default-authorizer`의 사용 방법을 보여줍니다.

AWS CLI

기본 권한 부여자 설정

다음 `set-default-authorizer` 예시에서는 `CustomAuthorizer`라는 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 AWS IoT API 참조의 CreateDefaultAuthorizer를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 SetDefaultAuthorizer를 참조하세요.

set-default-policy-version

다음 코드 예시에서는 `set-default-policy-version`의 사용 방법을 보여줍니다.

AWS CLI

정책의 기본 버전 설정

다음 set-default-policy-version 예시에서는 UpdateDeviceCertPolicy라는 정책의 기본 버전을 2로 설정합니다.

```
aws iot set-default-policy-version \
    --policy-name UpdateDeviceCertPolicy \
    --policy-version-id 2
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 SetDefaultPolicyVersion을 참조하세요.

set-v2-logging-level

다음 코드 예시에서는 set-v2-logging-level의 사용 방법을 보여줍니다.

AWS CLI

사물 그룹의 로깅 수준 설정

다음 set-v2-logging-level 예시에서는 지정된 사물 그룹에 대한 경고를 로깅하도록 로깅 수준을 설정합니다.

```
aws iot set-v2-logging-level \
    --log-target "{\"targetType\":\"THING_GROUP\",\"targetName\":\"LightBulbs\"}" \
    --log-level WARN
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 SetV2LoggingLevel을 참조하세요.

set-v2-logging-options

다음 코드 예시에서는 set-v2-logging-options의 사용 방법을 보여줍니다.

AWS CLI

로깅 옵션 설정

다음 set-v2-logging-options 예시에서는 기본 로깅 세부 정보 수준을 ERROR로 설정하고 로깅에 사용할 ARN을 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 세부 정보는 AWS CLI 명령 참조의 SetV2LoggingOptions를 참조하세요.

start-audit-mitigation-actions-task

다음 코드 예시에서는 start-audit-mitigation-actions-task의 사용 방법을 보여줍니다.

AWS CLI

감사 결과에 완화 조치 적용

다음 start-audit-mitigation-actions-task 예시에서는 지정된 단일 결과에 ResetPolicyVersionAction 조치(정책을 삭제함)를 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 조치 명령)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartAuditMitigationActionsTask를 참조하세요.

start-on-demand-audit-task

다음 코드 예시에서는 start-on-demand-audit-task의 사용 방법을 보여줍니다.

AWS CLI

즉시 감사 시작

다음 start-on-demand-audit-task 예시에서는 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 명령 감사를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartOnDemandAuditTask를 참조하세요.

tag-resource

다음 코드 예시에서는 tag-resource의 사용 방법을 보여줍니다.

AWS CLI

리소스의 태그 키 및 값 지정

다음 tag-resource 예시에서는 키 Assembly와 값 Fact1NW가 있는 태그를 사물 그룹 LightBulbs에 적용합니다.

```
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 AWS IoT 리소스 태그 지정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

test-authorization

다음 코드 예시에서는 test-authorization의 사용 방법을 보여줍니다.

AWS CLI

AWS IoT 정책 테스트

다음 test-authorization 예시에서는 지정된 위탁자에 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 AWS IoT API 참조의 TestAuthorization을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TestAuthorization을 참조하세요.

test-invoke-authorizer

다음 코드 예시에서는 `test-invoke-authorizer`의 사용 방법을 보여줍니다.

AWS CLI

사용자 지정 권한 부여자 테스트

다음 `test-invoke-authorizer` 예시에서는 사용자 지정 권한 부여자를 테스트합니다.

```
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 AWS IoT API 참조의 TestInvokeAuthorizer를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TestInvokeAuthorizer를 참조하세요.

transfer-certificate

다음 코드 예시에서는 transfer-certificate의 사용 방법을 보여줍니다.

AWS CLI

디바이스 인증서를 다른 AWS 계정으로 전송

다음 transfer-certificate 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 TransferCertificate를 참조하세요.

untag-resource

다음 코드 예시에서는 `untag-resource`의 사용 방법을 보여줍니다.

AWS CLI

리소스에서 태그 키 제거

다음 `untag-resource` 예시에서는 사물 그룹 `LightBulbs`에서 태그 `MyTag`와 해당 값을 제거합니다.

```
command
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 AWS IoT 리소스 태그 지정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

update-account-audit-configuration

다음 코드 예시에서는 `update-account-audit-configuration`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 감사 알림에 대해 Amazon SNS 알림 활성화

다음 `update-account-audit-configuration` 예시에서는 AWS IoT Device Defender 감사 알림에 대한 Amazon SNS 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.

예시 2: 감사 검사 활성화

다음 `update-account-audit-configuration` 예시에서는 AWS IoT Device Defender의 `AUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK`라는 감사 검사를 활성화합니다. 감사 검사가 AWS 계정에서 하나 이상의 예약된 감사에 적용되는 `targetCheckNames`의 일부인 경우, 감사 검사를 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 명령 감사를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateAccountAuditConfiguration을 참조하세요.

update-audit-suppression

다음 코드 예시에서는 `update-audit-suppression`의 사용 방법을 보여줍니다.

AWS CLI

감사 결과 억제 업데이트

다음 `update-audit-suppression` 예시에서는 감사 결과 억제의 만료 날짜를 2020년 9월 21일로 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 AWS IoT 개발자 안내서의 감사 결과 억제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateAuditSuppression을 참조하세요.

update-authorizer

다음 코드 예시에서는 update-authorizer의 사용 방법을 보여줍니다.

AWS CLI

사용자 지정 권한 부여자 업데이트

다음 update-authorizer 예시에서는 CustomAuthorizer2의 상태를 INACTIVE로 업데이트합니다.

```
aws iot update-authorizer \
    --authorizer-name CustomAuthorizer2 \
    --status INACTIVE
```

출력:

```
{
    "authorizerName": "CustomAuthorizer2",
    "authorizerArn": "arn:aws:iot:us-west-2:123456789012:authorizer/
CustomAuthorizer2"
}
```

자세한 내용은 AWS IoT API 참조의 UpdateAuthorizer를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateAuthorizer를 참조하세요.

update-billing-group

다음 코드 예시에서는 update-billing-group의 사용 방법을 보여줍니다.

AWS CLI

결제 그룹 정보 업데이트

다음 update-billing-group 예시에서는 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 AWS IoT 개발자 안내서의 결제 그룹을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateBillingGroup을 참조하세요.

update-ca-certificate

다음 코드 예시에서는 update-ca-certificate의 사용 방법을 보여줍니다.

AWS CLI

인증 기관(CA) 인증서 업데이트

다음 update-ca-certificate 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateCaCertificate를 참조하세요.

update-certificate

다음 코드 예시에서는 update-certificate의 사용 방법을 보여줍니다.

AWS CLI

디바이스 인증서 업데이트

다음 update-certificate 예시에서는 지정된 디바이스 인증서를 INACTIVE 상태로 설정합니다.

```
aws iot update-certificate \
    --certificate-
id d1eb269fb55a628552143c8f96eb3c258fcd5331ea113e766ba0c82bf225f0be \
    --new-status INACTIVE
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT API 참조의 UpdateCertificate를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateCertificate를 참조하세요.

update-custom-metric

다음 코드 예시에서는 update-custom-metric의 사용 방법을 보여줍니다.

AWS CLI

사용자 지정 지표 업데이트

다음 update-custom-metric 예시에서는 사용자 지정 지표를 업데이트하여 새 display-name을 생성합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateCustomMetric을 참조하세요.

update-dimension

다음 코드 예시에서는 update-dimension의 사용 방법을 보여줍니다.

AWS CLI

측정기준 업데이트

다음 update-dimension 예시에서는 측정기준을 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 측정기준을 사용하여 보안 프로필의 지표 범위 지정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateDimension을 참조하세요.

update-domain-configuration

다음 코드 예시에서는 update-domain-configuration의 사용 방법을 보여줍니다.

AWS CLI

도메인 구성 업데이트

다음 update-domain-configuration 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateDomainConfiguration을 참조하세요.

`update-dynamic-thing-group`

다음 코드 예시에서는 `update-dynamic-thing-group`의 사용 방법을 보여줍니다.

AWS CLI

동적 사물 그룹 업데이트

다음 `update-dynamic-thing-group` 예시에서는 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 AWS IoT 개발자 안내서의 동적 사물 그룹을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateDynamicThingGroup을 참조하세요.

update-event-configurations

다음 코드 예시에서는 update-event-configurations의 사용 방법을 보여줍니다.

AWS CLI

게시되는 이벤트 유형 표시

다음 update-event-configurations 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateEventConfigurations를 참조하세요.

update-indexing-configuration

다음 코드 예시에서는 update-indexing-configuration의 사용 방법을 보여줍니다.

AWS CLI

사물 인덱싱 활성화

다음 update-indexing-configuration 예시에서는 AWS_Things 인덱스를 사용하여 레지스트리 데이터, 섀도우 데이터 및 사물 연결 상태 검색을 지원하도록 사물 인덱싱을 활성화합니다.

```
aws iot update-indexing-configuration
    --thing-indexing-
configuration thingIndexingMode=REGISTRY_AND_SHADOW,thingConnectivityIndexingMode=STATUS
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 사물 인덱싱 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateIndexingConfiguration을 참조하세요.

update-job

다음 코드 예시에서는 update-job의 사용 방법을 보여줍니다.

AWS CLI

작업의 세부 상태 가져오기

다음 update-job 예시에서는 ID가 example-job-01인 작업의 세부 상태를 가져옵니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateJob을 참조하세요.

update-mitigation-action

다음 코드 예시에서는 update-mitigation-action의 사용 방법을 보여줍니다.

AWS CLI

완화 조치 업데이트

다음 update-mitigation-action 예시에서는 AddThingsToQuarantineGroupAction이라는 지정된 완화 조치를 업데이트하고 사물 그룹 이름을 변경한 다음, overrideDynamicGroups를 false로 설정합니다. 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 조치 명령)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateMitigationAction을 참조하세요.

update-provisioning-template

다음 코드 예시에서는 update-provisioning-template의 사용 방법을 보여줍니다.

AWS CLI

프로비저닝 템플릿 업데이트

다음 `update-provisioning-template` 예시에서는 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 AWS IoT 보안 터널링을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateProvisioningTemplate을 참조하세요.

`update-role-alias`

다음 코드 예시에서는 `update-role-alias`의 사용 방법을 보여줍니다.

AWS CLI

역할 별칭 업데이트

다음 `update-role-alias` 예시에서는 `LightBulbRole` 별칭 역할을 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 AWS IoT API 참조의 UpdateRoleAlias를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateRoleAlias를 참조하세요.

update-scheduled-audit

다음 코드 예시에서는 update-scheduled-audit의 사용 방법을 보여줍니다.

AWS CLI

예약된 감사 정의 업데이트

다음 update-scheduled-audit 예시에서는 AWS IoT Device Defender의 예약된 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 명령 감사를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateScheduledAudit을 참조하세요.

update-security-profile

다음 코드 예시에서는 update-security-profile의 사용 방법을 보여줍니다.

AWS CLI

보안 프로필 변경

다음 update-security-profile 예시에서는 AWS IoT Device Defender의 보안 프로필에 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 명령 탐지를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateSecurityProfile을 참조하세요.

update-stream

다음 코드 예시에서는 update-stream의 사용 방법을 보여줍니다.

AWS CLI

스트림 업데이트

다음 update-stream 예시에서는 기존 스트림을 업데이트합니다. 스트림 버전이 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 AWS IoT API 참조의 UpdateStream을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateStream을 참조하세요.

update-thing-group

다음 코드 예시에서는 update-thing-group의 사용 방법을 보여줍니다.

AWS CLI

사물 그룹의 정의 업데이트

다음 update-thing-group 예시에서는 지정된 사물 그룹의 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 AWS IoT 개발자 안내서의 사물 그룹을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateThingGroup을 참조하세요.

update-thing-groups-for-thing

다음 코드 예시에서는 update-thing-groups-for-thing의 사용 방법을 보여줍니다.

AWS CLI

사물이 속하는 그룹 변경

다음 `update-thing-groups-for-thing` 예시에서는 `DeadBulbs`라는 그룹에서 `MyLightBulb`라는 사물을 제거하는 동시에, 해당 사물을 `replaceableItems`라는 그룹에 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 그룹]을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [UpdateThingGroupsForThing]을 참조하세요.

`update-thing`

다음 코드 예시에서는 `update-thing`의 사용 방법을 보여줍니다.

AWS CLI

사물에 사물 유형 연결

다음 `update-thing` 예시에서는 AWS IoT 레지스트리의 사물에 사물 유형을 연결합니다. 연결하면 사물 유형에 의해 정의된 속성의 값을 입력합니다.

```
aws iot update-thing \
    --thing-name "MyOtherLightBulb" \
    --thing-type-name "LightBulb" \
    --attribute-payload "{"attributes": {"wattage":"75", "model":"123"}}"
```

이 명령은 출력을 생성하지 않습니다. 결과를 표시하려면 `describe-thing` 명령을 사용합니다.

자세한 내용은 AWS IoT 개발자 안내서의 [사물 유형]을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [UpdateThing]을 참조하세요.

`update-topic-rule-destination`

다음 코드 예시에서는 `update-topic-rule-destination`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 주제 규칙 대상 활성화

다음 `update-topic-rule-destination` 예시에서는 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 활성화]를 참조하세요.

예시 2: 주제 규칙 대상 비활성화

다음 `update-topic-rule-destination` 예시에서는 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 비활성화]를 참조하세요.

예시 3: 새 확인 메시지 전송

다음 `update-topic-rule-destination` 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateTopicRuleDestination을 참조하세요.

`validate-security-profile-behaviors`

다음 코드 예시에서는 `validate-security-profile-behaviors`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 보안 프로필의 동작 파라미터 검증

다음 `validate-security-profile-behaviors` 예시에서는 AWS IoT Device Defender 보안 프로필에서 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": []
}
```

예시 2: 보안 프로필의 잘못된 동작 파라미터 검증

다음 `validate-security-profile-behaviors` 예시에서는 AWS IoT Device Defender 보안 프로필에서 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 명령 탐지를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ValidateSecurityProfileBehaviors를 참조하세요.

AWS CLI를 사용한 AWS IoT Analytics 예시

다음 코드 예시에서는 AWS IoT Analytics에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`batch-put-message`

다음 코드 예시에서는 `batch-put-message` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

메시지를 채널에 전송

다음 batch-put-message 예시에서는 메시지를 지정된 채널로 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 AWS IoT Analytics API 참조의 BatchPutMessage를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 BatchPutMessage 섹션을 참조하세요.

cancel-pipeline-reprocessing

다음 코드 예시에서는 cancel-pipeline-reprocessing 코드를 사용하는 방법을 보여줍니다.

AWS CLI

파이프라인을 통한 데이터 재처리 취소

다음 cancel-pipeline-reprocessing 예시에서는 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 AWS IoT Analytics API 참조의 CancelPipelineReprocessing을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CancelPipelineReprocessing 섹션을 참조하세요.

create-channel

다음 코드 예시에서는 `create-channel` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

채널 생성

다음 `create-channel` 예시에서는 지정된 구성을 사용하여 채널을 생성합니다. 채널은 MQTT 주제로부터 데이터를 수집하고, 데이터를 파이프라인에 게시하기 전에 처리되지 않은 원시 메시지를 보관합니다.

```
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 AWS IoT Analytics API 참조의 CreateChannel을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateChannel 섹션을 참조하세요.

create-dataset-content

다음 코드 예시에서는 create-dataset-content 코드를 사용하는 방법을 보여줍니다.

AWS CLI

데이터세트의 콘텐츠를 생성하는 방법

다음 create-dataset-content 예시에서는 queryAction(SQL 쿼리) 또는 containerAction(컨테이너화된 애플리케이션 실행)을 적용하여, 지정된 데이터세트의 콘텐츠를 생성합니다.

```
aws iotanalytics create-dataset-content \
    --dataset-name mydataset
```

출력:

```
{
    "versionId": "d494b416-9850-4670-b885-ca22f1e89d62"
}
```

자세한 내용은 AWS IoT Analytics API 참조의 CreateDatasetContent를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateDatasetContent 섹션을 참조하세요.

create-dataset

다음 코드 예시에서는 create-dataset 코드를 사용하는 방법을 보여줍니다.

AWS CLI

데이터세트 생성

다음 `create-dataset` 예시에서는 데이터세트를 생성합니다. 데이터세트는 `queryAction`(SQL 쿼리) 또는 `containerAction`(컨테이너식 애플리케이션 실행)을 적용하여 데이터 스토어에서 검색된 데이터를 저장합니다. 이 작업은 데이터세트 스켈레톤을 생성합니다. 데이터세트는 `CreateDatasetContent`를 호출해서 수동으로, 또는 지정한 `trigger`에 따라 자동으로 채울 수 있습니다.

```
aws iotanalytics create-dataset \
    --cli-input-json file://create-dataset.json
```

`create-dataset.json`의 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 AWS IoT Analytics API 참조의 CreateDataset를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateDataset 섹션을 참조하세요.

create-datastore

다음 코드 예시에서는 `create-datastore` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

데이터 스토어 생성

다음 `create-datastore` 예시에서는 메시지의 저장소인 데이터 저장소를 만듭니다.

```
aws iotanalytics create-datastore \
    --cli-input-json file://create-datastore.json
```

`create-datastore.json`의 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 AWS IoT Analytics API 참조의 CreateDatastore를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateDatastore를 참조하세요.

create-pipeline

다음 코드 예시에서는 `create-pipeline` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

IoT Analytics 파이프라인 생성

다음 `create-pipeline` 예시에서는 파이프라인을 생성합니다. 파이프라인은 채널에서 전송된 메시지를 사용하고, 사용자가 메시지를 데이터 스토어에 저장하기 전에 처리할 수 있도록 합니다. `pipelineActivities` 배열에 채널 및 데이터 저장소 활동(선택적으로 23개의 추가 활동)을 모두 지정해야 합니다.

```
aws iotanalytics create-pipeline \
    --cli-input-json file://create-pipeline.json
```

`create-pipeline.json`의 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 AWS IoT Analytics API 참조의 CreatePipeline을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreatePipeline 섹션을 참조하세요.

`delete-channel`

다음 코드 예시에서는 `delete-channel` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

IoT Analytics 채널 삭제

다음 `delete-channel` 예시에서는 지정된 채널을 삭제합니다.

```
aws iotanalytics delete-channel \
    --channel-name mychannel
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT Analytics API 참조의 DeleteChannel을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteChannel 섹션을 참조하세요.

delete-dataset-content

다음 코드 예시에서는 delete-dataset-content 코드를 사용하는 방법을 보여줍니다.

AWS CLI

데이터세트의 콘텐츠를 삭제하는 방법

다음 delete-dataset-content 예시에서는 지정된 데이터세트의 콘텐츠를 삭제합니다.

```
aws iotanalytics delete-dataset-content \
    --dataset-name mydataset
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT Analytics API 참조의 DeleteDatasetContent를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteDatasetContent 섹션을 참조하세요.

delete-dataset

다음 코드 예시에서는 delete-dataset 코드를 사용하는 방법을 보여줍니다.

AWS CLI

데이터세트 삭제

다음 delete-dataset 예시에서는 지정된 데이터세트를 삭제합니다. 이 작업을 수행하기 전에 데이터세트의 내용을 삭제할 필요는 없습니다.

```
aws iotanalytics delete-dataset \
```

```
    --dataset-name mydataset
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT Analytics API 참조의 DeleteDataset를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteDataset 섹션을 참조하세요.

delete-datastore

다음 코드 예시에서는 delete-datastore 코드를 사용하는 방법을 보여줍니다.

AWS CLI

데이터 스토어 삭제

다음 delete-datastore 예시에서는 지정된 데이터 저장소를 삭제합니다.

```
aws iotanalytics delete-datastore \
    --datastore-name mydatastore
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT Analytics API 참조의 DeleteDatastore를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteDatastore를 참조하세요.

delete-pipeline

다음 코드 예시에서는 delete-pipeline 코드를 사용하는 방법을 보여줍니다.

AWS CLI

파이프라인 삭제

다음 delete-pipeline 예시에서는 지정한 파이프라인을 삭제합니다.

```
aws iotanalytics delete-pipeline \
    --pipeline-name mypipeline
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT Analytics API 참조의 DeletePipeline을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeletePipeline 섹션을 참조하세요.

describe-channel

다음 코드 예시에서는 describe-channel 코드를 사용하는 방법을 보여줍니다.

AWS CLI

채널에 대한 정보 검색

다음 describe-channel 예시에서는 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 AWS IoT Analytics API 참조의 DescribeChannel을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeChannel 섹션을 참조하세요.

describe-dataset

다음 코드 예시에서는 describe-dataset 코드를 사용하는 방법을 보여줍니다.

AWS CLI

데이터세트에 대한 정보 검색

다음 describe-dataset 예시에서는 지정된 데이터세트에 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 AWS IoT Analytics API 참조의 DescribeDataset를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDataset 섹션을 참조하세요.

`describe-datastore`

다음 코드 예시에서는 `describe-datastore` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

데이터 저장소에 대한 정보 검색

다음 `describe-datastore` 예시에서는 지정된 데이터 저장소에 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 AWS IoT Analytics API 참조의 DescribeDatastore를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDatastore 섹션을 참조하세요.

`describe-logging-options`

다음 코드 예시에서는 `describe-logging-options` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

현재 로깅 옵션을 검색하는 방법

다음 `describe-logging-options` 예시에서는 현재 AWS IoT Analytics 로깅 옵션을 표시합니다.

```
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 AWS IoT Analytics API 참조의 DescribeLoggingOptions를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeLoggingOptions 섹션을 참조하세요.

describe-pipeline

다음 코드 예시에서는 `describe-pipeline` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

파이프라인에 대한 정보 검색

다음 `describe-pipeline` 예시에서는 지정된 파이프라인의 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 AWS IoT Analytics API 참조의 DescribePipeline을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribePipeline 섹션을 참조하세요.

get-dataset-content

다음 코드 예시에서는 `get-dataset-content` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

데이터세트의 콘텐츠를 검색하는 방법

다음 `get-dataset-content` 예시에서는 데이터세트의 내용을 미리 서명된 URI로 검색합니다.

```
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 안내서의 GetDatasetContent를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetDatasetContent 섹션을 참조하세요.

`list-channels`

다음 코드 예시에서는 `list-channels` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

채널 목록 검색

다음 `list-channels` 예시에서는 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 AWS IoT Analytics API 참조의 ListChannels를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListChannels를 참조하세요.

list-dataset-contents

다음 코드 예시에서는 `list-dataset-contents` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

데이터세트 콘텐츠에 대한 정보를 나열하는 방법

다음 `list-dataset-contents` 예시에서는 생성한 데이터세트 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 AWS IoT Analytics API 참조의 ListDatasetContents를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListDatasetContents 섹션을 참조하세요.

list-datasets

다음 코드 예시에서는 list-datasets 코드를 사용하는 방법을 보여줍니다.

AWS CLI

데이터세트에 대한 정보 검색

다음 list-datasets 예시에서는 사용 가능한 데이터세트에 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 AWS IoT Analytics API 참조의 ListDatasets를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListDatasets 섹션을 참조하세요.

list-datastores

다음 코드 예시에서는 list-datastores 코드를 사용하는 방법을 보여줍니다.

AWS CLI

데이터 저장소 목록 검색

다음 `list-datastores` 예시에서는 사용 가능한 데이터 저장소에 대한 요약 정보를 표시합니다.

```
aws iotanalytics list-datastores
```

출력:

```
{
    "datastoreSummaries": [
        {
            "status": "ACTIVE",
            "datastoreName": "mydatastore",
            "creationTime": 1557858971.02,
            "lastUpdateTime": 1557858971.02
        }
    ]
}
```

자세한 내용은 AWS IoT Analytics API 참조의 ListDatastores를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListDatastores를 참조하세요.

list-pipelines

다음 코드 예시에서는 `list-pipelines` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

파이프라인 목록 검색

다음 `list-pipelines` 예시에서는 사용 가능한 파이프라인 목록을 표시합니다.

```
aws iotanalytics list-pipelines
```

출력:

```
{
    "pipelineSummaries": [
        {
            "pipelineName": "mypipeline",
            "creationTime": 1557859124.432,
            "lastUpdateTime": 1557859124.432,
            "reprocessingSummaries": []
```

```
        }
    ]
}
```

자세한 내용은 AWS IoT Analytics API 참조의 ListPipelines를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListPipelines 섹션을 참조하세요.

list-tags-for-resource

다음 코드 예시에서는 `list-tags-for-resource` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

리소스의 태그 나열

다음 `list-tags-for-resource` 예시에서는 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 AWS IoT Analytics API 참조의 ListTagsForResource를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource를 참조하세요.

put-logging-options

다음 코드 예시에서는 `put-logging-options` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

로깅 옵션을 설정하거나 업데이트하는 방법

다음 put-logging-options 예시에서는 AWS IoT Analytics 로깅 옵션을 설정하거나 업데이트합니다. loggingOptions 필드 값을 업데이트한 경우 변경 사항이 적용되기까지 최대 1분이 소요됩니다. 또한 'roleArn' 필드에서 지정한 역할에 연결된 정책을 변경하는 경우(예: 잘못된 정책 수정), 변경 사항이 적용되기까지 최대 5분이 소요됩니다.

```
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 AWS IoT Analytics API 참조의 PutLoggingOptions를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutLoggingOptions 섹션을 참조하세요.

run-pipeline-activity

다음 코드 예시에서는 run-pipeline-activity 코드를 사용하는 방법을 보여줍니다.

AWS CLI

파이프라인 활동을 시뮬레이션하는 방법

다음 run-pipeline-activity 예시에서는 메시지 페이로드에서 파이프라인 활동을 실행한 결과를 시뮬레이션합니다.

```
aws iotanalytics run-pipeline-activity \
    --pipeline-activity file://maths.json \
    --payloads file://payloads.json
```

maths.json의 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 AWS IoT Analytics API 참조의 RunPipelineActivity를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RunPipelineActivity 섹션을 참조하세요.

sample-channel-data

다음 코드 예시에서는 `sample-channel-data` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

채널에서 샘플 메시지를 검색하는 방법

다음 `sample-channel-data` 예시에서는 지정된 기간 동안 지정된 채널에서 수집한 메시지의 샘플을 가져옵니다. 메시지를 최대 10개 가져올 수 있습니다.

```
aws iotanalytics sample-channel-data \
```

```
    --channel-name mychannel
```

출력:


```
{
    "payloads": [
        "eyAidGVtcGVyYXR1cmUiOiAyMCB9",
        "eyAiZm9vIjogImJhciIgfQ=="
    ]
}
```


자세한 내용은 AWS IoT Analytics API 참조의 SampleChannelData를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 SampleChannelData 섹션을 참조하세요.

start-pipeline-reprocessing

다음 코드 예시에서는 start-pipeline-reprocessing 코드를 사용하는 방법을 보여줍니다.

AWS CLI

파이프라인 재처리를 시작하는 방법

다음 start-pipeline-reprocessing 예시에서는 지정된 파이프라인을 통한 원시 메시지 데이터의 재처리를 시작합니다.

```
aws iotanalytics start-pipeline-reprocessing \
    --pipeline-name mypipeline
```

출력:

```
{
    "reprocessingId": "6ad2764f-fb13-4de3-b101-4e74af03b043"
}
```

자세한 내용은 AWS IoT Analytics API 참조의 StartPipelineReprocessing을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartPipelineReprocessing 섹션을 참조하세요.

tag-resource

다음 코드 예시에서는 tag-resource 코드를 사용하는 방법을 보여줍니다.

AWS CLI

리소스에서 태그 추가 또는 수정

다음 `tag-resource` 예시에서는 지정된 리소스의 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 AWS IoT Analytics API 참조의 TagResource를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

`untag-resource`

다음 코드 예시에서는 `untag-resource` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

리소스에서 태그 제거

다음 `untag-resource` 예시에서는 지정된 리소스에서 지정된 키 이름을 가진 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 >를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

`update-channel`

다음 코드 예시에서는 `update-channel` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

채널을 수정하는 방법

다음 update-channel 예시에서는 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 AWS IoT Analytics API 참조의 UpdateChannel을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateChannel 섹션을 참조하세요.

update-dataset

다음 코드 예시에서는 update-dataset 코드를 사용하는 방법을 보여줍니다.

AWS CLI

데이터세트 업데이트

다음 update-dataset 예시에서는 지정된 데이터세트의 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 >를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateDataset 섹션을 참조하세요.

update-datastore

다음 코드 예시에서는 update-datastore 코드를 사용하는 방법을 보여줍니다.

AWS CLI

데이터 저장소 생성

다음 update-datastore 예시에서는 지정된 데이터 저장소의 설정을 수정합니다.

```
aws iotanalytics update-datastore \
    --cli-input-json file://update-datastore.json
```

update-datastore.json의 콘텐츠:

```
{
    "datastoreName": "mydatastore",
    "retentionPeriod": {
        "numberOfDays": 93
    }
}
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT Analytics API 참조의 UpdateDatastore를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateDatastore 섹션을 참조하세요.

update-pipeline

다음 코드 예시에서는 update-pipeline 코드를 사용하는 방법을 보여줍니다.

AWS CLI

파이프라인 업데이트

다음 update-pipeline 예시에서는 지정된 파이프라인의 설정을 수정합니다. pipelineActivities 배열에 채널 및 데이터 저장소 활동(선택적으로 23개의 추가 활동)을 모두 지정해야 합니다.

```
aws iotanalytics update-pipeline \
    --cli-input-json file://update-pipeline.json
```

update-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) + 273.15",
                "attribute": "tempK",
```

```
                "next": "myDatastoreActivity"
            }
        }
    ]
}
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT Analytics API 참조의 UpdatePipeline을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdatePipeline 섹션을 참조하세요.

AWS CLI를 사용한 Device Advisor 예시

다음 코드 예시에서는 Device Advisor와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`create-suite-definition`

다음 코드 예시에서는 `create-suite-definition`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: IoT Device Advisor 테스트 도구 모음 생성

다음 `create-suite-definition` 예시에서는 지정된 도구 모음 정의 구성을 사용하여 AWS IoT에 Device Advisor 테스트 도구 모음을 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 도구 모음 생성을 참조하세요.

예시 2: IoT Device Advisor 최신 자격 테스트 도구 모음 생성

다음 `create-suite-definition` 예시에서는 지정된 도구 모음 정의 구성을 사용하여 AWS IoT에 최신 버전의 Device Advisor 자격 테스트 도구 모음을 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 도구 모음 생성](#)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [CreateSuiteDefinition](#)을 참조하세요.

`delete-suite-definition`

다음 코드 예시에서는 `delete-suite-definition`의 사용 방법을 보여줍니다.

AWS CLI

IoT Device Advisor 테스트 도구 모음 삭제

다음 `delete-suite-definition` 예시에서는 지정된 도구 모음 정의 ID로 Device Advisor 테스트 도구 모음을 삭제합니다.

```
aws iotdeviceadvisor delete-suite-definition \
    --suite-definition-id 0jtsgio7yenu
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT API 참조의 [DeleteSuiteDefinition](#)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteSuiteDefinition](#)을 참조하세요.

`get-endpoint`

다음 코드 예시에서는 `get-endpoint`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: IoT Device Advisor 계정 수준 엔드포인트의 정보 가져오기

다음 `get-endpoint` 예시에서는 Device Advisor 계정 수준 테스트 엔드포인트의 정보를 가져옵니다.

```
aws iotdeviceadvisor get-endpoint
```

출력:

```
{
    "endpoint": "t6y4c143x9sfo.deviceadvisor.iot.us-east-1.amazonaws.com"
}
```

예시 2: IoT Device Advisor 디바이스 수준 엔드포인트의 정보 가져오기

다음 `get-endpoint` 예시에서는 지정된 사물 ARN 또는 인증서 ARN을 사용하는 Device Advisor 디바이스 수준 테스트 엔드포인트의 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 엔드드포인트 가져오기]를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [GetEndpoint]를 참조하세요.

get-suite-definition

다음 코드 예시에서는 `get-suite-definition`의 사용 방법을 보여줍니다.

AWS CLI

IoT Device Advisor 테스트 도구 모음의 정보 가져오기

다음 `get-suite-definition` 예시에서는 지정된 도구 모음 정의 ID로 Device Advisor 테스트 도구 모음의 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 도구 모음 정의 가져오기]를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [GetSuiteDefinition]을 참조하세요.

get-suite-run-report

다음 코드 예시에서는 `get-suite-run-report`의 사용 방법을 보여줍니다.

AWS CLI

IoT Device Advisor 자격 테스트 도구 모음 실행 보고서의 정보 가져오기

다음 `get-suite-run-report` 예시에서는 지정된 도구 모음 정의 ID 및 도구 모음 실행 ID로 성공적인 Device Advisor 자격 테스트 도구 모음 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 도구 모음 실행을 위한 자격 보고서 가져오기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetSuiteRunReport를 참조하세요.

get-suite-run

다음 코드 예시에서는 get-suite-run의 사용 방법을 보여줍니다.

AWS CLI

IoT Device Advisor 테스트 도구 모음 실행 상태의 정보 가져오기

다음 get-suite-run 예시에서는 지정된 도구 모음 정의 ID 및 도구 모음 실행 ID로 Device Advisor 테스트 도구 모음 실행 상태의 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 도구 모음 실행 가져오기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetSuiteRun을 참조하세요.

list-suite-definitions

다음 코드 예시에서는 `list-suite-definitions`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 생성한 IoT Device Advisor 테스트 도구 모음 나열

다음 `list-suite-definitions` 예시에서는 AWS IoT에서 생성한 최대 25개의 Device Advisor 테스트 도구 모음을 나열합니다. 테스트 도구 모음이 25개를 초과하는 경우 출력에 'nextToken'이 표시됩니다. 이 'nextToken'을 사용하면 생성한 나머지 테스트 도구 모음을 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"
}
```

예시 2: 지정된 설정으로 생성한 IoT Device Advisor 테스트 도구 모음 나열

다음 `list-suite-definitions` 예시에서는 AWS IoT에서 생성한 Device Advisor 테스트 도구 모음을 지정된 최대 결과 개수 내에서 나열합니다. 테스트 도구 모음이 최대 개수보다 많은 경우 출력에 'nextToken'이 표시됩니다. 'nextToken'이 있는 경우 'nextToken'을 사용하여 이전에 표시되지 않은 생성한 테스트 도구 모음을 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 AWS IoT API 참조의 ListSuiteDefinitions를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListSuiteDefinitions를 참조하세요.

list-suite-runs

다음 코드 예시에서는 `list-suite-runs`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 지정된 IoT Device Advisor 테스트 도구 모음 실행 상태의 모든 정보 나열

다음 `list-suite-runs` 예시에서는 Device Advisor 테스트 도구 모음 실행 상태의 모든 정보를 지정된 도구 모음 정의 ID를 사용하여 나열합니다. 테스트 도구 모음 실행이 25개를 초과하는 경우 출력에 'nextToken'이 표시됩니다. 이 'nextToken'을 사용하면 나머지 테스트 도구 모음 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
        }
    ]
}
```

예시 2: 지정된 IoT Device Advisor 테스트 도구 모음 실행 상태의 정보를 지정된 설정으로 나열

다음 `list-suite-runs` 예시에서는 Device Advisor 테스트 도구 모음 실행 상태의 정보를 지정된 도구 모음 정의 ID를 사용하여 최대 결과 개수 내에서 나열합니다. 테스트 도구 모음 실행이 최대 개수를 초과한 경우 출력에 'nextToken'이 표시됩니다. 'nextToken'이 있는 경우 'nextToken'을 사용하여 이전에 표시되지 않은 테스트 도구 모음 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 AWS IoT API 참조의 ListSuiteRuns를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListSuiteRuns를 참조하세요.

list-tags-for-resource

다음 코드 예시에서는 `list-tags-for-resource`의 사용 방법을 보여줍니다.

AWS CLI

IoT Device Advisor 리소스에 연결된 태그 나열

다음 `list-tags-for-resource` 예시에서는 Device Advisor 리소스에 연결된 태그를 나열합니다. Device Advisor 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 AWS IoT API 참조의 ListTagsForResource 및 서비스 승인 참조의 AWS IoT Core Device Advisor에 의해 정의된 리소스 유형을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource를 참조하세요.

start-suite-run

다음 코드 예시에서는 `start-suite-run`의 사용 방법을 보여줍니다.

AWS CLI

IoT Device Advisor 테스트 도구 모음 실행 시작

다음 `start-suite-run` 예시에서는 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 도구 모음 실행 시작을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartSuiteRun을 참조하세요.

stop-suite-run

다음 코드 예시에서는 `stop-suite-run`의 사용 방법을 보여줍니다.

AWS CLI

현재 실행 중인 IoT Device Advisor 테스트 도구 모음 중지

다음 `stop-suite-run` 예시에서는 지정된 도구 모음 정의 ID 및 도구 모음 실행 ID로 현재 실행 중인 Device Advisor 테스트 도구 모음을 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 도구 모음 실행 중지를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StopSuiteRun을 참조하세요.

tag-resource

다음 코드 예시에서는 `tag-resource`의 사용 방법을 보여줍니다.

AWS CLI

IoT Device Advisor 리소스의 기존 태그를 추가 및 수정

다음 `tag-resource` 예시에서는 지정된 리소스 ARN 및 태그를 사용하여 Device Advisor 리소스의 기존 태그를 추가 및 수정합니다. Device Advisor 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 AWS IoT API 참조의 TagResource 및 서비스 승인 참조의 AWS IoT Core Device Advisor에 의해 정의된 리소스 유형을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

untag-resource

다음 코드 예시에서는 `untag-resource`의 사용 방법을 보여줍니다.

AWS CLI

IoT Device Advisor 리소스에서 기존 태그 제거

다음 `untag-resource` 예시에서는 지정된 리소스 ARN 및 태그 키를 사용하여 Device Advisor 리소스에서 기존 태그를 제거합니다. Device Advisor 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 AWS IoT API 참조의 UntagResource 및 서비스 승인 참조의 AWS IoT Core Device Advisor에 의해 정의된 리소스 유형을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

update-suite-definition

다음 코드 예시에서는 update-suite-definition의 사용 방법을 보여줍니다.

AWS CLI

예시 1: IoT Device Advisor 테스트 도구 모음 업데이트

다음 update-suite-definition 예시에서는 AWS IoT의 Device Advisor 테스트 도구 모음을 지정된 도구 모음 정의 ID 및 도구 모음 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"
}
```

예시 2: IoT Device Advisor 자격 테스트 도구 모음 업데이트

다음 update-suite-definition 예시에서는 AWS IoT의 Device Advisor 자격 테스트 도구 모음을 지정된 도구 모음 정의 ID 및 도구 모음 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 AWS IoT API 참조의 UpdateSuiteDefinition을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateSuiteDefinition을 참조하세요.

AWS CLI를 사용한 AWS IoT data 예시

다음 코드 예시에서는 AWS IoT data에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`delete-thing-shadow`

다음 코드 예시에서는 `delete-thing-shadow`의 사용 방법을 보여줍니다.

AWS CLI

디바이스의 섀도우 문서 삭제

다음 delete-thing-shadow 예시에서는 MyRPi라는 디바이스의 전체 섀도우 문서를 삭제합니다.

```
aws iot-data delete-thing-shadow \
    --thing-name MyRPi \
    "output.txt"
```

이 명령은 디스플레이에 출력을 생성하지 않지만 삭제한 섀도우 문서의 버전과 타임스탬프를 확인하는 정보가 output.txt에 포함되어 있습니다.

```
{"version":2,"timestamp":1560270384}
```

자세한 내용은 AWS IoT 개발자 안내서의 섀도우 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteThingShadow를 참조하세요.

get-thing-shadow

다음 코드 예시에서는 get-thing-shadow의 사용 방법을 보여줍니다.

AWS CLI

사물 섀도우 문서 가져오기

다음 get-thing-shadow 예시에서는 지정된 IoT 사물의 사물 섀도우 문서를 가져옵니다.

```
aws iot-data get-thing-shadow \
    --thing-name MyRPi \
    output.txt
```

이 명령은 디스플레이에 출력을 생성하지 않지만, 다음은 output.txt의 내용을 보여줍니다.

```
{
  "state":{
    "reported":{
    "moisture":"low"
    }
  },
```

```
  "metadata":{
    "reported":{
      "moisture":{
        "timestamp":1560269319
      }
    }
  },
  "version":1,"timestamp":1560269405
}
```

자세한 내용은 AWS IoT 개발자 안내서의 디바이스 섀도우 서비스 데이터 흐름을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetThingShadow를 참조하세요.

update-thing-shadow

다음 코드 예시에서는 update-thing-shadow의 사용 방법을 보여줍니다.

AWS CLI

사물 섀도우 업데이트

다음 update-thing-shadow 예시에서는 지정된 사물의 현재 디바이스 섀도우 상태를 수정하고 파일 output.txt에 저장합니다.

```
aws iot-data update-thing-shadow \
    --thing-name MyRPi \
    --payload "{"state":{"reported":{"moisture":"okay"}}}" \
    "output.txt"
```

이 명령은 디스플레이에 출력을 생성하지 않지만, 다음은 output.txt의 내용을 보여줍니다.

```
{
    "state": {
        "reported": {
            "moisture": "okay"
        }
    },
    "metadata": {
        "reported": {
            "moisture": {
                "timestamp": 1560270036
            }
```

```
        }
    },
    "version": 2,
    "timestamp": 1560270036
}
```

자세한 내용은 AWS IoT 개발자 안내서의 [디바이스 섀도우 서비스 데이터 흐름]을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [UpdateThingShadow]를 참조하세요.

AWS CLI를 사용한 AWS IoT Events 예시

다음 코드 예시에서는 AWS IoT Events에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

작업

batch-put-message

다음 코드 예시에서는 `batch-put-message`의 사용 방법을 보여줍니다.

AWS CLI

AWS IoT Events로 메시지(입력) 전송

다음 `batch-put-message` 예시에서는 AWS IoT Events 시스템으로 메시지 세트를 전송합니다. 각 메시지 페이로드는 지정한 입력(`inputName`)으로 변환되고 해당 입력을 모니터링하는 모든 탐지기로 수집됩니다. 메시지가 여러 개 전송되는 경우 메시지가 처리되는 순서가 보장되지 않습니다. 순서를 보장하려면 메시지를 한 번에 하나씩 보내고 응답이 성공할 때까지 기다려야 합니다.

```
aws iotevents-data batch-put-message \
    --cli-input-json file://highPressureMessage.json
```

highPressureMessage.json의 콘텐츠:

```
{
    "messages": [
        {
            "messageId": "00001",
            "inputName": "PressureInput",
            "payload": "{\"motorid\": \"Fulton-A32\", \"sensorData\": {\"pressure\":
 80, \"temperature\": 39} }"
        }
    ]
}
```

출력:

```
{
    "BatchPutMessageErrorEntries": []
}
```

자세한 내용은 AWS IoT Events API 참조의 BatchPutMessage를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 BatchPutMessage를 참조하세요.

batch-update-detector

다음 코드 예시에서는 batch-update-detector의 사용 방법을 보여줍니다.

AWS CLI

탐지기(인스턴스) 업데이트

다음 batch-update-detector 예시에서는 지정된 탐지기 모델의 하나 이상의 탐지기(인스턴스)의 상태, 변숫값 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 API 참조의 BatchUpdateDetector를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 BatchUpdateDetector를 참조하세요.

create-detector-model

다음 코드 예시에서는 `create-detector-model`의 사용 방법을 보여줍니다.

AWS CLI

탐지기 모델 생성

다음 `create-detector-model` 예시에서는 파라미터 파일에 지정된 구성을 사용하여 탐지기 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 AWS IoT Events API 참조의 CreateDetectorModel을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateDetectorModel을 참조하세요.

create-input

다음 코드 예시에서는 create-input의 사용 방법을 보여줍니다.

AWS CLI

입력 생성

다음 create-input 예시에서는 입력을 생성합니다.

```
aws iotevents create-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": 1560795312.542,
        "creationTime": 1560795312.542,
        "inputName": "PressureInput",
        "inputDescription": "Pressure readings from a motor"
    }
}
```

자세한 내용은 AWS IoT Events API 참조의 CreateInput을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateInput을 참조하세요.

delete-detector-model

다음 코드 예시에서는 `delete-detector-model`의 사용 방법을 보여줍니다.

AWS CLI

탐지기 모델 삭제

다음 `delete-detector-model` 예시에서는 지정된 탐지기 모델을 삭제합니다. 탐지기 모델의 모든 활성 인스턴스도 삭제됩니다.

```
aws iotevents delete-detector-model \
    --detector-model-name motorDetectorModel
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT Events API 참조의 DeleteDetectorModel을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteDetectorModel을 참조하세요.

delete-input

다음 코드 예시에서는 `delete-input`의 사용 방법을 보여줍니다.

AWS CLI

입력 삭제

다음 `delete-input` 예시에서는 지정된 입력을 삭제합니다.

```
aws iotevents delete-input \
    --input-name PressureInput
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT Events API 참조의 DeleteInput을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteInput을 참조하세요.

`describe-detector-model`

다음 코드 예시에서는 `describe-detector-model`의 사용 방법을 보여줍니다.

AWS CLI

탐지기 모델 정보 가져오기

다음 `describe-detector-model` 예시에서는 지정된 탐지기 모델의 세부 정보를 표시합니다. `version` 파라미터가 지정되지 않았으므로 최신 버전 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 AWS IoT Events API 참조의 DescribeDetectorModel을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDetectorModel을 참조하세요.

`describe-detector`

다음 코드 예시에서는 `describe-detector`의 사용 방법을 보여줍니다.

AWS CLI

탐지기(인스턴스) 정보 가져오기

다음 `describe-detector` 예시에서는 지정된 탐지기(인스턴스)의 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 AWS IoT Events API 참조의 DescribeDetector를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDetector를 참조하세요.

`describe-input`

다음 코드 예시에서는 `describe-input`의 사용 방법을 보여줍니다.

AWS CLI

입력 정보 가져오기

다음 `describe-input` 예시에서는 지정된 입력의 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 AWS IoT Events API 참조의 DescribeInput을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeInput을 참조하세요.

`describe-logging-options`

다음 코드 예시에서는 `describe-logging-options`의 사용 방법을 보여줍니다.

AWS CLI

로깅 설정 정보 가져오기

다음 `describe-logging-options` 예시에서는 현재 AWS IoT Events 로깅 옵션의 현재 설정을 가져옵니다.

```
aws iotevents describe-logging-options
```

출력:

```
{
    "loggingOptions": {
        "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",
        "enabled": false,
        "level": "ERROR"
    }
}
```

자세한 내용은 AWS IoT Events API 참조의 DescribeLoggingOptions를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeLoggingOptions를 참조하세요.

list-detector-model-versions

다음 코드 예시에서는 `list-detector-model-versions`의 사용 방법을 보여줍니다.

AWS CLI

탐지기 모델 버전 정보 가져오기

다음 `list-detector-model-versions` 예시에서는 탐지기 모델의 모든 버전을 나열합니다. 각 탐지기 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 API 참조의 ListDetectorModelVersions를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListDetectorModelVersions를 참조하세요.

list-detector-models

다음 코드 예시에서는 `list-detector-models`의 사용 방법을 보여줍니다.

AWS CLI

탐지기 모델 목록 가져오기

다음 `list-detector-models` 예시에서는 생성한 탐지기 모델을 나열합니다. 각 탐지기 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 API 참조의 ListDetectorModels를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListDetectorModels를 참조하세요.

list-detectors

다음 코드 예시에서는 `list-detectors`의 사용 방법을 보여줍니다.

AWS CLI

탐지기 모델의 탐지기 목록 가져오기

다음 `list-detectors` 예시에서는 계정의 탐지기(탐지기 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 API 참조의 ListDetectors를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListDetectors를 참조하세요.

list-inputs

다음 코드 예시에서는 `list-inputs`의 사용 방법을 보여줍니다.

AWS CLI

입력 나열

다음 `list-inputs` 예시에서는 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 AWS IoT Events API 참조의 ListInputs를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListInputs를 참조하세요.

list-tags-for-resource

다음 코드 예시에서는 `list-tags-for-resource`의 사용 방법을 보여줍니다.

AWS CLI

리소스에 할당된 태그 나열

다음 `list-tags-for-resource` 예시에서는 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 AWS IoT Events API 참조의 ListTagsForResource를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource를 참조하세요.

put-logging-options

다음 코드 예시에서는 put-logging-options의 사용 방법을 보여줍니다.

AWS CLI

로깅 옵션 설정

다음 put-logging-options 예시에서는 AWS IoT Events의 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 AWS IoT Events API 참조의 PutLoggingOptions를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutLoggingOptions를 참조하세요.

tag-resource

다음 코드 예시에서는 tag-resource의 사용 방법을 보여줍니다.

AWS CLI

리소스에 태그 추가

다음 tag-resource 예시에서는 지정된 리소스에 연결된 태그를 추가하거나 수정합니다(키 deviceType이 이미 있는 경우).

```
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 API 참조의 TagResource를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

untag-resource

다음 코드 예시에서는 untag-resource 코드를 사용하는 방법을 보여줍니다.

AWS CLI

리소스에서 태그 제거

다음 untag-resource 예시에서는 지정된 리소스에서 지정된 키 이름이 있는 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 AWS IoT Events API 참조의 UntagResource를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

update-detector-model

다음 코드 예시에서는 update-detector-model의 사용 방법을 보여줍니다.

AWS CLI

탐지기 모델 업데이트

다음 update-detector-model 예시에서는 지정된 탐지기 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 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 AWS IoT Events API 참조의 UpdateDetectorModel을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateDetectorModel을 참조하세요.

update-input

다음 코드 예시에서는 update-input의 사용 방법을 보여줍니다.

AWS CLI

입력 업데이트

다음 update-input 예시에서는 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 AWS IoT Events API 참조의 UpdateInput을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateInput을 참조하세요.

AWS CLI를 사용한 AWS IoT Events-Data 예시

다음 코드 예시에서는 AWS IoT Events-Data에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

작업

batch-put-message

다음 코드 예시에서는 `batch-put-message`의 사용 방법을 보여줍니다.

AWS CLI

AWS IoT Events로 메시지(입력) 전송

다음 `batch-put-message` 예시에서는 AWS IoT Events 시스템으로 메시지 세트를 전송합니다. 각 메시지 페이로드는 지정한 입력(`inputName`)으로 변환되고 해당 입력을 모니터링하는 모든 탐지기로 수집됩니다. 메시지가 여러 개 전송되는 경우 메시지가 처리되는 순서가 보장되지 않습니다. 순서를 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*의 BatchPutMessage를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 BatchPutMessage를 참조하세요.

batch-update-detector

다음 코드 예시에서는 batch-update-detector의 사용 방법을 보여줍니다.

AWS CLI

탐지기(인스턴스) 업데이트

다음 batch-update-detector 예시에서는 지정된 탐지기 모델의 하나 이상의 탐지기(인스턴스)의 상태, 변숫값 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*의 BatchUpdateDetector를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 BatchUpdateDetector를 참조하세요.

create-detector-model

다음 코드 예시에서는 create-detector-model의 사용 방법을 보여줍니다.

AWS CLI

탐지기 모델 생성

다음 create-detector-model 예시에서는 탐지기 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 AWS IoT Events 개발자 안내서*의 CreateDetectorModel을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateDetectorModel을 참조하세요.

create-input

다음 코드 예시에서는 create-input의 사용 방법을 보여줍니다.

AWS CLI

입력 생성

다음 create-input 예시에서는 입력을 생성합니다.

```
aws iotevents create-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": 1560795312.542,
        "creationTime": 1560795312.542,
        "inputName": "PressureInput",
        "inputDescription": "Pressure readings from a motor"
```

```
    }
}
```

자세한 내용은 AWS IoT Events 개발자 안내서*의 CreateInput을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateInput을 참조하세요.

delete-detector-model

다음 코드 예시에서는 `delete-detector-model`의 사용 방법을 보여줍니다.

AWS CLI

탐지기 모델 삭제

다음 `delete-detector-model` 예시에서는 탐지기 모델을 삭제합니다. 탐지기 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*의 DeleteDetectorModel을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteDetectorModel을 참조하세요.

delete-input

다음 코드 예시에서는 `delete-input`의 사용 방법을 보여줍니다.

AWS CLI

입력 삭제

다음 `delete-input` 예시에서는 입력을 삭제합니다.

```
aws iotevents delete-input \
    --input-name PressureInput
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT Events 개발자 안내서*의 DeleteInput을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteInput을 참조하세요.

describe-detector-model

다음 코드 예시에서는 describe-detector-model의 사용 방법을 보여줍니다.

AWS CLI

탐지기 모델 정보 가져오기

다음 describe-detector-model 예시에서는 탐지기 모델을 설명합니다. version 파라미터가 지정되지 않은 경우 명령은 최신 버전의 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*의 DescribeDetectorModel을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDetectorModel을 참조하세요.

describe-detector

다음 코드 예시에서는 describe-detector의 사용 방법을 보여줍니다.

AWS CLI

탐지기(인스턴스) 정보 가져오기

다음 describe-detector 예시에서는 지정된 탐지기(인스턴스)의 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*의 DescribeDetector를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDetector를 참조하세요.

describe-input

다음 코드 예시에서는 describe-input의 사용 방법을 보여줍니다.

AWS CLI

입력 정보 가져오기

다음 describe-input 예시에서는 입력 세부 정보를 가져옵니다.

```
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*의 DescribeInput을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeInput을 참조하세요.

describe-logging-options

다음 코드 예시에서는 describe-logging-options의 사용 방법을 보여줍니다.

AWS CLI

로깅 설정 정보 가져오기

다음 describe-logging-options 예시에서는 현재 AWS IoT Events의 로깅 옵션을 가져옵니다.

```
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*의 DescribeLoggingOptions를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeLoggingOptions를 참조하세요.

list-detector-model-versions

다음 코드 예시에서는 `list-detector-model-versions`의 사용 방법을 보여줍니다.

AWS CLI

탐지기 모델 버전 정보 가져오기

다음 `list-detector-model-versions` 예시에서는 탐지기 모델의 모든 버전을 나열합니다. 각 탐지기 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*의 ListDetectorModelVersions를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListDetectorModelVersions를 참조하세요.

list-detector-models

다음 코드 예시에서는 `list-detector-models`의 사용 방법을 보여줍니다.

AWS CLI

탐지기 모델 목록 가져오기

다음 `list-detector-models` 예시에서는 생성한 탐지기 모델을 나열합니다. 각 탐지기 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*의 ListDetectorModels를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListDetectorModels를 참조하세요.

list-detectors

다음 코드 예시에서는 `list-detectors`의 사용 방법을 보여줍니다.

AWS CLI

탐지기 모델의 탐지기 목록 가져오기

다음 `list-detectors` 예시에서는 탐지기(탐지기 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*의 ListDetectors를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListDetectors를 참조하세요.

`list-inputs`

다음 코드 예시에서는 `list-inputs`의 사용 방법을 보여줍니다.

AWS CLI

입력 나열

다음 `list-inputs` 예시에서는 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*의 ListInputs를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListInputs를 참조하세요.

`list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`의 사용 방법을 보여줍니다.

AWS CLI

리소스에 할당된 태그 나열

다음 `list-tags-for-resource` 예시에서는 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*의 ListTagsForResource를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource를 참조하세요.

`put-logging-options`

다음 코드 예시에서는 `put-logging-options`의 사용 방법을 보여줍니다.

AWS CLI

로깅 옵션 설정

다음 `list-tags-for-resource` 예시에서는 AWS IoT Events의 로깅 옵션을 설정하거나 업데이트합니다. `loggingOptions` 필드 값을 업데이트한 경우 변경 사항이 적용되기까지 최대 1분이 소요될 수 있습니다. 또한 `roleArn` 필드에서 지정한 역할에 연결된 정책을 변경하는 경우(예: 잘못된 정책 수정), 변경 사항이 적용되기까지 최대 5분이 소요될 수 있습니다.

```
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*의 PutLoggingOptions를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutLoggingOptions를 참조하세요.

`tag-resource`

다음 코드 예시에서는 `tag-resource`의 사용 방법을 보여줍니다.

AWS CLI

리소스에 태그 추가

다음 `tag-resource` 예시에서는 지정된 리소스의 태그를 추가하거나 수정합니다. 태그는 리소스 관리에 사용할 수 있는 메타데이터입니다.

```
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 AWS IoT Events 개발자 안내서*의 TagResource를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

untag-resource

다음 코드 예시에서는 `untag-resource` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

리소스에서 태그 제거

다음 `untag-resource` 예시에서는 리소스에서 지정된 태그를 제거합니다.

```
aws iotevents untag-resource \
    --cli-input-json file://pressureInput.untag.json
```

`pressureInput.untag.json`의 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*의 UntagResource를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

update-detector-model

다음 코드 예시에서는 update-detector-model의 사용 방법을 보여줍니다.

AWS CLI

탐지기 모델 업데이트

다음 update-detector-model 예시에서는 탐지기 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 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 AWS IoT Events 개발자 안내서*의 UpdateDetectorModel을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateDetectorModel을 참조하세요.

update-input

다음 코드 예시에서는 update-input의 사용 방법을 보여줍니다.

AWS CLI

입력 업데이트

다음 update-input 예시에서는 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*의 UpdateInput을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateInput을 참조하세요.

AWS CLI를 사용한 AWS IoT Greengrass 예시

다음 코드 예시에서는 AWS IoT Greengrass에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- [작업](#)

작업

associate-role-to-group

다음 코드 예시에서는 `associate-role-to-group`의 사용 방법을 보여줍니다.

AWS CLI

Greengrass 그룹에 역할 연결

다음 `associate-role-to-group` 예시에서는 지정된 IAM 역할을 Greengrass 그룹에 연결합니다. 그룹 역할은 로컬 Lambda 함수 및 커넥터에서 AWS 서비스에 액세스하는 데 사용됩니다. 예를 들어 그룹 역할은 CloudWatch Logs 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 구성](#)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [AssociateRoleToGroup](#)을 참조하세요.

associate-service-role-to-account

다음 코드 예시에서는 `associate-service-role-to-account`의 사용 방법을 보여줍니다.

AWS CLI

AWS 계정에 서비스 역할 연결

다음 `associate-service-role-to-account` 예시는 ARN에서 지정한 IAM 서비스 역할을 AWS 계정의 AWS IoT Greengrass에 연결합니다. 이전에 IAM에서 서비스 역할을 생성했어야 하며, AWS IoT Greengrass가 이 역할을 수임할 수 있도록 정책 문서를 해당 문서에 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.

- API 세부 정보는 AWS CLI 명령 참조의 AssociateServiceRoleToAccount를 참조하세요.

`create-connector-definition-version`

다음 코드 예시에서는 `create-connector-definition-version`의 사용 방법을 보여줍니다.

AWS CLI

커넥터 정의 버전 생성

다음 `create-connector-definition-version` 예시에서는 커넥터 정의 버전을 생성하고 지정된 커넥터 정의에 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [CreateConnectorDefinitionVersion](https://docs.aws.amazon.com/goto/aws-cli/greengrass-2017-06-07/CreateConnectorDefinitionVersion)을 참조하세요.

`create-connector-definition`

다음 코드 예시에서는 `create-connector-definition`의 사용 방법을 보여줍니다.

AWS CLI

커넥터 정의 생성

다음 `create-connector-definition` 예시에서는 커넥터 정의와 초기 커넥터 정의 버전을 생성합니다. 초기 버전에는 커넥터가 하나 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 AWS IoT Greengrass 개발자 안내서의 Greengrass 커넥터 시작하기(CLI)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateConnectorDefinition을 참조하세요.

`create-core-definition-version`

다음 코드 예시에서는 `create-core-definition-version`의 사용 방법을 보여줍니다.

AWS CLI

코어 정의 버전 생성

다음 `create-core-definition-version` 예시에서는 코어 정의 버전을 생성하고 지정된 코어 정의에 연결합니다. 버전에는 하나의 코어만 포함할 수 있습니다. 코어를 생성하기 전에 먼저 해당 AWS IoT 사물을 생성하고 프로비저닝해야 합니다. 이 프로세스에는 `create-core-definition-version` 명령에 필요한 `ThingArn` 및 `CertificateArn`을 반환하는 다음 `iot` 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"
}
```

사물에 대한 퍼블릭 및 프라이빗 키와 코어 디바이스 인증서를 생성합니다. 이 예시에서는 `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 사용자의 정책은 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.

인증서를 사물에 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 버전을 생성합니다.

```
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 AWS IoT Greengrass 코어 구성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateCoreDefinitionVersion을 참조하세요.

create-core-definition

다음 코드 예시에서는 `create-core-definition`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 빈 코어 정의 생성

다음 `create-core-definition` 예시에서는 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"
}
```

예시 2: 초기 버전으로 코어 정의 생성

다음 `create-core-definition` 예시에서는 초기 코어 정의 버전을 포함하는 코어 정의를 생성합니다. 버전에는 하나의 코어만 포함할 수 있습니다. 코어를 생성하기 전에 먼저 해당 AWS IoT 사물을 생성하고 프로비저닝해야 합니다. 이 프로세스에는 `create-core-definition` 명령에 필요한 `ThingArn` 및 `CertificateArn`을 반환하는 다음 `iot` 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"
}
```

사물에 대한 퍼블릭 및 프라이빗 키와 코어 디바이스 인증서를 생성합니다. 이 예시에서는 `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 사용자의 정책은 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.

인증서를 사물에 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 AWS IoT Greengrass 코어 구성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateCoreDefinition을 참조하세요.

`create-deployment`

다음 코드 예시에서는 `create-deployment`의 사용 방법을 보여줍니다.

AWS CLI

Greengrass 그룹의 버전에 대한 배포 생성

다음 `create-deployment` 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateDeployment를 참조하세요.

`create-device-definition-version`

다음 코드 예시에서는 `create-device-definition-version`의 사용 방법을 보여줍니다.

AWS CLI

디바이스 정의 버전 생성

다음 `create-device-definition-version` 예시에서는 디바이스 정의 버전을 생성하고 지정된 디바이스 정의에 연결합니다. 버전은 두 디바이스를 정의합니다. Greengrass 디바이스를 생성하려면 먼저 해당 AWS IoT 사물을 생성하고 프로비저닝해야 합니다. 이 프로세스에는 Greengrass 명령에 필요한 정보를 얻기 위해 실행해야 하는 다음 `iot` 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"
}
```

사물에 대한 퍼블릭 및 프라이빗 키와 디바이스 인증서를 생성합니다. 이 예시에서는 `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 "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 사용자의 정책은 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"
```

인증서를 사물에 연결합니다.

```
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 예시의 처음 두 명령에서 `ThingArn` 및 `CertificateArn`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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateDeviceDefinitionVersion을 참조하세요.

`create-device-definition`

다음 코드 예시에서는 `create-device-definition`의 사용 방법을 보여줍니다.

AWS CLI

작업 정의 생성

다음 `create-device-definition` 예시에서는 초기 디바이스 정의 버전이 포함된 디바이스 정의를 생성합니다. 초기 버전은 두 디바이스를 정의합니다. Greengrass 디바이스를 생성하려면 먼저 해당 AWS IoT 사물을 생성하고 프로비저닝해야 합니다. 이 프로세스에는 Greengrass 명령에 필요한 정보를 얻기 위해 실행해야 하는 다음 `iot` 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"
}
```

사물에 대한 퍼블릭 및 프라이빗 키와 디바이스 인증서를 생성합니다. 이 예시에서는 `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 "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 사용자의 정책은 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"
```

인증서를 사물에 연결합니다.

```
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 예시의 처음 두 명령에서 ThingArn 및 CertificateArn를 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateDeviceDefinition을 참조하세요.

`create-function-definition-version`

다음 코드 예시에서는 `create-function-definition-version`의 사용 방법을 보여줍니다.

AWS CLI

함수 정의의 버전 생성

다음 `create-function-definition-version` 예시에서는 지정된 함수 정의의 새 버전을 생성합니다. 이 버전은 ID가 `Hello-World-function`인 단일 함수를 지정하고, 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateFunctionDefinitionVersion을 참조하세요.

`create-function-definition`

다음 코드 예시에서는 `create-function-definition`의 사용 방법을 보여줍니다.

AWS CLI

Lambda 함수 정의 생성

다음 `create-function-definition` 예시에서는 Lambda 함수 목록(이 경우 `TempMonitorFunction`이라는 함수 하나만 있는 목록)과 해당 구성을 제공하여 Lambda 함수 정의와 초기 버전을 생성합니다. 함수 정의를 생성하려면 먼저 Lambda 함수 ARN이 필요합니다. 함수와 해당 별칭을 생성하려면 Lambda의 `create-function` 및 `publish-version` 명령을 사용합니다. Greengrass 그룹 역할에 권한이 지정되어 있기 때문에 AWS IoT Greengrass가 해당 역할을 사용하지 않더라도 Lambda의 `create-function` 명령에는 실행 역할의 ARN이 필요합니다. IAM `create-role` 명령을 사용해 빈 역할을 생성하여 Lambda의 `create-function`에서 사용할 ARN을 가져오거나 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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateFunctionDefinition을 참조하세요.

`create-group-certificate-authority`

다음 코드 예시에서는 `create-group-certificate-authority`의 사용 방법을 보여줍니다.

AWS CLI

그룹의 인증 기관(CA) 생성

다음 `create-group-certificate-authority` 예시에서는 지정된 그룹의 CA를 생성하거나 교체합니다.

```
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 AWS IoT Greengrass 보안을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateGroupCertificateAuthority를 참조하세요.

create-group-version

다음 코드 예시에서는 create-group-version의 사용 방법을 보여줍니다.

AWS CLI

Greengrass 그룹의 버전 생성

다음 create-group-version 예시에서는 그룹 버전을 생성하고 지정된 그룹에 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 명령을 사용합니다. 초기 버전으로 커넥터 정의를 생성하려면 create-connector-definition 명령을 사용합니다. 초기 버전으로 함수 정의를 생성하려면 create-function-definition 명령을 사용합니다. 초기 버전으로 구독 정의를 생성하려면 create-subscription-definition 명령을 사용합니다. 최신 코어 정의 버전의 ARN을 가져오려면 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 AWS IoT Greengrass 그룹 객체 모델 개요를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateGroupVersion을 참조합니다.

create-group

다음 코드 예시에서는 create-group의 사용 방법을 보여줍니다.

AWS CLI

Greeengrass 그룹 생성

다음 create-group 예시에서는 cli-created-group이라는 그룹을 생성합니다.

```
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 AWS IoT Greengrass 그룹 객체 모델 개요를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateGroup을 참조하세요.

create-logger-definition-version

다음 코드 예시에서는 create-logger-definition-version의 사용 방법을 보여줍니다.

AWS CLI

로거 정의 버전 생성

다음 `create-logger-definition-version` 예시에서는 로거 정의 버전을 생성하고 로거 정의에 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 Logs 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 AWS IoT Greengrass Logs로 모니터링을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateLoggerDefinitionVersion을 참조하세요.

`create-logger-definition`

다음 코드 예시에서는 `create-logger-definition`의 사용 방법을 보여줍니다.

AWS CLI

로거 정의 생성

다음 `create-logger-definition` 예시에서는 초기 로거 정의 버전을 포함하는 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 Logs 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 AWS IoT Greengrass Logs로 모니터링을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateLoggerDefinition을 참조하세요.

create-resource-definition-version

다음 코드 예시에서는 `create-resource-definition-version`의 사용 방법을 보여줍니다.

AWS CLI

리소스 정의의 버전 생성

다음 `create-resource-definition-version` 예시에서는 TwilioAuthToken의 새 버전을 생성합니다.

```
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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateResourceDefinitionVersion을 참조하세요.

`create-resource-definition`

다음 코드 예시에서는 `create-resource-definition`의 사용 방법을 보여줍니다.

AWS CLI

리소스 정의 생성

다음 `create-resource-definition` 예시에서는 Greengrass 그룹에 사용할 리소스 목록을 포함하는 리소스 정의를 생성합니다. 이 예시에서는 리소스 목록을 제공하여 리소스 정의의 초기 버전을 포함합니다. 목록에는 Twilio 권한 부여 토큰에 대한 리소스 하나와 AWS Secrets Manager에 저장된 시크릿에 대한 ARN이 포함되어 있습니다. 리소스 정의를 생성하려면 먼저 시크릿을 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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateResourceDefinition을 참조하세요.

`create-software-update-job`

다음 코드 예시에서는 `create-software-update-job`의 사용 방법을 보여줍니다.

AWS CLI

코어에 대한 업데이트 작업 생성

다음 `create-software-update-job` 예시에서는 `MyFirstGroup_Core`라는 코어에서 AWS IoT Greengrass 코어 소프트웨어를 업데이트하는 무선 업데이트(OTA) 작업을 생성합니다. 이 명령에는 Amazon S3의 소프트웨어 업데이트 패키지에 대한 액세스를 허용하고 신뢰할 수 있는 엔터티로 `iot.amazonaws.com`을 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 AWS IoT Greengrass 개발자 안내서의 AWS IoT Greengrass 코어 소프트웨어 OTA 업데이트를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateSoftwareUpdateJob을 참조하세요.

create-subscription-definition-version

다음 코드 예시에서는 create-subscription-definition-version의 사용 방법을 보여줍니다.

AWS CLI

구독 정의의 새 버전 생성

다음 create-subscription-definition-version 예시에서는 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateSubscriptionDefinitionVersion을 참조하세요.

create-subscription-definition

다음 코드 예시에서는 create-subscription-definition의 사용 방법을 보여줍니다.

AWS CLI

구독 정의 생성

다음 create-subscription-definition 예시에서는 구독 정의를 생성하고 초기 버전을 지정합니다. 초기 버전에는 세 가지 구독이 포함되어 있습니다. 하나는 커넥터가 구독하는 MQTT 주제용이고, 다른 하나는 함수가 AWS IoT에서 온도 판독값을 수신할 수 있도록 허용하기 위한 것이고, 다른 하나는 AWS IoT가 커넥터에서 상태 정보를 수신할 수 있도록 허용하기 위한 것입니다. 이 예시에서는 Lambda의 create-alias 명령을 사용하여 이전에 생성된 Lambda 함수 별칭에 대한 ARN을 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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateSubscriptionDefinition을 참조하세요.

`delete-connector-definition`

다음 코드 예시에서는 `delete-connector-definition`의 사용 방법을 보여줍니다.

AWS CLI

커넥터 정의 삭제

다음 `delete-connector-definition` 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteConnectorDefinition을 참조하세요.

`delete-core-definition`

다음 코드 예시에서는 `delete-core-definition`의 사용 방법을 보여줍니다.

AWS CLI

코어 정의 삭제

다음 `delete-core-definition` 예시에서는 모든 버전을 포함하여 지정된 Greengrass 코어 정의를 삭제합니다. Greengrass 그룹에 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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteCoreDefinition을 참조하세요.

delete-device-definition

다음 코드 예시에서는 delete-device-definition의 사용 방법을 보여줍니다.

AWS CLI

디바이스 정의 삭제

다음 delete-device-definition 예시에서는 모든 버전을 포함하여 지정된 디바이스 정의를 삭제합니다. 그룹 버전에서 사용하는 디바이스 정의 버전을 삭제하면 해당 그룹 버전을 성공적으로 배포할 수 없습니다.

```
aws greengrass delete-device-definition \
    --device-definition-id "f9ba083d-5ad4-4534-9f86-026a45df1ccd"
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteDeviceDefinition을 참조하세요.

delete-function-definition

다음 코드 예시에서는 delete-function-definition의 사용 방법을 보여줍니다.

AWS CLI

함수 정의 삭제

다음 delete-function-definition 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteFunctionDefinition을 참조하세요.

delete-group

다음 코드 예시에서는 delete-group의 사용 방법을 보여줍니다.

AWS CLI

그룹 삭제

다음 delete-group 예시에서는 지정된 Greengrass 그룹을 삭제합니다.

```
aws greengrass delete-group \
    --group-id "4e22bd92-898c-436b-ade5-434d883ff749"
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteGroup을 참조하세요.

delete-logger-definition

다음 코드 예시에서는 delete-logger-definition의 사용 방법을 보여줍니다.

AWS CLI

로거 정의 삭제

다음 delete-logger-definition 예시에서는 모든 로거 정의 버전을 포함하여 지정된 로거 정의를 삭제합니다. 그룹 버전에서 사용되는 로거 정의 버전을 삭제하면 해당 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 AWS IoT Greengrass Logs로 모니터링을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteLoggerDefinition을 참조하세요.

delete-resource-definition

다음 코드 예시에서는 delete-resource-definition의 사용 방법을 보여줍니다.

AWS CLI

리소스 정의 삭제

다음 delete-resource-definition 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteResourceDefinition을 참조하세요.

delete-subscription-definition

다음 코드 예시에서는 delete-subscription-definition의 사용 방법을 보여줍니다.

AWS CLI

구독 정의 삭제

다음 delete-subscription-definition 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteSubscriptionDefinition을 참조하세요.

disassociate-role-from-group

다음 코드 예시에서는 disassociate-role-from-group의 사용 방법을 보여줍니다.

AWS CLI

Greengrass 그룹에서 역할 연결 해제

다음 disassociate-role-from-group 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 DisassociateRoleFromGroup을 참조하세요.

disassociate-service-role-from-account

다음 코드 예시에서는 `disassociate-service-role-from-account`의 사용 방법을 보여줍니다.

AWS CLI

AWS 계정에서 서비스 역할 연결 해제

다음 `disassociate-service-role-from-account` 예시에서는 AWS 계정에 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.

- API 세부 정보는 AWS CLI 명령 참조의 DisassociateServiceRoleFromAccount를 참조하세요.

get-associated-role

다음 코드 예시에서는 `get-associated-role`의 사용 방법을 보여줍니다.

AWS CLI

Greengrass 그룹에 연결된 역할 가져오기

다음 `get-associated-role` 예시에서는 지정된 Greengrass 그룹에 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.

- API 세부 정보는 AWS CLI 명령 참조의 GetAssociatedRole을 참조하세요.

`get-bulk-deployment-status`

다음 코드 예시에서는 `get-bulk-deployment-status`의 사용 방법을 보여줍니다.

AWS CLI

일괄 배포의 상태 확인

다음 `get-bulk-deployment-status` 예시에서는 지정된 일괄 배포 작업의 상태 정보를 가져옵니다. 이 예시에서는 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 일괄 배포 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetBulkDeploymentStatus를 참조하세요.

`get-connectivity-info`

다음 코드 예시에서는 `get-connectivity-info`의 사용 방법을 보여줍니다.

AWS CLI

Greengrass 코어의 연결 정보 가져오기

다음 `get-connectivity-info` 예시에서는 지정된 Greengrass 코어에 디바이스가 연결하는 데 사용할 수 있는 엔드포인트를 표시합니다. 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetConnectivityInfo를 참조하세요.

get-connector-definition-version

다음 코드 예시에서는 get-connector-definition-version의 사용 방법을 보여줍니다.

AWS CLI

커넥터 정의의 특정 버전에 대한 정보 가져오기

다음 get-connector-definition-version 예시에서는 지정된 커넥터 정의의 지정된 버전에 대한 정보를 가져옵니다. 커넥터 정의의 모든 버전의 ID를 가져오려면 list-connector-definition-versions 명령을 사용합니다. 커넥터 정의에 추가된 마지막 버전의 ID를 가져오려면 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.

- API 세부 정보는 AWS CLI 명령 참조의 GetConnectorDefinitionVersion을 참조하세요.

get-connector-definition

다음 코드 예시에서는 get-connector-definition의 사용 방법을 보여줍니다.

AWS CLI

커넥터 정의의 정보 가져오기

다음 get-connector-definition 예시에서는 지정된 커넥터 정의의 정보를 가져옵니다. 커넥터 정의의 ID를 가져오려면 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.

- API 세부 정보는 AWS CLI 명령 참조의 GetConnectorDefinition을 참조하세요.

get-core-definition-version

다음 코드 예시에서는 get-core-definition-version의 사용 방법을 보여줍니다.

AWS CLI

Greengrass 코어 정의의 특정 버전에 대한 세부 정보 가져오기

다음 get-core-definition-version 예시에서는 지정된 코어 정의의 지정된 버전에 대한 정보를 가져옵니다. 코어 정의의 모든 버전의 ID를 가져오려면 list-core-definition-versions 명령을 사용합니다. 코어 정의에 추가된 마지막 버전의 ID를 가져오려면 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetCoreDefinitionVersion을 참조하세요.

get-core-definition

다음 코드 예시에서는 get-core-definition의 사용 방법을 보여줍니다.

AWS CLI

Greengrass 코어 정의의 세부 정보 가져오기

다음 get-core-definition 예시에서는 지정된 코어 정의의 정보를 가져옵니다. 코어 정의의 ID를 가져오려면 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": {}
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetCoreDefinition을 참조하세요.

get-deployment-status

다음 코드 예시에서는 get-deployment-status의 사용 방법을 보여줍니다.

AWS CLI

배포 상태 가져오기

다음 get-deployment-status 예시에서는 지정된 Greengrass 그룹의 지정된 배포 상태를 가져옵니다. 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetDeploymentStatus를 참조하세요.

get-device-definition-version

다음 코드 예시에서는 get-device-definition-version의 사용 방법을 보여줍니다.

AWS CLI

디바이스 정의 버전 가져오기

다음 get-device-definition-version 예시에서는 지정된 디바이스 정의의 지정된 버전에 대한 정보를 가져옵니다. 디바이스 정의의 모든 버전의 ID를 가져오려면 list-device-definition-versions 명령을 사용합니다. 디바이스 정의에 추가된 마지막 버전의 ID를 가져오려면 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetDeviceDefinitionVersion을 참조하세요.

get-device-definition

다음 코드 예시에서는 get-device-definition의 사용 방법을 보여줍니다.

AWS CLI

디바이스 정의 가져오기

다음 get-device-definition 예시에서는 지정된 디바이스 정의의 정보를 가져옵니다. 디바이스 정의의 ID를 가져오려면 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"
}
```


- API 세부 정보는 AWS CLI 명령 참조의 GetDeviceDefinition을 참조하세요.

get-function-definition-version

다음 코드 예시에서는 get-function-definition-version의 사용 방법을 보여줍니다.

AWS CLI

Lambda 함수의 특정 버전에 대한 세부 정보 가져오기

다음 get-function-definition-version 예시에서는 지정된 함수 정의의 지정된 버전에 대한 정보를 가져옵니다. 함수 정의의 모든 버전의 ID를 가져오려면 list-function-definition-versions 명령을 사용합니다. 함수 정의에 추가된 마지막 버전의 ID를 가져오려면 get-function-definition 명령을 사용하여 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetFunctionDefinitionVersion을 참조하세요.

get-function-definition

다음 코드 예시에서는 `get-function-definition`의 사용 방법을 보여줍니다.

AWS CLI

함수 정의 가져오기

다음 get-function-definition 예시에서는 지정된 함수 정의의 세부 정보를 표시합니다. 함수 정의의 ID를 가져오려면 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": {}
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetFunctionDefinition을 참조하세요.

get-group-certificate-authority

다음 코드 예시에서는 get-group-certificate-authority의 사용 방법을 보여줍니다.

AWS CLI

Greengrass 그룹에 연결된 CA 가져오기

다음 get-group-certificate-authority 예시에서는 지정된 Greengrass 그룹에 연결된 인증 기관(CA)을 가져옵니다. 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"
}
```


- API 세부 정보는 AWS CLI 명령 참조의 [GetGroupCertificateAuthority](#)를 참조하세요.

`get-group-certificate-configuration`

다음 코드 예시에서는 `get-group-certificate-configuration`의 사용 방법을 보여줍니다.

AWS CLI

Greengrass 그룹에서 사용하는 인증 기관의 구성 가져오기

다음 `get-group-certificate-configuration` 예시에서는 지정된 Greengrass 그룹에서 사용하는 인증 기관(CA)의 구성을 가져옵니다.

```
aws greengrass get-group-certificate-configuration \
    --group-id "1013db12-8b58-45ff-acc7-704248f66731"
```

출력:

```
{
    "CertificateAuthorityExpiryInMilliseconds": 2524607999000,
    "CertificateExpiryInMilliseconds": 604800000,
    "GroupId": "1013db12-8b58-45ff-acc7-704248f66731"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetGroupCertificateConfiguration을 참조하세요.

get-group-version

다음 코드 예시에서는 get-group-version의 사용 방법을 보여줍니다.

AWS CLI

Greengrass 그룹의 버전 정보 가져오기

다음 get-group-version 예시에서는 지정된 그룹의 지정된 버전에 대한 정보를 가져옵니다. 그룹의 모든 버전의 ID를 가져오려면 list-group-versions 명령을 사용합니다. 그룹에 추가된 마지막 버전의 ID를 가져오려면 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetGroupVersion을 참조하세요.

get-group

다음 코드 예시에서는 get-group의 사용 방법을 보여줍니다.

AWS CLI

Greengrass 그룹의 정보 가져오기

다음 get-group 예시에서는 지정된 Greengrass 그룹의 정보를 가져옵니다. 그룹의 ID를 가져오려면 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": {}
```

```
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [GetGroup](https://docs.aws.amazon.com/goto/aws-cli/greengrass-2017-06-07/GetGroup)을 참조하세요.

get-logger-definition-version

다음 코드 예시에서는 `get-logger-definition-version`의 사용 방법을 보여줍니다.

AWS CLI

로거 정의의 버전 정보 가져오기

다음 `get-logger-definition-version` 예시에서는 지정된 로거 정의의 지정된 버전에 대한 정보를 가져옵니다. 로거 정의의 모든 버전의 ID를 가져오려면 `list-logger-definition-versions` 명령을 사용합니다. 로거 정의에 추가된 마지막 버전의 ID를 가져오려면 `get-logger-definition` 명령을 사용하여 `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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [GetLoggerDefinitionVersion](https://docs.aws.amazon.com/goto/aws-cli/greengrass-2017-06-07/GetLoggerDefinitionVersion)을 참조하세요.

get-logger-definition

다음 코드 예시에서는 `get-logger-definition`의 사용 방법을 보여줍니다.

AWS CLI

로거 정의의 정보 가져오기

다음 `get-logger-definition` 예시에서는 지정된 로거 정의의 정보를 가져옵니다. 로거 정의의 ID를 가져오려면 `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": {}
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetLoggerDefinition을 참조하세요.

`get-resource-definition-version`

다음 코드 예시에서는 `get-resource-definition-version`의 사용 방법을 보여줍니다.

AWS CLI

리소스 정의의 특정 버전 정보 가져오기

다음 `get-resource-definition-version` 예시에서는 지정된 리소스 정의의 지정된 버전에 대한 정보를 가져옵니다. 리소스 정의의 모든 버전의 ID를 가져오려면 `list-resource-definition-versions` 명령을 사용합니다. 리소스 정의에 추가된 마지막 버전의 ID를 가져오려면 `get-resource-definition` 명령을 사용하여 `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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetResourceDefinitionVersion을 참조하세요.

get-resource-definition

다음 코드 예시에서는 get-resource-definition의 사용 방법을 보여줍니다.

AWS CLI

리소스 정의의 정보 가져오기

다음 get-resource-definition 예시에서는 지정된 리소스 정의의 정보를 가져옵니다. 리소스 정의의 ID를 가져오려면 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": {}
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetResourceDefinition을 참조하세요.

`get-service-role-for-account`

다음 코드 예시에서는 `get-service-role-for-account`의 사용 방법을 보여줍니다.

AWS CLI

계정에 연결된 서비스 역할의 세부 정보 가져오기

다음 `get-service-role-for-account` 예시에서는 AWS 계정에 연결된 서비스 역할의 정보를 가져옵니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 GetServiceRoleForAccount를 참조하세요.

get-subscription-definition-version

다음 코드 예시에서는 get-subscription-definition-version의 사용 방법을 보여줍니다.

AWS CLI

구독 정의의 특정 버전 정보 가져오기

다음 get-subscription-definition-version 예시에서는 지정된 구독 정의의 지정된 버전에 대한 정보를 가져옵니다. 구독 정의의 모든 버전의 ID를 가져오려면 list-subscription-definition-versions 명령을 사용합니다. 구독 정의에 추가된 마지막 버전의 ID를 가져오려면 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetSubscriptionDefinitionVersion을 참조하세요.

get-subscription-definition

다음 코드 예시에서는 get-subscription-definition의 사용 방법을 보여줍니다.

AWS CLI

구독 정의의 정보 가져오기

다음 get-subscription-definition 예시에서는 지정된 구독 정의의 정보를 가져옵니다. 구독 정의의 ID를 가져오려면 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": {}
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetSubscriptionDefinition을 참조하세요.

get-thing-runtime-configuration

다음 코드 예시에서는 get-thing-runtime-configuration의 사용 방법을 보여줍니다.

AWS CLI

Greengrass 코어의 런타임 구성 가져오기

다음 get-thing-runtime-configuration 예시에서는 Greengrass 코어의 런타임 구성을 가져옵니다. 런타임 구성을 가져오려면 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.

- API 세부 정보는 AWS CLI 명령 참조의 GetThingRuntimeConfiguration을 참조하세요.

`list-bulk-deployment-detailed-reports`

다음 코드 예시에서는 `list-bulk-deployment-detailed-reports`의 사용 방법을 보여줍니다.

AWS CLI

일괄 배포의 개별 배포 정보 나열

다음 `list-bulk-deployment-detailed-reports` 예시에서는 상태를 포함하여 일괄 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 일괄 배포 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListBulkDeploymentDetailedReports를 참조하세요.

`list-bulk-deployments`

다음 코드 예시에서는 `list-bulk-deployments`의 사용 방법을 보여줍니다.

AWS CLI

일괄 배포 나열

다음 `list-bulk-deployments` 예시에서는 모든 일괄 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 일괄 배포 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListBulkDeployments를 참조하세요.

`list-connector-definition-versions`

다음 코드 예시에서는 `list-connector-definition-versions`의 사용 방법을 보여줍니다.

AWS CLI

커넥터 정의에 사용할 수 있는 버전 나열

다음 `list-connector-definition-versions` 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListConnectorDefinitionVersions를 참조하세요.

list-connector-definitions

다음 코드 예시에서는 `list-connector-definitions`의 사용 방법을 보여줍니다.

AWS CLI

정의된 Greengrass 커넥터 나열

다음 `list-connector-definitions` 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 [ListConnectorDefinitions](#)를 참조하세요.

`list-core-definition-versions`

다음 코드 예시에서는 `list-core-definition-versions`의 사용 방법을 보여줍니다.

AWS CLI

Greengrass 코어 정의의 버전 나열

다음 `list-core-definitions` 예시에서는 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [ListCoreDefinitionVersions](#)를 참조하세요.

`list-core-definitions`

다음 코드 예시에서는 `list-core-definitions`의 사용 방법을 보여줍니다.

AWS CLI

Greengrass 코어 정의 나열

다음 `list-core-definitions` 예시에서는 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListCoreDefinitions를 참조하세요.

`list-deployments`

다음 코드 예시에서는 `list-deployments`의 사용 방법을 보여줍니다.

AWS CLI

Greengrass 그룹의 배포 나열

다음 `list-deployments` 예시에서는 지정된 Greengrass 그룹의 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"
        }
```

```
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListDeployments를 참조하세요.

list-device-definition-versions

다음 코드 예시에서는 `list-device-definition-versions`의 사용 방법을 보여줍니다.

AWS CLI

디바이스 정의의 버전 나열

다음 `list-device-definition-versions` 예시에서는 지정된 디바이스 정의에 연결된 디바이스 정의 버전을 표시합니다.

```
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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListDeviceDefinitionVersions를 참조하세요.

list-device-definitions

다음 코드 예시에서는 list-device-definitions의 사용 방법을 보여줍니다.

AWS CLI

디바이스 정의 나열

다음 list-device-definitions 예시에서는 지정된 AWS 리전의 AWS 계정에 있는 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListDeviceDefinitions를 참조하세요.

list-function-definition-versions

다음 코드 예시에서는 `list-function-definition-versions`의 사용 방법을 보여줍니다.

AWS CLI

Lambda 함수 버전 나열

다음 `list-function-definition-versions` 예시에서는 지정된 Lambda 함수의 모든 버전 목록을 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListFunctionDefinitionVersions를 참조하세요.

list-function-definitions

다음 코드 예시에서는 `list-function-definitions`의 사용 방법을 보여줍니다.

AWS CLI

Lambda 함수 나열

다음 `list-function-definitions` 예시에서는 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListFunctionDefinitions를 참조하세요.

list-group-certificate-authorities

다음 코드 예시에서는 `list-group-certificate-authorities`의 사용 방법을 보여줍니다.

AWS CLI

그룹의 현재 CA 나열

다음 `list-group-certificate-authorities` 예시에서는 지정된 Greengrass 그룹의 현재 인증 기관(CA)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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListGroupCertificateAuthorities를 참조하세요.

list-group-versions

다음 코드 예시에서는 list-group-versions의 사용 방법을 보여줍니다.

AWS CLI

Greengrass 그룹의 버전 나열

다음 list-group-versions 예시에서는 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListGroupVersions를 참조하세요.

list-groups

다음 코드 예시에서는 list-groups의 사용 방법을 보여줍니다.

AWS CLI

Greengrass 그룹 나열

다음 list-groups 예시에서는 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListGroups를 참조하세요.

`list-logger-definition-versions`

다음 코드 예시에서는 `list-logger-definition-versions`의 사용 방법을 보여줍니다.

AWS CLI

로거 정의의 버전 목록 가져오기

다음 `list-logger-definition-versions` 예시에서는 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListLoggerDefinitionVersions를 참조하세요.

`list-logger-definitions`

다음 코드 예시에서는 `list-logger-definitions`의 사용 방법을 보여줍니다.

AWS CLI

로거 정의 목록 가져오기

다음 `list-logger-definitions` 예시에서는 AWS 계정의 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListLoggerDefinitions를 참조하세요.

`list-resource-definition-versions`

다음 코드 예시에서는 `list-resource-definition-versions`의 사용 방법을 보여줍니다.

AWS CLI

리소스 정의의 버전 나열

다음 `list-resource-definition-versions` 예시에서는 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListResourceDefinitionVersions를 참조하세요.

list-resource-definitions

다음 코드 예시에서는 `list-resource-definitions`의 사용 방법을 보여줍니다.

AWS CLI

정의된 리소스 나열

다음 `list-resource-definitions` 예시에서는 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListResourceDefinitions를 참조하세요.

list-subscription-definition-versions

다음 코드 예시에서는 `list-subscription-definition-versions`의 사용 방법을 보여줍니다.

AWS CLI

구독 정의의 버전 나열

다음 `list-subscription-definition-versions` 예시에서는 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListSubscriptionDefinitionVersions를 참조하세요.

`list-subscription-definitions`

다음 코드 예시에서는 `list-subscription-definitions`의 사용 방법을 보여줍니다.

AWS CLI

구독 정의 목록 가져오기

다음 `list-subscription-definitions` 예시에서는 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListSubscriptionDefinitions를 참조하세요.

`list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`의 사용 방법을 보여줍니다.

AWS CLI

리소스에 연결된 태그 나열

다음 `list-tags-for-resource` 예시에서는 지정된 리소스에 연결된 태그와 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 리소스에 태그 지정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource를 참조하세요.

`reset-deployments`

다음 코드 예시에서는 `reset-deployments`의 사용 방법을 보여줍니다.

AWS CLI

Greengrass 그룹의 배포 정보 정리

다음 `reset-deployments` 예시에서는 지정된 Greengrass 그룹의 배포 정보를 정리합니다. `--force option`을 추가하면 코어 디바이스가 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.

- API 세부 정보는 AWS CLI 명령 참조의 ResetDeployments를 참조하세요.

start-bulk-deployment

다음 코드 예시에서는 `start-bulk-deployment`의 사용 방법을 보여줍니다.

AWS CLI

일괄 배포 작업 시작

다음 `start-bulk-deployment` 예시에서는 S3 버킷에 저장된 파일을 사용해 배포할 그룹을 지정하여 일괄 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 일괄 배포 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartBulkDeployment를 참조하세요.

stop-bulk-deployment

다음 코드 예시에서는 `stop-bulk-deployment`의 사용 방법을 보여줍니다.

AWS CLI

일괄 배포 중지

다음 `stop-bulk-deployment` 예시에서는 지정된 일괄 배포를 중지합니다. 완료된 일괄 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 일괄 배포 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StopBulkDeployment를 참조하세요.

tag-resource

다음 코드 예시에서는 `tag-resource`의 사용 방법을 보여줍니다.

AWS CLI

리소스에 태그 적용

다음 `tag-resource` 예시에서는 지정된 Greengrass 리소스에 `ResourceType` 및 `ResourceSubType`라는 2개의 태그를 적용합니다. 이 작업은 새 태그와 값을 추가하거나 기존 태그의 값을 업데이트할 수 있습니다. 태그를 제거하려면 `untag-resource` 명령을 사용합니다.

```
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 리소스에 태그 지정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

untag-resource

다음 코드 예시에서는 `untag-resource`의 사용 방법을 보여줍니다.

AWS CLI

리소스에서 태그 및 값 제거

다음 `untag-resource` 예시에서는 키가 `Category`인 태그를 지정된 Greengrass 그룹에서 제거합니다. 지정된 리소스에 대한 키 `Category`가 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 리소스에 태그 지정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

`update-connectivity-info`

다음 코드 예시에서는 `update-connectivity-info`의 사용 방법을 보여줍니다.

AWS CLI

Greengrass 코어의 연결 정보 업데이트

다음 `update-connectivity-info` 예시에서는 지정된 Greengrass 코어에 디바이스가 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"
```

```
}
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateConnectivityInfo를 참조하세요.

update-connector-definition

다음 코드 예시에서는 update-connector-definition의 사용 방법을 보여줍니다.

AWS CLI

커넥터 정의의 이름 업데이트

다음 update-connector-definition 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateConnectorDefinition을 참조하세요.

update-core-definition

다음 코드 예시에서는 update-core-definition의 사용 방법을 보여줍니다.

AWS CLI

코어 정의 업데이트

다음 update-core-definition 예시에서는 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 AWS IoT Greengrass 코어 구성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateCoreDefinition을 참조하세요.

update-device-definition

다음 코드 예시에서는 update-device-definition의 사용 방법을 보여줍니다.

AWS CLI

디바이스 정의 업데이트

다음 update-device-definition 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateDeviceDefinition을 참조하세요.

update-function-definition

다음 코드 예시에서는 update-function-definition의 사용 방법을 보여줍니다.

AWS CLI

함수 정의의 이름 업데이트

다음 update-function-definition 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateFunctionDefinition을 참조하세요.

update-group-certificate-configuration

다음 코드 예시에서는 update-group-certificate-configuration의 사용 방법을 보여줍니다.

AWS CLI

그룹 인증서의 만료 업데이트

다음 update-group-certificate-configuration 예시에서는 지정된 그룹에 대해 생성된 인증서에 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 AWS IoT Greengrass 보안을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateGroupCertificateConfiguration을 참조하세요.

update-group

다음 코드 예시에서는 update-group의 사용 방법을 보여줍니다.

AWS CLI

그룹 이름 업데이트

다음 update-group 예시에서는 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 AWS IoT Greengrass 구성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateGroup을 참조하세요.

update-logger-definition

다음 코드 예시에서는 update-logger-definition의 사용 방법을 보여줍니다.

AWS CLI

로거 정의 업데이트

다음 update-logger-definition 예시에서는 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 AWS IoT Greengrass Logs로 모니터링을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateLoggerDefinition을 참조하세요.

update-resource-definition

다음 코드 예시에서는 update-resource-definition의 사용 방법을 보여줍니다.

AWS CLI

리소스 정의의 이름 업데이트

다음 update-resource-definition 예시에서는 지정된 리소스 정의의 이름을 업데이트합니다. 리소스의 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 사용하여 로컬 리소스에 액세스를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateResourceDefinition을 참조하세요.

update-subscription-definition

다음 코드 예시에서는 `update-subscription-definition`의 사용 방법을 보여줍니다.

AWS CLI

구독 정의의 이름 업데이트

다음 `update-subscription-definition` 예시에서는 지정된 구독 정의의 이름을 업데이트합니다. 구독의 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 안내서의 제목을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateSubscriptionDefinition을 참조하세요.

update-thing-runtime-configuration

다음 코드 예시에서는 `update-thing-runtime-configuration`의 사용 방법을 보여줍니다.

AWS CLI

Greengrass 코어의 런타임 구성에서 원격 측정 켜기

다음 `update-thing-runtime-configuration` 예시에서는 Greengrass 코어의 런타임 구성을 업데이트하여 원격 측정을 켭니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateThingRuntimeConfiguration을 참조하세요.

AWS CLI를 사용한 AWS IoT Greengrass V2 예시

다음 코드 예시에서는 AWS IoT Greengrass V2에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`associate-service-role-to-account`

다음 코드 예시에서는 `associate-service-role-to-account` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS 계정에 Greengrass 서비스 역할 연결

다음 `associate-service-role-to-account` 예시에서는 AWS 계정에 대한 서비스 역할을 AWS IoT Greengrass에 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 Greengrass V2 개발자 안내서의 Greengrass 서비스 역할을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AssociateServiceRoleToAccount 섹션을 참조하세요.

batch-associate-client-device-with-core-device

다음 코드 예시에서는 batch-associate-client-device-with-core-device 코드를 사용하는 방법을 보여줍니다.

AWS CLI

클라이언트 디바이스를 코어 디바이스와 연결

다음 batch-associate-client-device-with-core-device 예시에서는 두 개의 클라이언트 디바이스를 하나의 코어 디바이스에 연결합니다.

```
aws greengrassv2 batch-associate-client-device-with-core-device \
  --core-device-thing-name MyGreengrassCore \
  --entries thingName=MyClientDevice1 thingName=MyClientDevice2
```

출력:

```
{
    "errorEntries": []
}
```

자세한 내용은 AWS IoT Greengrass V2 개발자 안내서의 로컬 IoT 디바이스와 상호 작용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 BatchAssociateClientDeviceWithCoreDevice 섹션을 참조하세요.

batch-disassociate-client-device-from-core-device

다음 코드 예시에서는 batch-disassociate-client-device-from-core-device 코드를 사용하는 방법을 보여줍니다.

AWS CLI

코어 디바이스에서 클라이언트 디바이스의 연결 해제

다음 batch-disassociate-client-device-from-core-device 예시에서는 코어 디바이스에서 두 개의 클라이언트 디바이스를 연결 해제합니다.

```
aws greengrassv2 batch-disassociate-client-device-from-core-device \
  --core-device-thing-name MyGreengrassCore \
  --entries thingName=MyClientDevice1 thingName=MyClientDevice2
```

출력:

```
{
    "errorEntries": []
}
```

자세한 내용은 AWS IoT Greengrass V2 개발자 안내서의 로컬 IoT 디바이스와 상호 작용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 BatchDisassociateClientDeviceFromCoreDevice 섹션을 참조하세요.

cancel-deployment

다음 코드 예시에서는 cancel-deployment 코드를 사용하는 방법을 보여줍니다.

AWS CLI

배포 취소

다음 cancel-deployment 예시에서는 사물 그룹에 대한 연속 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.

- API 세부 정보는 AWS CLI 명령 참조의 CancelDeployment 섹션을 참조하세요.

`create-component-version`

다음 코드 예시에서는 `create-component-version` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

예시 1: 레시피에서 구성 요소 버전 생성

다음 `create-component-version` 예시에서는 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 사용자 지정 구성 요소 생성 및 배포할 구성 요소 업로드를 참조하세요.

예시 2: AWS Lambda 함수에서 구성 요소 버전 생성

다음 `create-component-version` 예시는 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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateComponentVersion 섹션을 참조하세요.

`create-deployment`

다음 코드 예시에서는 `create-deployment` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

예시 1: 배포 생성

다음 `create-deployment` 예시에서는 AWS IoT Greengrass 명령줄 인터페이스를 핵심 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.

예시 2: 구성 요소 구성을 업데이트하는 배포 생성

다음 `create-deployment` 예시에서는 핵심 디바이스 그룹에 AWS IoT Greengrass 핵 구성 요소를 배포합니다. 이 배포는 핵 구성 요소에 대해 다음 구성 업데이트를 적용합니다.

대상 디바이스의 프록시 설정을 기본값인 프록시 없음 설정으로 재설정합니다. 대상 디바이스의 MQTT 설정을 기본값으로 재설정합니다. 핵의 JVM 옵션을 설정합니다.핵의 로깅 수준을 설정합니다.

```
aws greengrassv2 create-deployment \
    --cli-input-json file://nucleus-deployment.json
```

nucleus-deployment.json의 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 업로드를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateDeployment를 참조하세요.

delete-component

다음 코드 예시에서는 `delete-component` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

구성 요소 버전 삭제

다음 `delete-component` 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteComponent 섹션을 참조하세요.

delete-core-device

다음 코드 예시에서는 `delete-core-device` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

디바이스 삭제

다음 `delete-core-device` 예시에서는 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 AWS IoT Greengrass 코어 소프트웨어 제거를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteCoreDevice 섹션을 참조하세요.

describe-component

다음 코드 예시에서는 describe-component 코드를 사용하는 방법을 보여줍니다.

AWS CLI

구성 요소 버전 설명

다음 describe-component 예시에서는 Hello World 구성 요소를 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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeComponent 섹션을 참조하세요.

`disassociate-service-role-from-account`

다음 코드 예시에서는 `disassociate-service-role-from-account` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS 계정에서 Greengrass 서비스 역할의 연결 해제

다음 `disassociate-service-role-from-account` 예시에서는 AWS 계정에 대한 AWS IoT Greengrass에서 Greengrass 서비스 역할을 연결 해제하는 예시입니다.

```
aws greengrassv2 disassociate-service-role-from-account
```

출력:

```
{
    "disassociatedAt": "2022-01-19T19:26:09Z"
}
```

자세한 내용은 AWS IoT Greengrass V2 개발자 안내서의 Greengrass 서비스 역할을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisassociateServiceRoleFromAccount 섹션을 참조하세요.

`get-component-version-artifact`

다음 코드 예시에서는 `get-component-version-artifact` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

구성 요소 아티팩트를 다운로드할 URL 가져오기

다음 `get-component-version-artifact` 예시에서는 로컬 디버그 콘솔 구성 요소의 JAR 파일을 다운로드할 수 있는 URL을 가져옵니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 GetComponentVersionArtifact 섹션을 참조하세요.

`get-component`

다음 코드 예시에서는 `get-component` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

예시 1: YAML 형식(Linux, macOS 또는 Unix)으로 구성 요소의 레시피 다운로드

다음 `get-component` 예시는 Hello World 구성 요소의 레시피를 YAML 형식의 파일로 다운로드하는 예시입니다. 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 AWS 명령줄 인터페이스 사용 설명서의 명령어 출력 제어를 참조하세요. `base64` 유틸리티를 사용합니다. 이 유틸리티는 추출된 블롭을 원본 텍스트로 디코딩합니다. `get-component` 명령이 성공하면 반환되는 블롭은 base64로 인코딩된 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.

예시 2: 구성 요소의 레시피를 YAML 형식으로 다운로드(Windows CMD)

다음 `get-component` 예시는 Hello World 구성 요소의 레시피를 YAML 형식의 파일로 다운로드하는 예시입니다. 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.

예시 3: 구성 요소의 레시피를 YAML 형식으로 다운로드하는 방법(Windows PowerShell)

다음 `get-component` 예시는 Hello World 구성 요소의 레시피를 YAML 형식의 파일로 다운로드하는 예시입니다. 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.

- API 세부 정보는 AWS CLI 명령 참조의 GetComponent 섹션을 참조하세요.

get-connectivity-info

다음 코드 예시에서는 get-connectivity-info 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Greengrass 코어 디바이스의 연결 정보를 가져오는 방법

다음 get-connectivity-info 예시는 Greengrass 코어 디바이스에 대한 연결 정보를 가져오는 예시입니다. 클라이언트 디바이스는 이 정보를 사용하여 이 코어 디바이스에서 실행되는 MQTT 브로커에 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.

- API 세부 정보는 AWS CLI 명령 참조의 GetConnectivityInfo 섹션을 참조하세요.

get-core-device

다음 코드 예시에서는 get-core-device 코드를 사용하는 방법을 보여줍니다.

AWS CLI

코어 디바이스 가져오기

다음 `get-core-device` 예시는 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.

- API 세부 정보는 AWS CLI 명령 참조의 GetCoreDevice 섹션을 참조하세요.

get-deployment

다음 코드 예시에서는 `get-deployment` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

배포를 가져오는 방법

다음 `get-deployment` 예시에서는 핵심 디바이스 그룹에 AWS IoT Greengrass 핵 구성 요소를 배포하는 방법에 대한 정보를 얻습니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 GetDeployment 섹션을 참조하세요.

`get-service-role-for-account`

다음 코드 예시에서는 `get-service-role-for-account` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS 계정의 Greengrass 서비스 역할을 가져오는 방법

다음 `get-service-role-for-account` 예시에서는 계정에 대해 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 Greengrass V2 개발자 안내서의 Greengrass 서비스 역할을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetServiceRoleForAccount 섹션을 참조하세요.

`list-client-devices-associated-with-core-device`

다음 코드 예시에서는 `list-client-devices-associated-with-core-device` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

클라이언트 디바이스를 코어 디바이스와 연결

다음 `list-client-devices-associated-with-core-device` 예시에서는 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 AWS IoT Greengrass V2 개발자 안내서의 로컬 IoT 디바이스와 상호 작용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListClientDevicesAssociatedWithCoreDevice 섹션을 참조하세요.

list-component-versions

다음 코드 예시에서는 `list-component-versions` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

구성 요소 버전 나열

다음 `list-component-versions` 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListComponentVersions 섹션을 참조하세요.

`list-components`

다음 코드 예시에서는 `list-components` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

구성 요소 나열

다음 `list-components` 예시에서는 현재 리전 내 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListComponents 섹션을 참조하세요.

`list-core-devices`

다음 코드 예시에서는 `list-core-devices` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

코어 디바이스를 나열하는 방법

다음 `list-core-devices` 예시에서는 현재 리전 내 AWS 계정의 AWS IoT Greengrass 핵심 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListCoreDevices 섹션을 참조하세요.

`list-deployments`

다음 코드 예시에서는 `list-deployments` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

배포 나열

다음 `list-deployments` 예시에서는 현재 리전 내 AWS 계정에 정의된 각 배포의 최신 개정 버전을 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListDeployments 섹션을 참조하세요.

list-effective-deployments

다음 코드 예시에서는 `list-effective-deployments` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

배포 작업 나열

다음 `list-effective-deployments` 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListEffectiveDeployments 섹션을 참조하세요.

list-installed-components

다음 코드 예시에서는 `list-installed-components` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

코어 디바이스에 설치된 구성 요소를 나열하는 방법

다음 `list-installed-components` 예시에서는 AWS IoT Greengrass 코어 디바이스에 설치되는 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListInstalledComponents 섹션을 참조하세요.

list-tags-for-resource

다음 코드 예시에서는 list-tags-for-resource 코드를 사용하는 방법을 보여줍니다.

AWS CLI

리소스의 태그 나열

다음 list-tags-for-resource 예시에서는 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 리소스에 태그 지정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource를 참조하세요.

tag-resource

다음 코드 예시에서는 tag-resource 코드를 사용하는 방법을 보여줍니다.

AWS CLI

리소스에 태그를 추가하는 방법

다음 tag-resource 예시에서는 AWS IoT Greengrass 코어 디바이스에 소유자 태그를 추가합니다. 이 태그를 사용하여 소유자에 따라 코어 디바이스에 대한 액세스 권한을 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 리소스에 태그 지정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

untag-resource

다음 코드 예시에서는 untag-resource의 사용 방법을 보여줍니다.

AWS CLI

리소스에서 태그 제거

다음 untag-resource 예시에서는 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 리소스에 태그 지정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

update-connectivity-info

다음 코드 예시에서는 update-connectivity-info 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Greengrass 코어 디바이스의 연결 정보를 업데이트하는 방법

다음 update-connectivity-info 예시는 Greengrass 코어 디바이스에 대한 연결 정보를 가져오는 예시입니다. 클라이언트 디바이스는 이 정보를 사용하여 이 코어 디바이스에서 실행되는 MQTT 브로커에 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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateConnectivityInfo 섹션을 참조하세요.

AWS CLI를 사용한 AWS IoT Jobs SDK release 예시

다음 코드 예시에서는 AWS IoT Jobs SDK release에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

describe-job-execution

다음 코드 예시에서는 describe-job-execution의 사용 방법을 보여줍니다.

AWS CLI

작업 실행 세부 정보 가져오기

다음 describe-job-execution 예시에서는 지정된 작업 및 사물의 최근 실행에 대한 세부 정보를 가져옵니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeJobExecution을 참조하세요.

get-pending-job-executions

다음 코드 예시에서는 get-pending-job-executions의 사용 방법을 보여줍니다.

AWS CLI

종료 상태가 아닌 모든 사물 관련 작업 목록 가져오기

다음 get-pending-job-executions 예시에서는 지정된 사물과 관련하여 종료 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.

- API 세부 정보는 AWS CLI 명령 참조의 GetPendingJobExecutions를 참조하세요.

`start-next-pending-job-execution`

다음 코드 예시에서는 start-next-pending-job-execution의 사용 방법을 보여줍니다.

AWS CLI

사물의 다음 대기 중 작업 실행을 가져오고 시작

다음 start-next-pending-job-execution 예시에서는 지정된 사물과 관련하여 상태가 IN_PROGRESS 또는 QUEUED인 다음 작업 실행을 가져오고 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.

- API 세부 정보는 AWS CLI 명령 참조의 StartNextPendingJobExecution을 참조하세요.

update-job-execution

다음 코드 예시에서는 `update-job-execution`의 사용 방법을 보여줍니다.

AWS CLI

작업 실행의 상태 업데이트

다음 `update-job-execution` 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateJobExecution을 참조하세요.

AWS CLI를 사용한 AWS IoT SiteWise 예시

다음 코드 예시에서는 AWS IoT SiteWise에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`associate-assets`

다음 코드 예시에서는 `associate-assets`의 사용 방법을 보여줍니다.

AWS CLI

하위 자산을 상위 자산에 연결하는 방법

다음 `associate-assets` 예시에서는 풍력 터빈 자산을 풍력 발전 단지 자산에 연결합니다. 풍력 터빈 자산 모델은 풍력 발전 단지 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 Associating assets를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AssociateAssets 섹션을 참조하세요.

batch-associate-project-assets

다음 코드 예시에서는 batch-associate-project-assets의 사용 방법을 보여줍니다.

AWS CLI

자산을 프로젝트에 연결하는 방법

다음 batch-associate-project-assets 예시에서는 풍력 발전 단지 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 모니터 애플리케이션 설명서의 Adding assets to projects를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 BatchAssociateProjectAssets 섹션을 참조하세요.

batch-disassociate-project-assets

다음 코드 예시에서는 batch-disassociate-project-assets의 사용 방법을 보여줍니다.

AWS CLI

프로젝트에서 자산을 연결 해제하는 방법

다음 batch-disassociate-project-assets 예시에서는 프로젝트에서 풍력 발전 단지 자산을 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 모니터 애플리케이션 설명서의 [Adding assets to projects](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [BatchDisassociateProjectAssets](#) 섹션을 참조하세요.

batch-put-asset-property-value

다음 코드 예시에서는 `batch-put-asset-property-value`의 사용 방법을 보여줍니다.

AWS CLI

자산 속성으로 데이터 보내기

다음 `batch-put-asset-property-value` 예시에서는 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.json`의 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 사용 설명서의 Ingesting data using the AWS IoT SiteWise API 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 BatchPutAssetPropertyValue 섹션을 참조하세요.

create-access-policy

다음 코드 예시에서는 `create-access-policy`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 포털에 대한 사용자 관리 액세스 권한 부여

다음 `create-access-policy` 예시에서는 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.json의 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 Adding or removing portal administrators를 참조하세요.

예시 2: 사용자에게 프로젝트에 대한 읽기 전용 액세스 권한 부여

다음 `create-access-policy` 예시에서는 풍력 발전소 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.json의 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 모니터 애플리케이션 설명서의 Assigning project viewers를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateAccessPolicy 섹션을 참조하세요.

`create-asset-model`

다음 코드 예시에서는 `create-asset-model`의 사용 방법을 보여줍니다.

AWS CLI

자산 모델 생성

다음 `create-asset-model` 예시에서는 다음 속성을 가진 풍력 터빈을 정의하는 자산 모델을 생성합니다.

일련 번호 - 풍력 터빈의 일련 번호발전 전력 - 풍력 터빈의 발전 전력 데이터 스트림온도 C - 풍력 터빈의 온도 데이터 스트림(섭씨 단위)온도 F - 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 Defining asset models를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateAssetModel 섹션을 참조하세요.

create-asset

다음 코드 예시에서는 create-asset의 사용 방법을 보여줍니다.

AWS CLI

자산 생성

다음 create-asset 예시에서는 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 Creating assets를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateAsset 섹션을 참조하세요.

create-dashboard

다음 코드 예시에서는 create-dashboard의 사용 방법을 보여줍니다.

AWS CLI

대시보드 생성

다음 create-dashboard 예시에서는 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.json`의 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 [Creating dashboards (CLI)](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [CreateDashboard](#) 섹션을 참조하세요.

`create-gateway`

다음 코드 예시에서는 `create-gateway`의 사용 방법을 보여줍니다.

AWS CLI

게이트웨이 생성

다음 `create-gateway` 예시에서는 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 Configuring a gateway를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateGateway 섹션을 참조하세요.

`create-portal`

다음 코드 예시에서는 `create-portal`의 사용 방법을 보여줍니다.

AWS CLI

포털 생성

다음 `create-portal` 예시에서는 풍력 발전 단지 회사의 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 Getting started with AWS IoT SiteWise Monitor 및 AWS IoT SiteWise 사용 설명서의 Enabling AWS SSO를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreatePortal 섹션을 참조하세요.

create-project

다음 코드 예시에서는 create-project의 사용 방법을 보여줍니다.

AWS CLI

프로젝트 생성

다음 create-project 예시에서는 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 모니터 애플리케이션 설명서의 Creating projects를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateProject를 참조하세요.

delete-access-policy

다음 코드 예시에서는 delete-access-policy의 사용 방법을 보여줍니다.

AWS CLI

프로젝트 또는 포털에 대한 사용자의 액세스를 취소하는 방법

다음 delete-access-policy 예시에서는 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 Adding or removing portal administrators를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteAccessPolicy 섹션을 참조하세요.

delete-asset-model

다음 코드 예시에서는 delete-asset-model의 사용 방법을 보여줍니다.

AWS CLI

자산 모델 삭제

다음 delete-asset-model 예시에서는 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 Deleting asset models를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteAssetModel 섹션을 참조하세요.

delete-asset

다음 코드 예시에서는 delete-asset의 사용 방법을 보여줍니다.

AWS CLI

자산 삭제

다음 delete-asset 예시에서는 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 Deleting assets를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteAsset 섹션을 참조하세요.

delete-dashboard

다음 코드 예시에서는 delete-dashboard의 사용 방법을 보여줍니다.

AWS CLI

대시보드 삭제

다음 delete-dashboard 예시에서는 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 모니터 애플리케이션 설명서의 Deleting dashboards를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteDashboard 섹션을 참조하세요.

delete-gateway

다음 코드 예시에서는 delete-gateway의 사용 방법을 보여줍니다.

AWS CLI

게이트웨이 삭제

다음 delete-gateway 예시에서는 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 Ingesting data using a gateway를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteGateway 섹션을 참조하세요.

delete-portal

다음 코드 예시에서는 delete-portal의 사용 방법을 보여줍니다.

AWS CLI

포털 삭제

다음 delete-portal 예시에서는 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 Deleting a portal을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeletePortal 섹션을 참조하세요.

delete-project

다음 코드 예시에서는 delete-project의 사용 방법을 보여줍니다.

AWS CLI

프로젝트 삭제

다음 delete-project 예시에서는 풍력 발전 단지 프로젝트를 삭제합니다.

```
aws iotsitewise delete-project \
    --project-id a1b2c3d4-5678-90ab-cdef-eeeeeEXAMPLE
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT SiteWise 모니터 애플리케이션 설명서의 Deleting projects를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteProject 섹션을 참조하세요.

describe-access-policy

다음 코드 예시에서는 describe-access-policy의 사용 방법을 보여줍니다.

AWS CLI

액세스 정책을 설명하는 방법

다음 describe-access-policy 예시에서는 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 Adding or removing portal administrators를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeAccessPolicy 섹션을 참조하세요.

describe-asset-model

다음 코드 예시에서는 describe-asset-model의 사용 방법을 보여줍니다.

AWS CLI

자산 모델을 설명하는 방법

다음 describe-asset-model 예시에서는 풍력 발전 단지 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 Describing a specific asset model을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeAssetModel 섹션을 참조하세요.

describe-asset-property

다음 코드 예시에서는 describe-asset-property의 사용 방법을 보여줍니다.

AWS CLI

자산 속성을 설명하는 방법

다음 describe-asset-property 예시에서는 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 Describing a specific asset property를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeAssetProperty 섹션을 참조하세요.

describe-asset

다음 코드 예시에서는 `describe-asset`의 사용 방법을 보여줍니다.

AWS CLI

자산을 설명하는 방법

다음 `describe-asset` 예시에서는 풍력 발전 단지 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 Describing a specific asset을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeAsset 섹션을 참조하세요.

describe-dashboard

다음 코드 예시에서는 describe-dashboard의 사용 방법을 보여줍니다.

AWS CLI

대시보드를 설명하는 방법

다음 describe-dashboard 예시에서는 지정된 풍력 발전 단지 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 모니터 애플리케이션 설명서의 Viewing dashboards를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDashboard 섹션을 참조하세요.

describe-gateway-capability-configuration

다음 코드 예시에서는 describe-gateway-capability-configuration의 사용 방법을 보여줍니다.

AWS CLI

게이트웨이 기능을 설명하는 방법

다음 describe-gateway-capability-configuration 예시에서는 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 Configuring data sources를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeGatewayCapabilityConfiguration 섹션을 참조하세요.

describe-gateway

다음 코드 예시에서는 describe-gateway의 사용 방법을 보여줍니다.

AWS CLI

게이트웨이를 설명하는 방법

다음 describe-gateway 예시에서는 게이트웨이를 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 Ingesting data using a gateway를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeGateway 섹션을 참조하세요.

`describe-logging-options`

다음 코드 예시에서는 `describe-logging-options`의 사용 방법을 보여줍니다.

AWS CLI

현재 AWS IoT SiteWise 로깅 옵션을 검색하는 방법

다음 `describe-logging-options` 예시에서는 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 Monitoring AWS IoT SiteWise with Amazon CloudWatch Logs를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeLoggingOptions 섹션을 참조하세요.

describe-portal

다음 코드 예시에서는 describe-portal의 사용 방법을 보여줍니다.

AWS CLI

포털을 설명하는 방법

다음 describe-portal 예시에서는 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 Administering your portals를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribePortal 섹션을 참조하세요.

describe-project

다음 코드 예시에서는 describe-project의 사용 방법을 보여줍니다.

AWS CLI

프로젝트 설명

다음 describe-project 예시에서는 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 모니터 애플리케이션 설명서의 Viewing project details를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeProject 섹션을 참조하세요.

`disassociate-assets`

다음 코드 예시에서는 `disassociate-assets`의 사용 방법을 보여줍니다.

AWS CLI

상위 자산에서 하위 자산을 연결 해제하는 방법

다음 `disassociate-assets` 예시에서는 풍력 터빈 자산을 풍력 발전 단지 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 Associating assets를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisassociateAssets 섹션을 참조하세요.

`get-asset-property-aggregates`

다음 코드 예시에서는 `get-asset-property-aggregates`의 사용 방법을 보여줍니다.

AWS CLI

자산 속성의 집계된 평균 및 개수 값을 검색하는 방법

다음 `get-asset-property-aggregates` 예시에서는 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 Querying asset property aggregates를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetAssetPropertyAggregates 섹션을 참조하세요.

get-asset-property-value-history

다음 코드 예시에서는 `get-asset-property-value-history`의 사용 방법을 보여줍니다.

AWS CLI

자산 속성의 기록 값을 검색하는 방법

다음 `get-asset-property-value-history` 예시에서는 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 Querying historical asset property values를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetAssetPropertyValueHistory 섹션을 참조하세요.

`get-asset-property-value`

다음 코드 예시에서는 `get-asset-property-value`의 사용 방법을 보여줍니다.

AWS CLI

자산 속성의 현재 값을 검색하는 방법

다음 `get-asset-property-value` 예시에서는 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 Querying current asset property values를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetAssetPropertyValue 섹션을 참조하세요.

`list-access-policies`

다음 코드 예시에서는 `list-access-policies`의 사용 방법을 보여줍니다.

AWS CLI

모든 액세스 정책 나열

다음 `list-access-policies` 예시에서는 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 Administering your portals를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListAccessPolicies 섹션을 참조하세요.

`list-asset-models`

다음 코드 예시에서는 `list-asset-models`의 사용 방법을 보여줍니다.

AWS CLI

모든 자산 모델 나열

다음 `list-asset-models` 예시에서는 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 Listing all asset models를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListAssetModels 섹션을 참조하세요.

list-assets

다음 코드 예시에서는 `list-assets`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 모든 최상위 자산 나열

다음 `list-assets` 예시에서는 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 [Listing assets](#)를 참조하세요.

예시 2: 자산 모델을 기반으로 모든 자산 나열

다음 `list-assets` 예시에서는 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 Listing assets를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListAssets 섹션을 참조하세요.

list-associated-assets

다음 코드 예시에서는 `list-associated-assets`의 사용 방법을 보여줍니다.

AWS CLI

특정 계층 구조의 자산에 연결된 모든 자산 나열

다음 `list-associated-assets` 예시에서는 지정된 풍력 발전 단지 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 Listing assets associated to a specific asset을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListAssociatedAssets 섹션을 참조하세요.

list-dashboards

다음 코드 예시에서는 `list-dashboards`의 사용 방법을 보여줍니다.

AWS CLI

프로젝트의 모든 대시보드를 나열하는 방법

다음 `list-dashboards` 예시에서는 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 모니터 애플리케이션 설명서의 Viewing dashboards를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListDashboards 섹션을 참조하세요.

`list-gateways`

다음 코드 예시에서는 `list-gateways`의 사용 방법을 보여줍니다.

AWS CLI

모든 게이트웨이를 나열하는 방법

다음 `list-gateways` 예시에서는 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 Ingesting data using a gateway를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListGateways 섹션을 참조하세요.

list-portals

다음 코드 예시에서는 `list-portals`의 사용 방법을 보여줍니다.

AWS CLI

모든 포털을 나열하는 방법

다음 `list-portals` 예시에서는 현재 리전 내 AWS 계정의 모든 유지 관리 기간을 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 Administering your portals를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ListPortals](https://docs.aws.amazon.com/goto/aws-cli/iotsitewise-2019-12-02/ListPortals) 섹션을 참조하세요.

list-project-assets

다음 코드 예시에서는 `list-project-assets`의 사용 방법을 보여줍니다.

AWS CLI

프로젝트에 연결된 모든 자산을 나열하는 방법

다음 `list-project-assets` 예시에서는 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 모니터 애플리케이션 설명서의 [Adding assets to projects](https://docs.aws.amazon.com/iot-sitewise/latest/appguide/add-assets-to-projects-sd.html)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ListProjectAssets](https://docs.aws.amazon.com/goto/aws-cli/iotsitewise-2019-12-02/ListProjectAssets) 섹션을 참조하세요.

list-projects

다음 코드 예시에서는 `list-projects`의 사용 방법을 보여줍니다.

AWS CLI

포털의 모든 프로젝트를 나열하는 방법

다음 `list-projects` 예시에서는 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 모니터 애플리케이션 설명서의 Viewing project details를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListProjects 섹션을 참조하세요.

list-tags-for-resource

다음 코드 예시에서는 list-tags-for-resource의 사용 방법을 보여줍니다.

AWS CLI

리소스의 모든 태그 나열

다음 list-tags-for-resource 예시에서는 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 Tagging your resources를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource를 참조하세요.

put-logging-options

다음 코드 예시에서는 put-logging-options의 사용 방법을 보여줍니다.

AWS CLI

로깅 수준을 지정하는 방법

다음 put-logging-options 예시에서는 AWS IoT SiteWise 에서 INFO 수준 로깅을 활성화합니다. 다른 수준에는 DEBUG 및 OFF가 포함됩니다.

```
aws iotsitewise put-logging-options \
    --logging-options level=INFO
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT SiteWise 사용 설명서의 Monitoring AWS IoT SiteWise with Amazon CloudWatch Logs를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutLoggingOptions 섹션을 참조하세요.

tag-resource

다음 코드 예시에서는 tag-resource의 사용 방법을 보여줍니다.

AWS CLI

리소스에 태그 추가

다음 tag-resource 예시에서는 풍력 터빈 자산에 소유자 태그를 추가합니다. 이를 통해 자산 소유자에 따라 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 Tagging your resources를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

untag-resource

다음 코드 예시에서는 untag-resource의 사용 방법을 보여줍니다.

AWS CLI

리소스에서 태그 제거

다음 untag-resource 예시에서는 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 Tagging your resources를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

update-access-policy

다음 코드 예시에서는 update-access-policy의 사용 방법을 보여줍니다.

AWS CLI

프로젝트 시청자에게 프로젝트의 소유권을 부여하는 방법

다음 update-access-policy 예시에서는 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 모니터 애플리케이션 설명서의 Assigning project owners를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateAccessPolicy 섹션을 참조하세요.

update-asset-model

다음 코드 예시에서는 update-asset-model의 사용 방법을 보여줍니다.

AWS CLI

자산 모델 생성

다음 update-asset-model 예시에서는 풍력 발전소 자산 모델의 설명을 업데이트합니다. 이 예시에는 update-asset-model이 기존 모델을 새 모델로 덮어쓰기 때문에 모델의 기존 ID와 정의가 포함되어 있습니다.

```
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 Updating asset models를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateAssetModel 섹션을 참조하세요.

update-asset-property

다음 코드 예시에서는 update-asset-property의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 자산 속성의 별칭 업데이트

다음 update-asset-property 예시에서는 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 Mapping industrial data streams to asset properties를 참조하세요.

예시 2: 자산 속성 알림 활성화

다음 `update-asset-property` 예시에서는 풍력 터빈 자산의 전력 속성에 대한 자산 속성 업데이트 알림을 활성화합니다. 속성 값 업데이트는 MQTT 주제 `$aws/sitewise/asset-models/<assetModelId>/assets/<assetId>/properties/<propertyId>`에 게시되며, 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 Interacting with other services를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateAssetProperty 섹션을 참조하세요.

`update-asset`

다음 코드 예시에서는 `update-asset`의 사용 방법을 보여줍니다.

AWS CLI

자산의 이름 업데이트

다음 `update-asset` 예시에서는 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 Updating assets를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateAsset 섹션을 참조하세요.

update-dashboard

다음 코드 예시에서는 update-dashboard의 사용 방법을 보여줍니다.

AWS CLI

대시보드 업데이트

다음 update-dashboard 예시에서는 풍력 발전 단지의 총 발전량을 표시하는 대시보드의 꺾은선형 차트의 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 Creating dashboards (CLI)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateDashboard 섹션을 참조하세요.

`update-gateway-capability-configuration`

다음 코드 예시에서는 `update-gateway-capability-configuration`의 사용 방법을 보여줍니다.

AWS CLI

게이트웨이 기능을 업데이트하는 방법

다음 `update-gateway-capability-configuration` 예시에서는 다음 속성을 사용하여 OPC-UA 소스를 구성합니다

모든 인증서를 신뢰합니다.Basic256 알고리즘을 사용하여 메시지를 보호합니다. 서명 및 암호화 모드를 사용하여 연결을 보호합니다. AWS Secrets Manager 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.json`의 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 Configuring data sources를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateGatewayCapabilityConfiguration 섹션을 참조하세요.

update-gateway

다음 코드 예시에서는 update-gateway의 사용 방법을 보여줍니다.

AWS CLI

게이트웨이의 이름 업데이트

다음 update-gateway 예시에서는 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 Ingesting data using a gateway를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateGateway 섹션을 참조하세요.

update-portal

다음 코드 예시에서는 update-portal의 사용 방법을 보여줍니다.

AWS CLI

포털의 세부 정보를 업데이트하는 방법

다음 update-portal 예시에서는 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 Administering your portals를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdatePortal 섹션을 참조하세요.

update-project

다음 코드 예시에서는 update-project의 사용 방법을 보여줍니다.

AWS CLI

프로젝트 세부 정보를 업데이트하는 방법

다음 update-project 예시에서는 풍력 발전 단지 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 모니터 애플리케이션 설명서의 Changing project details를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateProject 섹션을 참조하세요.

AWS CLI를 사용한 AWS IoT Things Graph 예시

다음 코드 예시에서는 AWS IoT Things Graph에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

associate-entity-to-thing

다음 코드 예시에서는 `associate-entity-to-thing` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사물을 디바이스와 연결하는 방법

다음 `associate-entity-to-thing` 예시에서는 사물을 디바이스와 연결합니다. 이 예시에서는 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 사용자 안내서의 Creating and Uploading Models를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AssociateEntityToThing 섹션을 참조하세요.

`create-flow-template`

다음 코드 예시에서는 `create-flow-template` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

흐름 생성

다음 `create-flow-template` 예시에서는 흐름(워크플로)을 생성합니다. `MyFlowDefinition`의 값은 흐름을 모델링하는 GraphQL입니다.

```
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 사용자 안내서의 Working with Flows를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateFlowTemplate 섹션을 참조하세요.

`create-system-instance`

다음 코드 예시에서는 `create-system-instance` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

시스템 인스턴스 생성

다음 `create-system-instance` 예시에서는 시스템 인스턴스를 생성합니다. `MySystemInstanceDefinition`의 값은 시스템 인스턴스를 모델링하는 GraphQL입니다.

```
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 사용자 안내서의 Working with Systems and Flow Configurations를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateSystemInstance 섹션을 참조하세요.

`create-system-template`

다음 코드 예시에서는 `create-system-template` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

시스템 생성

다음 `create-system-template` 예시에서는 시스템을 생성합니다. MySystemDefinition의 값은 시스템을 모델링하는 GraphQL입니다.

```
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 사용자 안내서의 Creating Systems를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateSystemTemplate 섹션을 참조하세요.

delete-flow-template

다음 코드 예시에서는 delete-flow-template 코드를 사용하는 방법을 보여줍니다.

AWS CLI

흐름을 삭제하는 방법

다음 delete-flow-template 예시에서는 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 사용자 안내서의 Lifecycle Management for AWS IoT Things Graph Entities, Flows, Systems, and Deployments를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteFlowTemplate 섹션을 참조하세요.

delete-namespace

다음 코드 예시에서는 delete-namespace 코드를 사용하는 방법을 보여줍니다.

AWS CLI

네임스페이스 삭제

다음 delete-namespace 예시에서는 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 사용자 안내서의 Lifecycle Management for AWS IoT Things Graph Entities, Flows, Systems, and Deployments를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteNamespace 섹션을 참조하세요.

delete-system-instance

다음 코드 예시에서는 `delete-system-instance` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

시스템 인스턴스 삭제

다음 `delete-system-instance` 예시에서는 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 사용자 안내서의 Lifecycle Management for AWS IoT Things Graph Entities, Flows, Systems, and Deployments를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteSystemInstance 섹션을 참조하세요.

delete-system-template

다음 코드 예시에서는 `delete-system-template` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

시스템 삭제

다음 `delete-system-template` 예시에서는 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 사용자 안내서의 Lifecycle Management for AWS IoT Things Graph Entities, Flows, Systems, and Deployments를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteSystemTemplate 섹션을 참조하세요.

deploy-system-instance

다음 코드 예시에서는 deploy-system-instance 코드를 사용하는 방법을 보여줍니다.

AWS CLI

시스템 인스턴스를 배포하는 방법

다음 delete-system-template 예시에서는 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 사용자 안내서의 Working with Systems and Flow Configurations를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeploySystemInstance 섹션을 참조하세요.

deprecate-flow-template

다음 코드 예시에서는 deprecate-flow-template 코드를 사용하는 방법을 보여줍니다.

AWS CLI

흐름 사용을 중단하는 방법

다음 `deprecate-flow-template` 예시에서는 흐름(워크플로) 사용을 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 사용자 안내서의 Lifecycle Management for AWS IoT Things Graph Entities, Flows, Systems, and Deployments를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeprecateFlowTemplate 섹션을 참조하세요.

deprecate-system-template

다음 코드 예시에서는 `deprecate-system-template` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

시스템 사용을 중단하는 방법

다음 `deprecate-system-template` 예시에서는 시스템 사용을 중지합니다.

```
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 사용자 안내서의 Lifecycle Management for AWS IoT Things Graph Entities, Flows, Systems, and Deployments를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeprecateSystemTemplate 섹션을 참조하세요.

describe-namespace

다음 코드 예시에서는 `describe-namespace` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

네임스페이스에 대한 설명을 가져오는 방법

다음 describe-namespace 예시에서는 네임스페이스 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 사용자 안내서의 Namespaces를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeNamespace 섹션을 참조하세요.

dissociate-entity-from-thing

다음 코드 예시에서는 dissociate-entity-from-thing 코드를 사용하는 방법을 보여줍니다.

AWS CLI

디바이스에서 사물을 연결 해제하는 방법

다음 dissociate-entity-from-thing 예시에서는 디바이스에서 사물을 연결 해제합니다.

```
aws iotthingsgraph dissociate-entity-from-thing \
    --thing-name "MotionSensorName" \
    --entity-type "DEVICE"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 사물 그래프 사용자 안내서의 Creating and Uploading Models를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DissociateEntityFromThing 섹션을 참조하세요.

get-entities

다음 코드 예시에서는 get-entities 코드를 사용하는 방법을 보여줍니다.

AWS CLI

엔터티에 대한 정의를 가져오는 방법

다음 `get-entities` 예시에서는 디바이스 모델의 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 사용자 안내서의 Creating and Uploading Models를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetEntities 섹션을 참조하세요.

`get-flow-template-revisions`

다음 코드 예시에서는 `get-flow-template-revisions` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

흐름에 대한 개정 정보를 가져오는 방법

다음 `get-flow-template-revisions` 예시에서는 흐름(워크플로)의 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 사용자 안내서의 Working with Flows를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetFlowTemplateRevisions 섹션을 참조하세요.

get-flow-template

다음 코드 예시에서는 get-flow-template 코드를 사용하는 방법을 보여줍니다.

AWS CLI

흐름 정의 가져오기

다음 get-flow-template 예시에서는 흐름(워크플로) 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 사용자 안내서의 Working with Flows를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetFlowTemplate 섹션을 참조하세요.

get-namespace-deletion-status

다음 코드 예시에서는 get-namespace-deletion-status 코드를 사용하는 방법을 보여줍니다.

AWS CLI

네임스페이스 삭제 태스크의 상태를 가져오기

다음 get-namespace-deletion-status 예시에서는 네임스페이스 삭제 태스크의 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 사용자 안내서의 Namespaces를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetNamespaceDeletionStatus 섹션을 참조하세요.

get-system-instance

다음 코드 예시에서는 `get-system-instance` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

시스템 인스턴스 가져오기

다음 `get-system-instance` 예시에서는 시스템 인스턴스의 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 사용자 안내서의 Working with Systems and Flow Configurations를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetSystemInstance 섹션을 참조하세요.

`get-system-template-revisions`

다음 코드 예시에서는 `get-system-template-revisions` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

시스템에 대한 개정 정보를 가져오는 방법

다음 `get-system-template-revisions` 예시에서는 시스템의 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 사용자 안내서의 Working with Systems and Flow Configurations를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetSystemTemplateRevisions 섹션을 참조하세요.

get-system-template

다음 코드 예시에서는 get-system-template 코드를 사용하는 방법을 보여줍니다.

AWS CLI

시스템을 가져오는 방법

다음 get-system-template 예시에서는 시스템 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 사용자 안내서의 Working with Systems and Flow Configurations를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetSystemTemplate 섹션을 참조하세요.

get-upload-status

다음 코드 예시에서는 `get-upload-status` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

엔터티 업로드 상태를 가져오는 방법

다음 `get-upload-status` 예시에서는 엔터티 업로드 작업의 상태를 가져옵니다. `MyUploadId`의 값은 `upload-entity-definitions` 작업에서 반환되는 ID 값입니다.

```
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 사용자 안내서의 Modeling Entities를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetUploadStatus 섹션을 참조하세요.

list-flow-execution-messages

다음 코드 예시에서는 `list-flow-execution-messages` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

흐름 실행의 이벤트에 대한 정보를 가져오는 방법

다음 `list-flow-execution-messages` 예시에서는 흐름 실행의 이벤트 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 사용자 안내서의 Working with Flows를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListFlowExecutionMessages를 참조하세요.

list-tags-for-resource

다음 코드 예시에서는 `list-tags-for-resource` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

리소스의 모든 태그 나열

다음 `list-tags-for-resource` 예시에서는 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 사용자 안내서의 Tagging Your AWS IoT Things Graph Resources를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource를 참조하세요.

search-entities

다음 코드 예시에서는 search-entities 코드를 사용하는 방법을 보여줍니다.

AWS CLI

엔터티를 검색하는 방법

다음 search-entities 예시에서는 유형 EVENT의 모든 엔터티를 검색합니다.

```
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 사용자 안내서의 AWS IoT Things Graph Data Model Reference를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 SearchEntities 섹션을 참조하세요.

search-flow-executions

다음 코드 예시에서는 search-flow-executions 코드를 사용하는 방법을 보여줍니다.

AWS CLI

흐름 실행을 검색하는 방법

다음 search-flow-executions 예시에서는 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 사용자 안내서의 Working with Systems and Flow Configurations를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 SearchFlowExecutions 섹션을 참조하세요.

search-flow-templates

다음 코드 예시에서는 `search-flow-templates` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

흐름(또는 워크플로)을 검색하는 방법

다음 `search-flow-templates` 예시에서는 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 사용자 안내서의 Working with Flows를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 SearchFlowTemplates 섹션을 참조하세요.

search-system-instances

다음 코드 예시에서는 search-system-instances 코드를 사용하는 방법을 보여줍니다.

AWS CLI

시스템 인스턴스를 검색하는 방법

다음 search-system-instances 예시에서는 지정된 흐름을 포함하는 모든 시스템을 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 사용자 안내서의 Working with Systems and Flow Configurations를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 SearchSystemInstances 섹션을 참조하세요.

`search-system-templates`

다음 코드 예시에서는 `search-system-templates` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

시스템 검색

다음 `search-system-templates` 예시에서는 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 사용자 안내서의 Working with Flows를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 SearchSystemTemplates 섹션을 참조하세요.

search-things

다음 코드 예시에서는 search-things 코드를 사용하는 방법을 보여줍니다.

AWS CLI

디바이스 및 디바이스 모델과 연결된 사물을 검색하는 방법

다음 search-things 예시에서는 HCSR501MotionSensor 디바이스에 연결된 모든 사물을 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 사용자 안내서의 Creating and Uploading Models를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 SearchThings 섹션을 참조하세요.

tag-resource

다음 코드 예시에서는 tag-resource 코드를 사용하는 방법을 보여줍니다.

AWS CLI

리소스의 태그 생성

다음 tag-resource 예시에서는 지정된 리소스에 대한 태그를 만듭니다.

```
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 사용자 안내서의 Tagging Your AWS IoT Things Graph Resources를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

undeploy-system-instance

다음 코드 예시에서는 `undeploy-system-instance` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

대상에서 시스템 인스턴스를 배포 취소하는 방법

다음 `undeploy-system-instance` 예시에서는 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 사용자 안내서의 Lifecycle Management for AWS IoT Things Graph Entities, Flows, Systems, and Deployments를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UndeploySystemInstance 섹션을 참조하세요.

untag-resource

다음 코드 예시에서는 untag-resource 코드를 사용하는 방법을 보여줍니다.

AWS CLI

리소스의 태그 제거

다음 untag-resource 예시에서는 지정된 리소스의 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 사용자 안내서의 Tagging Your AWS IoT Things Graph Resources를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

update-flow-template

다음 코드 예시에서는 update-flow-template 코드를 사용하는 방법을 보여줍니다.

AWS CLI

흐름을 업데이트하는 방법

다음 update-flow-template 예시에서는 흐름(워크플로)을 업데이트합니다. MyFlowDefinition의 값은 흐름을 모델링하는 GraphQL입니다.

```
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 사용자 안내서의 Working with Flows를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateFlowTemplate 섹션을 참조하세요.

update-system-template

다음 코드 예시에서는 `update-system-template` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

시스템을 업데이트하는 방법

다음 `update-system-template` 예시에서는 시스템을 업데이트합니다. `MySystemDefinition`의 값은 시스템을 모델링하는 GraphQL입니다.

```
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 사용자 안내서의 Creating Systems를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [UpdateSystemTemplate](#) 섹션을 참조하세요.

`upload-entity-definitions`

다음 코드 예시에서는 `upload-entity-definitions` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

개체 정의를 업로드하는 방법

다음 `upload-entity-definitions` 예시에서는 네임스페이스에 엔터티 정의를 업로드합니다. `MyEntityDefinitions`의 값은 엔터티를 모델링하는 GraphQL입니다.

```
aws iotthingsgraph upload-entity-definitions \
    --document language=GRAPHQL,text="MyEntityDefinitions"
```

출력:

```
{
    "uploadId": "f6294f1e-b109-4bbe-9073-f451a2dda2da"
}
```

자세한 내용은 AWS IoT 사물 그래프 사용자 안내서의 Modeling Entities를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UploadEntityDefinitions 섹션을 참조하세요.

AWS CLI를 사용한 AWS IoT 무선 예시

다음 코드 예시에서는 AWS IoT 무선에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`associate-aws-account-with-partner-account`

다음 코드 예시에서는 `associate-aws-account-with-partner-account` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

파트너 계정을 AWS 계정과 연결하는 방법

다음 `associate-aws-account-with-partner-account` 예시에서는 다음 Sidewalk 계정 자격 증명을 AWS 계정에 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 [Amazon Sidewalk Integration for AWS IoT Core]을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [AssociateAwsAccountWithPartnerAccount] 섹션을 참조하세요.

`associate-wireless-device-with-thing`

다음 코드 예시에서는 `associate-wireless-device-with-thing` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

무선 디바이스에 사물을 연결하는 방법

다음 associate-wireless-device-with-thing 예시에서는 지정된 ID로 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 Add your gateways and wireless devices to AWS IoT Core for LoRaWAN을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AssociateWirelessDeviceWithThing 섹션을 참조하세요.

associate-wireless-gateway-with-certificate

다음 코드 예시에서는 associate-wireless-gateway-with-certificate 코드를 사용하는 방법을 보여줍니다.

AWS CLI

인증서를 무선 게이트웨이에 연결하는 방법

다음 associate-wireless-gateway-with-certificate 예시에서는 무선 게이트웨이를 인증서에 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 Add your gateways and wireless devices to AWS IoT Core for LoRaWAN을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AssociateWirelessGatewayWithCertificate 섹션을 참조하세요.

associate-wireless-gateway-with-thing

다음 코드 예시에서는 associate-wireless-gateway-with-thing 코드를 사용하는 방법을 보여줍니다.

AWS CLI

무선 게이트웨이에 사물을 연결하는 방법

다음 associate-wireless-gateway-with-thing 예시에서는 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 Add your gateways and wireless devices to AWS IoT Core for LoRaWAN을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AssociateWirelessGatewayWithThing 섹션을 참조하세요.

create-destination

다음 코드 예시에서는 create-destination 코드를 사용하는 방법을 보여줍니다.

AWS CLI

IoT 무선 대상을 생성하는 방법

다음 create-destination 예시에서는 디바이스 메시지를 AWS IoT 규칙에 매핑하기 위한 대상을 생성합니다. 이 명령을 실행하기 전에 AWS IoT 규칙으로 데이터를 전송하는 데 필요한 권한을 AWS IoT Core for LoRaWAN에 부여하는 IAM 역할을 생성해야 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 Add destinations to AWS IoT Core for LoRaWAN을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateDestination 섹션을 참조하세요.

`create-device-profile`

다음 코드 예시에서는 `create-device-profile` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

새 디바이스 프로파일 생성

다음 `create-device-profile` 예시에서는 새로운 IoT 무선 디바이스 프로파일을 만듭니다.

```
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 Add profiles to AWS IoT Core for LoRaWAN을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateDeviceProfile 섹션을 참조하세요.

`create-service-profile`

다음 코드 예시에서는 `create-service-profile` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

새 서비스 프로파일 생성

다음 `create-service-profile` 예시에서는 서비스 프로파일을 만듭니다.

```
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 Add profiles to AWS IoT Core for LoRaWAN을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateServiceProfile 섹션을 참조하세요.

`create-wireless-device`

다음 코드 예시에서는 `create-wireless-device` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

IoT 무선 디바이스를 생성하는 방법

다음 `create-wireless-device` 예시에서는 LoRaWAN 유형의 무선 디바이스 리소스를 만듭니다.

```
aws iotwireless create-wireless-device \
    --cli-input-json file://input.json
```

`input.json`의 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 Connecting devices and gateways to AWS IoT Core for LoRaWAN을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateWirelessDevice 섹션을 참조하세요.

`create-wireless-gateway-task-definition`

다음 코드 예시에서는 `create-wireless-gateway-task-definition` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

무선 게이트웨이 태스크 정의 생성

다음 `create-wireless-gateway-task-definition`은 지정된 현재 버전이 있는 모든 게이트웨이에 대해 이 태스크 정의를 사용하여 태스크를 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 Connecting devices and gateways to AWS IoT Core for LoRaWAN을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateWirelessGatewayTaskDefinition 섹션을 참조하세요.

create-wireless-gateway-task

다음 코드 예시에서는 create-wireless-gateway-task 코드를 사용하는 방법을 보여줍니다.

AWS CLI

무선 게이트웨이에 대한 태스크를 생성하는 방법

다음 `create-wireless-gateway-task` 예시에서는 무선 게이트웨이에 대한 태스크를 만듭니다.

```
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 Connecting devices and gateways to AWS IoT Core for LoRaWAN을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateWirelessGatewayTask 섹션을 참조하세요.

`create-wireless-gateway`

다음 코드 예시에서는 `create-wireless-gateway` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

무선 게이트웨이를 생성하는 방법

다음 `create-wireless-gateway` 예시에서는 무선 LoRaWAN 디바이스 게이트웨이를 만듭니다.

```
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 Connecting devices and gateways to AWS IoT Core for LoRaWAN을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateWirelessGateway 섹션을 참조하세요.

delete-destination

다음 코드 예시에서는 `delete-destination` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

IoT 무선 대상을 삭제하는 방법

다음 `delete-destination` 예시에서는 생성한 이름 `IoTWirelessDestination`을 사용하는 무선 대상 리소스를 삭제합니다.

```
aws iotwireless delete-destination \
    --name "IoTWirelessDestination"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 Add destinations to AWS IoT Core for LoRaWAN을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteDestination 섹션을 참조하세요.

delete-device-profile

다음 코드 예시에서는 `delete-device-profile` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

디바이스 프로파일 삭제

다음 `delete-device-profile` 예시에서는 생성한 지정된 ID로 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 Add profiles to AWS IoT Core for LoRaWAN을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteDeviceProfile 섹션을 참조하세요.

`delete-service-profile`

다음 코드 예시에서는 `delete-service-profile` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

서비스 프로파일을 삭제하는 방법

다음 `delete-service-profile` 예시에서는 생성한 지정된 ID로 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 Add profiles to AWS IoT Core for LoRaWAN을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteServiceProfile 섹션을 참조하세요.

`delete-wireless-device`

다음 코드 예시에서는 `delete-wireless-device` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

무선 디바이스를 삭제하는 방법

다음 `delete-wireless-device` 예시에서는 지정된 ID로 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 Connecting devices and gateways to AWS IoT Core for LoRaWAN을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteWirelessDevice 섹션을 참조하세요.

delete-wireless-gateway-task-definition

다음 코드 예시에서는 `delete-wireless-gateway-task-definition` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

무선 게이트웨이 태스크 정의 삭제

다음 `delete-wireless-gateway-task-definition` 예시에서는 다음 ID로 생성한 무선 게이트웨이 태스크 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 Connecting devices and gateways to AWS IoT Core for LoRaWAN을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteWirelessGatewayTaskDefinition 섹션을 참조하세요.

delete-wireless-gateway-task

다음 코드 예시에서는 `delete-wireless-gateway-task` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

무선 게이트웨이 태스크를 삭제하는 방법

다음 `delete-wireless-gateway-task` 예시에서는 지정된 ID로 무선 게이트웨이 태스크를 삭제합니다.

```
aws iotwireless delete-wireless-gateway-task \
```

```
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 Connecting devices and gateways to AWS IoT Core for LoRaWAN을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteWirelessGatewayTask 섹션을 참조하세요.

delete-wireless-gateway

다음 코드 예시에서는 delete-wireless-gateway 코드를 사용하는 방법을 보여줍니다.

AWS CLI

무선 게이트웨이를 삭제하는 방법

다음 delete-wireless-gateway 예시에서는 지정된 ID로 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 Connecting devices and gateways to AWS IoT Core for LoRaWAN을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteWirelessGateway 섹션을 참조하세요.

disassociate-aws-account-from-partner-account

다음 코드 예시에서는 disassociate-aws-account-from-partner-account 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS 계정에서 파트너 계정 연결을 해제하는 방법

다음 disassociate-aws-account-from-partner-account 예시에서는 현재 연결된 AWS 계정에서 파트너 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 Add your gateways and wireless devices to AWS IoT Core for LoRaWAN을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisassociateAwsAccountFromPartnerAccount 섹션을 참조하세요.

disassociate-wireless-device-from-thing

다음 코드 예시에서는 disassociate-wireless-device-from-thing 코드를 사용하는 방법을 보여줍니다.

AWS CLI

무선 디바이스에서 사물의 연결을 해제하는 방법

다음 disassociate-wireless-device-from-thing 예시에서는 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 Add your gateways and wireless devices to AWS IoT Core for LoRaWAN을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisassociateWirelessDeviceFromThing 섹션을 참조하세요.

disassociate-wireless-gateway-from-certificate

다음 코드 예시에서는 disassociate-wireless-gateway-from-certificate 코드를 사용하는 방법을 보여줍니다.

AWS CLI

무선 게이트웨이에서 인증서 연결을 해제하는 방법

다음 disassociate-wireless-gateway-from-certificate는 현재 연결된 인증서에서 무선 게이트웨이의 연결을 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 Add your gateways and wireless devices to AWS IoT Core for LoRaWAN을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisassociateWirelessGatewayFromCertificate 섹션을 참조하세요.

disassociate-wireless-gateway-from-thing

다음 코드 예시에서는 disassociate-wireless-gateway-from-thing 코드를 사용하는 방법을 보여줍니다.

AWS CLI

무선 게이트웨이에서 사물의 연결을 해제하는 방법

다음 disassociate-wireless-gateway-from-thing 예시에서는 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 Add your gateways and wireless devices to AWS IoT Core for LoRaWAN을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisassociateWirelessGatewayFromThing 섹션을 참조하세요.

get-destination

다음 코드 예시에서는 get-destination 코드를 사용하는 방법을 보여줍니다.

AWS CLI

IoT 무선 대상에 대한 정보를 가져오는 방법

다음 `get-destination` 예시에서는 생성한 이름 `IoTWirelessDestination`을 사용하여 대상 리소스의 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 Add destinations to AWS IoT Core for LoRaWAN을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetDestination 섹션을 참조하세요.

`get-device-profile`

다음 코드 예시에서는 `get-device-profile` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

인스턴스 프로파일 정보 가져오기

다음 `get-device-profile` 예시에서는 생성한 지정된 ID로 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 Add profiles to AWS IoT Core for LoRaWAN을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetDeviceProfile 섹션을 참조하세요.

get-partner-account

다음 코드 예시에서는 `get-partner-account` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

파트너 계정 정보를 가져오는 방법

다음 `get-partner-account` 예시에서는 다음 ID로 Sidewalk 계정의 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 Amazon Sidewalk Integration for AWS IoT Core을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetPartnerAccount 섹션을 참조하세요.

get-service-endpoint

다음 코드 예시에서는 get-service-endpoint 코드를 사용하는 방법을 보여줍니다.

AWS CLI

서비스 엔드포인트 가져오기

다음 get-service-endpoint 예시에서는 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 [Connecting devices and gateways to AWS IoT Core for LoRaWAN]을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [GetServiceEndpoint] 섹션을 참조하세요.

`get-service-profile`

다음 코드 예시에서는 `get-service-profile` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

인스턴스 프로파일 정보 가져오기

다음 `get-service-profile` 예시에서는 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 Add profiles to AWS IoT Core for LoRaWAN을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetServiceProfile 섹션을 참조하세요.

`get-wireless-device-statistics`

다음 코드 예시에서는 `get-wireless-device-statistics` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

무선 디바이스에 대한 작동 정보를 가져오는 방법

다음 `get-wireless-device-statistics` 예시에서는 무선 디바이스의 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 Connecting devices and gateways to AWS IoT Core for LoRaWAN을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetWirelessDeviceStatistics 섹션을 참조하세요.

get-wireless-device

다음 코드 예시에서는 get-wireless-device 코드를 사용하는 방법을 보여줍니다.

AWS CLI

무선 디바이스에 대한 정보를 가져오는 방법

다음 get-wireless-device 예시에서는 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 Connecting devices and gateways to AWS IoT Core for LoRaWAN을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetWirelessDevice 섹션을 참조하세요.

get-wireless-gateway-certificate

다음 코드 예시에서는 get-wireless-gateway-certificate 코드를 사용하는 방법을 보여줍니다.

AWS CLI

무선 게이트웨이와 연결된 인증서의 ID를 가져오는 방법

다음 get-wireless-gateway-certificate 예시에서는 지정된 ID로 무선 게이트웨이에 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 Connecting devices and gateways to AWS IoT Core for LoRaWAN을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetWirelessGatewayCertificate 섹션을 참조하세요.

get-wireless-gateway-firmware-information

다음 코드 예시에서는 get-wireless-gateway-firmware-information 코드를 사용하는 방법을 보여줍니다.

AWS CLI

무선 게이트웨이에 대한 펌웨어 정보를 가져오는 방법

다음 get-wireless-gateway-firmware-information 예시에서는 무선 게이트웨이의 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 Connecting devices and gateways to AWS IoT Core for LoRaWAN을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetWirelessGatewayFirmwareInformation 섹션을 참조하세요.

get-wireless-gateway-statistics

다음 코드 예시에서는 get-wireless-gateway-statistics 코드를 사용하는 방법을 보여줍니다.

AWS CLI

무선 게이트웨이에 대한 작동 정보를 가져오는 방법

다음 get-wireless-gateway-statistics 예시에서는 무선 게이트웨이의 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 Connecting devices and gateways to AWS IoT Core for LoRaWAN을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetWirelessGatewayStatistics 섹션을 참조하세요.

get-wireless-gateway-task-definition

다음 코드 예시에서는 get-wireless-gateway-task-definition 코드를 사용하는 방법을 보여줍니다.

AWS CLI

무선 게이트웨이 태스크 정의의 정보 가져오기

다음 get-wireless-gateway-task-definition 예시에서는 지정된 ID로 무선 태스크 정의의 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 Connecting devices and gateways to AWS IoT Core for LoRaWAN을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetWirelessGatewayTaskDefinition 섹션을 참조하세요.

get-wireless-gateway-task

다음 코드 예시에서는 `get-wireless-gateway-task` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

무선 게이트웨이 태스크에 대한 정보를 가져오는 방법

다음 `get-wireless-gateway-task` 예시에서는 지정된 ID를 가진 첨부 파일에 대한 정보를 반환합니다.

```
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 Connecting devices and gateways to AWS IoT Core for LoRaWAN을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetWirelessGatewayTask 섹션을 참조하세요.

get-wireless-gateway

다음 코드 예시에서는 `get-wireless-gateway` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

무선 게이트웨이에 대한 정보를 가져오는 방법

다음 get-wireless-gateway 예시에서는 무선 게이트웨이 myFirstLoRaWANGateway의 정보를 가져옵니다.

```
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 Connecting devices and gateways to AWS IoT Core for LoRaWAN을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetWirelessGateway 섹션을 참조하세요.

list-destinations

다음 코드 예시에서는 list-destinations 코드를 사용하는 방법을 보여줍니다.

AWS CLI

무선 대상을 나열하는 방법

다음 list-destinations 예시에서는 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 Add destinations to AWS IoT Core for LoRaWAN을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListDestinations 섹션을 참조하세요.

list-device-profiles

다음 코드 예시에서는 `list-device-profiles` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

디바이스 프로파일을 나열하는 방법

다음 `list-device-profiles` 예시에서는 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 [Add profiles to AWS IoT Core for LoRaWAN](#)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ListDeviceProfiles](#) 섹션을 참조하세요.

`list-partner-accounts`

다음 코드 예시에서는 `list-partner-accounts` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

파트너 계정을 나열하는 방법

다음 `list-partner-accounts` 예시에서는 AWS 계정에 연결된 사용 가능한 파트너 계정을 나열합니다.

```
aws iotwireless list-partner-accounts
```

출력:

```
{
    "Sidewalk": [
        {
            "AmazonId": "78965678771228",
            "Fingerprint":
 "bd96d8ef66dbfd2160eb60e156849e82ad7018b8b73c1ba0b4fc65c32498ee35"
        },
        {
```

```
            "AmazonId": "89656787651228",
            "Fingerprint":
 "bc5e99e151c07be14be7e6603e4489c53f858b271213a36ebe3370777ba06e9b"
        }
    ]
}
```

자세한 내용은 AWS IoT 개발자 안내서의 Amazon Sidewalk Integration for AWS IoT Core을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListPartnerAccounts 섹션을 참조하세요.

list-service-profiles

다음 코드 예시에서는 `list-service-profiles` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

서비스 프로파일을 나열하는 방법

다음 `list-service-profiles` 예시에서는 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 Add profiles to AWS IoT Core for LoRaWAN을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListServiceProfiles 섹션을 참조하세요.

list-tags-for-resource

다음 코드 예시에서는 `list-tags-for-resource` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

리소스에 할당된 태그 나열

다음 `list-tags-for-resource` 예시에서는 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 AWS IoT Core for LoRaWAN 리소스 설명을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource를 참조하세요.

list-wireless-devices

다음 코드 예시에서는 `list-wireless-devices` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용 가능한 무선 디바이스를 나열하는 방법

다음 `list-wireless-devices` 예시에서는 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 Connecting devices and gateways to AWS IoT Core for LoRaWAN을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListWirelessDevices 섹션을 참조하세요.

`list-wireless-gateway-task-definitions`

다음 코드 예시에서는 `list-wireless-gateway-task-definitions` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

무선 게이트웨이 태스크 정의를 나열하는 방법

다음 `list-wireless-gateway-task-definitions` 예시에서는 AWS 계정에 등록된 사용 가능한 무선 게이트웨이 태스크 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 Connecting devices and gateways to AWS IoT Core for LoRaWAN을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListWirelessGatewayTaskDefinitions 섹션을 참조하세요.

list-wireless-gateways

다음 코드 예시에서는 `list-wireless-gateways` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

무선 게이트웨이를 나열하는 방법

다음 `list-wireless-gateways` 예시에서는 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 Connecting devices and gateways to AWS IoT Core for LoRaWAN을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListWirelessGateways 섹션을 참조하세요.

send-data-to-wireless-device

다음 코드 예시에서는 `send-data-to-wireless-device` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

무선 디바이스로 데이터를 보내려면

다음 `send-data-to-wireless-device` 예시에서는 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 Connecting devices and gateways to AWS IoT Core for LoRaWAN을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 SendDataToWirelessDevice 섹션을 참조하세요.

tag-resource

다음 코드 예시에서는 tag-resource 코드를 사용하는 방법을 보여줍니다.

AWS CLI

리소스의 태그 키 및 값을 지정하는 방법

다음 tag-resource 예시에서는 무선 대상 IoTWirelessDestination에 태그 키 MyTag 및 값 MyValue를 지정합니다.

```
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 AWS IoT Core for LoRaWAN 리소스 설명을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

test-wireless-device

다음 코드 예시에서는 test-wireless-device 코드를 사용하는 방법을 보여줍니다.

AWS CLI

무선 디바이스를 테스트하는 방법

다음 `test-wireless-device` 예시에서는 `Hello`의 업링크 데이터를 지정된 ID를 사용하여 디바이스로 보냅니다.

```
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 AWS IoT Core for LoRaWAN에 디바이스 및 게이트웨이 연결을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TestWirelessDevice 섹션을 참조하세요.

`untag-resource`

다음 코드 예시에서는 `untag-resource` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

리소스에서 하나 이상의 태그 제거

다음 `untag-resource` 예시에서는 무선 대상 `IoTWirelessDestination`에서 태그 `MyTag` 및 해당 값을 제거합니다.

```
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 AWS IoT Core for LoRaWAN 리소스 설명을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

update-destination

다음 코드 예시에서는 update-destination 코드를 사용하는 방법을 보여줍니다.

AWS CLI

대상의 속성 업데이트

다음 update-destination 예시에서는 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 Add destinations to AWS IoT Core for LoRaWAN을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateDestination 섹션을 참조하세요.

update-partner-account

다음 코드 예시에서는 update-partner-account 코드를 사용하는 방법을 보여줍니다.

AWS CLI

파트너 계정의 속성을 업데이트하는 방법

다음 update-partner-account는 지정된 ID로 계정의 AppServerPrivateKey를 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 Amazon Sidewalk Integration for AWS IoT Core을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdatePartnerAccount 섹션을 참조하세요.

`update-wireless-device`

다음 코드 예시에서는 `update-wireless-device` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

무선 디바이스의 속성을 업데이트하는 방법

다음 `update-wireless-device` 예시에서는 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 Connecting devices and gateways to AWS IoT Core for LoRaWAN을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateWirelessDevice 섹션을 참조하세요.

`update-wireless-gateway`

다음 코드 예시에서는 `update-wireless-gateway` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

무선 게이트웨이를 업데이트하는 방법

다음 `update-wireless-gateway` 예시에서는 무선 게이트웨이의 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 Connecting devices and gateways to AWS IoT Core for LoRaWAN을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateWirelessGateway 섹션을 참조하세요.

AWS CLI를 사용한 Amazon IVS 예시

다음 코드 예시에서는 Amazon IVS에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

작업

batch-get-channel

다음 코드 예시에서는 batch-get-channel 코드를 사용하는 방법을 보여줍니다.

AWS CLI

여러 채널에 대한 채널 구성 정보를 가져오는 방법

다음 batch-get-channel 예시에서는 지정된 채널의 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,
            "containerFormat": "TS",
            "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",
            "insecureIngest": false,
            "latencyMode": "LOW",
            "multitrackInputConfiguration": {
```

```
                "enabled": false,
                "maximumResolution": "FULL_HD",
                "policy": "ALLOW"
            },
            "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,
            "containerFormat": "FRAGMENTED_MP4",
            "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",
            "insecureIngest": false,
            "latencyMode": "LOW",
            "multitrackInputConfiguration": {
                "enabled": true,
                "maximumResolution": "FULL_HD",
                "policy": "ALLOW"
            },
            "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 Low-Latency 사용 설명서의 채널 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 BatchGetChannel 섹션을 참조하세요.

batch-get-stream-key

다음 코드 예시에서는 batch-get-stream-key 코드를 사용하는 방법을 보여줍니다.

AWS CLI

여러 스트림 키에 대한 정보를 가져오는 방법

다음 batch-get-stream-key 예시에서는 지정된 스트림 키의 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 Low-Latency 사용 설명서의 채널 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 BatchGetStreamKey 섹션을 참조하세요.

batch-start-viewer-session-revocation

다음 코드 예시에서는 batch-start-viewer-session-revocation 코드를 사용하는 방법을 보여줍니다.

AWS CLI

여러 채널 ARN 및 뷰어 ID 페어에 대한 뷰어 세션을 취소하는 방법

다음 batch-start-viewer-session-revocation 예시에서는 여러 채널 ARN 및 뷰어 ID 페어에서 세션 취소를 동시에 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 내용은 Amazon Interactive Video Service 사용자 안내서의 프라이빗 채널 설정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 BatchStartViewerSessionRevocation 섹션을 참조하세요.

create-channel

다음 코드 예시에서는 create-channel 코드를 사용하는 방법을 보여줍니다.

AWS CLI

예시 1: 녹음이 없는 채널을 생성하는 방법

다음 create-channel 예시에서는 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,
        "containerFormat": "TS",
        "name": "test-channel",
        "latencyMode": "LOW",
        "multitrackInputConfiguration": {
            "enabled": false,
            "maximumResolution": "FULL_HD",
            "policy": "ALLOW"
        },
        "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 Low-Latency 사용 설명서의 채널 생성을 참조하세요.

예시 2: ARN에서 지정한 RecordingConfiguration 리소스를 사용하여 레코딩이 활성화된 채널을 생성하는 방법

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",
        "containerFormat": "TS",
        "name": "test-channel-with-recording",
        "latencyMode": "LOW",
        "multitrackInputConfiguration": {
            "enabled": false,
            "maximumResolution": "FULL_HD",
            "policy": "ALLOW"
        },
        "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 Low-Latency 사용 설명서의 Amazon S3에 레코딩을 참조하세요.

예시 3: ARN에서 지정한 재생 제한 정책을 사용하여 채널을 생성하는 방법

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",
        "containerFormat": "TS",
```

```
        "name": "test-channel-with-playback-restriction-policy",
        "latencyMode": "LOW",
        "multitrackInputConfiguration": {
            "enabled": false,
            "maximumResolution": "FULL_HD",
            "policy": "ALLOW"
        },
        "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 Low-Latency 사용 설명서의 원치 않는 콘텐츠 및 뷰어를 참조하세요.

예제 4: 멀티트랙이 활성화된 채널 생성

다음 `create-channel` 예제에서는 스트리밍을 시작하기 위한 새 채널과 연결된 스트림 키를 생성하고 멀티트랙을 활성화합니다.

```
aws ivs create-channel \
    --name 'test-channel' \
    --no-insecure-ingest \
```

```
    --container-format 'FRAGMENTED_MP4' \
    --multitrack-input-configuration '{"enabled": true,"maximumResolution":
 "FULL_HD","policy": "ALLOW"}'
```

출력:

```
{
    "channel": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
        "authorized": false,
        "containerFormat": "FRAGMENTED_MP4",
        "name": "test-channel",
        "latencyMode": "LOW",
        "multitrackInputConfiguration": {
            "enabled": true,
            "maximumResolution": "FULL_HD",
            "policy": "ALLOW"
        },
        "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 Low-Latency 사용 설명서의 채널 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateChannel 섹션을 참조하세요.

create-playback-restriction-policy

다음 코드 예시에서는 create-playback-restriction-policy 코드를 사용하는 방법을 보여줍니다.

AWS CLI

재생 제한 정책 생성

다음 create-playback-restriction-policy 예시에서는 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 Low-Latency 사용 설명서의 원치 않는 콘텐츠 및 뷰어를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreatePlaybackRestrictionPolicy 섹션을 참조하세요.

create-recording-configuration

다음 코드 예시에서는 create-recording-configuration 코드를 사용하는 방법을 보여줍니다.

AWS CLI

RecordingConfiguration 리소스를 생성하는 방법

다음 create-recording-configuration 예시에서는 Amazon S3으로의 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 레코딩을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateRecordingConfiguration 섹션을 참조하세요.

`create-stream-key`

다음 코드 예시에서는 `create-stream-key` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

스트림 키 생성

다음 `create-stream-key` 예시에서는 지정된 Amazon 리소스 이름(ARN)의 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 Low-Latency 사용 설명서의 채널 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateStreamKey 섹션을 참조하세요.

delete-channel

다음 코드 예시에서는 delete-channel 코드를 사용하는 방법을 보여줍니다.

AWS CLI

채널 및 관련 스트림 키를 삭제하는 방법

다음 delete-channel 예시에서는 지정된 Amazon 리소스 이름(ARN)이 있는 채널을 삭제합니다.

```
aws ivs delete-channel \
    --arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 IVS Low-Latency 사용 설명서의 채널 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteChannel 섹션을 참조하세요.

delete-playback-key-pair

다음 코드 예시에서는 delete-playback-key-pair 코드를 사용하는 방법을 보여줍니다.

AWS CLI

지정된 재생 키 페어를 삭제하는 방법

다음 delete-playback-key-pair 예시에서는 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 내용은 Amazon Interactive Video Service 사용자 안내서의 프라이빗 채널 설정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeletePlaybackKeyPair 섹션을 참조하세요.

delete-playback-restriction-policy

다음 코드 예시에서는 delete-playback-restriction-policy 코드를 사용하는 방법을 보여줍니다.

AWS CLI

재생 제한 정책 삭제

다음 delete-playback-restriction-policy 예시에서는 지정된 정책 Amazon 리소스 이름(ARN)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 Low-Latency 사용 설명서의 원치 않는 콘텐츠 및 뷰어를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeletePlaybackRestrictionPolicy 섹션을 참조하세요.

delete-recording-configuration

다음 코드 예시에서는 delete-recording-configuration 코드를 사용하는 방법을 보여줍니다.

AWS CLI

ARN에서 지정한 RecordingConfiguration 리소스를 삭제하는 방법

다음 delete-recording-configuration 예시에서는 지정된 ARN을 사용하여 RecordingConfiguration 리소스를 삭제합니다.

```
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 레코딩을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteRecordingConfiguration 섹션을 참조하세요.

`delete-stream-key`

다음 코드 예시에서는 `delete-stream-key` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

데이터 스트림 삭제

다음 `delete-stream-key` 예시에서는 지정된 Amazon 리소스 이름(ARN)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 Low-Latency 사용 설명서의 [채널 생성](#)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteStreamKey](#) 섹션을 참조하세요.

`get-channel`

다음 코드 예시에서는 `get-channel` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

채널의 구성 정보를 가져오는 방법

다음 `get-channel` 예시에서는 지정된 채널 Amazon 리소스 이름(ARN)의 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",
        "authorized": false,
        "containerFormat": "TS",
        "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",
        "insecureIngest": false,
```

```
        "latencyMode": "LOW",
        "multitrackInputConfiguration": {
            "enabled": false,
            "maximumResolution": "FULL_HD",
            "policy": "ALLOW"
        },
        "name": "channel-1",
        "playbackRestrictionPolicyArn": "",
        "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/api/
video/v1/us-west-2.123456789012.channel.abcdEFGH.m3u8",
        "preset": "",
        "recordingConfigurationArn": "",
        "srt": {
            "endpoint": "a1b2c3d4e5f6.srt.live-video.net",
            "passphrase":
 "AB1C2defGHijkLMNo3PqQRstUvwxyzaBCDEfghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"
        },
        "tags": {}
        "type": "STANDARD",
    }
}
```

자세한 내용은 IVS Low-Latency 사용 설명서의 채널 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetChannel 섹션을 참조하세요.

get-playback-key-pair

다음 코드 예시에서는 get-playback-key-pair 코드를 사용하는 방법을 보여줍니다.

AWS CLI

지정된 재생 키 페어를 가져오는 방법

다음 get-playback-key-pair 예시에서는 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 내용은 Amazon Interactive Video Service 사용자 안내서의 프라이빗 채널 설정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetPlaybackKeyPair 섹션을 참조하세요.

get-playback-restriction-policy

다음 코드 예시에서는 `get-playback-restriction-policy` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

재생 제한 정책의 구성 정보를 가져오는 방법

다음 `get-playback-restriction-policy` 예시에서는 지정된 정책 Amazon 리소스 이름(ARN)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 Low-Latency 사용 설명서의 원치 않는 콘텐츠 및 뷰어를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetPlaybackRestrictionPolicy 섹션을 참조하세요.

get-recording-configuration

다음 코드 예시에서는 `get-recording-configuration` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

RecordingConfiguration 리소스에 대한 정보를 가져오는 방법

다음 `get-recording-configuration` 예시에서는 지정된 ARN의 RecordingConfiguration 리소스 정보를 가져옵니다.

```
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 레코딩을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetRecordingConfiguration 섹션을 참조하세요.

`get-stream-key`

다음 코드 예시에서는 `get-stream-key` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

스트림에 대한 정보 가져오기

다음 `get-stream-key` 예시에서는 지정된 스트림 키의 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 Low-Latency 사용 설명서의 채널 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetStreamKey 섹션을 참조하세요.

get-stream-session

다음 코드 예시에서는 get-stream-session 코드를 사용하는 방법을 보여줍니다.

AWS CLI

지정된 스트림의 메타데이터를 가져오는 방법

다음 get-stream-session 예제에서는 지정된 채널 Amazon 리소스 이름(ARN) 및 지정된 스트림의 메타데이터 구성을 가져옵니다. streamId가 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",
            "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",
            "playbackUrl": "url-string",
            "authorized": false,
            "insecureIngest": false,
            "preset": ""
        },
```

```
        "ingestConfiguration": {
            "audio": {
                "channels": 2,
                "codec": "mp4a.40.2",
                "sampleRate": 8000,
                "targetBitrate": 46875,
                "track": "Track0"
            },
            "video": {
                "avcProfile": "Baseline",
                "avcLevel": "4.2",
                "codec": "avc1.42C02A",
                "encoder": "Lavf58.45.100",
                "level": "4.2",
                "profile": "Baseline",
                "targetBitrate": 8789062,
                "targetFramerate": 60,
                "track": "Track0",
                "videoHeight": 1080,
                "videoWidth": 1920
            }
        },
        "ingestConfigurations": {
            "audioConfigurations": [
                {
                    "channels": 2,
                    "codec": "mp4a.40.2",
                    "sampleRate": 8000,
                    "targetBitrate": 46875,
                    "track": "Track0"
                }
            ],
            "videoConfigurations": [
                {
                    "codec": "avc1.42C02A",
                    "encoder": "Lavf58.45.100",
                    "level": "4.2",
                    "profile": "Baseline",
                    "targetBitrate": 8789062,
                    "targetFramerate": 60,
                    "track": "Track0",
                    "videoHeight": 1080,
                    "videoWidth": 1920
                }
```

```
            ]
        },
        "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": [
            {
                "code": "StreamTakeoverInvalidPriority",
                "name": "Stream Takeover Failure",
                "type": "IVS Stream State Change",
                "eventTime": "2023-06-26T19:09:48+00:00"
            },
            {
                "name": "Stream Takeover",
                "type": "IVS Stream State Change",
                "eventTime": "2023-06-26T19:09:47+00:00"
            },
```

```
            {
                "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 Low-Latency 사용 설명서의 채널 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetStreamSession 섹션을 참조하세요.

get-stream

다음 코드 예시에서는 get-stream 코드를 사용하는 방법을 보여줍니다.

AWS CLI

스트림에 대한 정보 가져오기

다음 get-stream 예시에서는 지정된 채널의 스트림 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 Low-Latency 사용 설명서의 채널 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetStream 섹션을 참조하세요.

`import-playback-key-pair`

다음 코드 예시에서는 `import-playback-key-pair` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

새 키 페어의 퍼블릭 부분을 가져오는 방법

다음 `import-playback-key-pair` 예시에서는 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 내용은 Amazon Interactive Video Service 사용자 안내서의 프라이빗 채널 설정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ImportPlaybackKeyPair 섹션을 참조하세요.

list-channels

다음 코드 예시에서는 `list-channels` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

예시 1: 모든 채널에 대한 요약 정보를 가져오는 방법

다음 `list-channels` 예시에서는 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 Low-Latency 사용 설명서의 채널 생성을 참조하세요.

예시 2: 모든 채널에 대한 요약 정보를 가져오는 방법 지정된 RecordingConfiguration ARN으로 필터링합니다.

다음 `list-channels` 예시에서는 AWS 계정에서 지정된 RecordingConfiguration ARN과 연결된 모든 채널을 나열합니다.

```
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 Low-Latency 사용 설명서의 Amazon S3에 레코딩을 참조하세요.

예시 3: 지정된 PlaybackRestrictionPolicy ARN으로 필터링된 모든 채널에 대한 요약 정보를 가져오는 방법

다음 `list-channels` 예시에서는 AWS 계정에서 지정된 PlaybackRestrictionPolicy ARN과 연결된 모든 채널을 나열합니다.

```
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 Low-Latency 사용 설명서의 원치 않는 콘텐츠 및 뷰어를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListChannels를 참조하세요.

`list-playback-key-pairs`

다음 코드 예시에서는 `list-playback-key-pairs` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

재생 키 페어의 요약 정보 가져오기

다음 `list-playback-key-pairs` 예시에서는 모든 키 페어 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 내용은 Amazon Interactive Video Service 사용자 안내서의 프라이빗 채널 설정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListPlaybackKeyPairs 섹션을 참조하세요.

`list-playback-restriction-policies`

다음 코드 예시에서는 `list-playback-restriction-policies` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

모든 재생 제한 정책에 대한 요약 정보를 가져오는 방법

다음 `list-playback-restriction-policies` 예시에서는 AWS 계정의 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 Low-Latency 사용 설명서의 원치 않는 콘텐츠 및 뷰어를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListPlaybackRestrictionPolicies 섹션을 참조하세요.

list-recording-configurations

다음 코드 예시에서는 `list-recording-configurations` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

이 계정에서 생성된 모든 RecordingConfiguration 리소스를 나열하는 방법

다음 `list-recording-configurations` 예시에서는 계정 내 모든 RecordingConfiguration 리소스의 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 레코딩을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListRecordingConfigurations 섹션을 참조하세요.

list-stream-keys

다음 코드 예시에서는 `list-stream-keys` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

스트림 키 목록을 가져오는 방법

다음 `list-stream-keys` 예시에서는 지정된 Amazon 리소스 이름(ARN)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": {}
        }
    ]
}
```

자세한 내용은 IVS Low-Latency 사용 설명서의 채널 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListStreamKeys 섹션을 참조하세요.

list-stream-sessions

다음 코드 예시에서는 `list-stream-sessions` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

현재 AWS 리전의 지정된 채널에 대한 현재 및 이전 스트림의 요약을 가져오는 방법

다음 `list-stream-sessions` 예시에서는 지정된 채널 Amazon 리소스 이름(ARN)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 Low-Latency 사용 설명서의 채널 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListStreamSessions 섹션을 참조하세요.

list-streams

다음 코드 예시에서는 `list-streams` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

라이브 스트림 목록과 상태를 가져오는 방법

다음 `list-streams` 예시에서는 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 Low-Latency 사용 설명서의 채널 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListStreams를 참조하세요.

list-tags-for-resource

다음 코드 예시에서는 `list-tags-for-resource` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS 리소스의 모든 태그를 나열하는 방법(예: 채널, 스트림 키)

다음 `list-tags-for-resource` 예시에서는 지정된 리소스 Amazon 리소스 이름(ARN)의 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource를 참조하세요.

put-metadata

다음 코드 예시에서는 `put-metadata` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

지정된 채널의 활성 스트림에 메타데이터를 삽입하는 방법

다음 `put-metadata` 예시에서는 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 Low-Latency 사용 설명서의 채널 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutMetadata 섹션을 참조하세요.

`start-viewer-session-revocation`

다음 코드 예시에서는 `start-viewer-session-revocation` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

지정된 다중 채널 ARN 및 뷰어 ID 페어에 대한 뷰어 세션을 취소하는 방법

다음 `start-viewer-session-revocation` 예시에서는 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 내용은 Amazon Interactive Video Service 사용자 안내서의 프라이빗 채널 설정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartViewerSessionRevocation 섹션을 참조하세요.

`stop-stream`

다음 코드 예시에서는 `stop-stream` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

지정된 스트림을 중지하는 방법

다음 `stop-stream` 예시는 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 Low-Latency 사용 설명서의 채널 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StopStream 섹션을 참조하세요.

tag-resource

다음 코드 예시에서는 `tag-resource` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS 리소스(예: 채널, 스트림 키)에 대한 태그를 추가하거나 업데이트하는 방법

다음 `tag-resource` 예시에서는 지정된 리소스 Amazon 리소스 이름(ARN)의 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.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

untag-resource

다음 코드 예시에서는 `untag-resource` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS 리소스의 태그를 제거하는 방법(예: 채널, 스트림 키)

다음 `untag-resource` 예시에서는 지정된 리소스 Amazon 리소스 이름(ARN)의 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.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

update-channel

다음 코드 예시에서는 update-channel 코드를 사용하는 방법을 보여줍니다.

AWS CLI

예시 1: 채널의 구성 정보를 업데이트하는 방법

다음 update-channel 예시에서는 지정된 채널 ARN의 채널 구성을 업데이트하여 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",
        "containerFormat": "TS",
        "multitrackInputConfiguration": {
            "enabled": false,
            "maximumResolution": "FULL_HD",
            "policy": "ALLOW"
        },
        "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 Low-Latency 사용 설명서의 채널 생성을 참조하세요.

예시 2: 채널의 구성을 업데이트하여 레코딩을 활성화하는 방법

다음 update-channel 예시에서는 지정된 채널 ARN의 채널 구성을 업데이트하여 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",
        "containerFormat": "TS",
        "multitrackInputConfiguration": {
            "enabled": false,
            "maximumResolution": "FULL_HD",
            "policy": "ALLOW"
        },
        "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 Low-Latency 사용 설명서의 Amazon S3에 레코딩을 참조하세요.

예시 3: 채널의 구성을 업데이트하여 레코딩을 비활성화하는 방법

다음 `update-channel` 예시에서는 지정된 채널 ARN의 채널 구성을 업데이트하여 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",
        "containerFormat": "TS",
        "multitrackInputConfiguration": {
            "enabled": false,
            "maximumResolution": "FULL_HD",
            "policy": "ALLOW"
        },
        "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 Low-Latency 사용 설명서의 Amazon S3에 레코딩을 참조하세요.

예시 4: 재생 제한을 활성화하도록 채널의 구성을 업데이트하는 방법

다음 `update-channel` 예시에서는 지정된 채널 ARN의 채널 구성을 업데이트하여 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",
        "containerFormat": "TS",
        "multitrackInputConfiguration": {
            "enabled": false,
            "maximumResolution": "FULL_HD",
            "policy": "ALLOW"
        },
        "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 Low-Latency 사용 설명서의 원치 않는 콘텐츠 및 뷰어를 참조하세요.

예시 5: 채널 구성을 업데이트하여 재생 제한을 비활성화하는 방법

다음 update-channel 예시에서는 지정된 채널 ARN의 채널 구성을 업데이트하여 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",
        "containerFormat": "TS",
        "multitrackInputConfiguration": {
            "enabled": false,
            "maximumResolution": "FULL_HD",
            "policy": "ALLOW"
        },
        "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 Low-Latency 사용 설명서의 원치 않는 콘텐츠 및 뷰어를 참조하세요.

예제 6: 채널의 구성을 업데이트하여 멀티트랙을 활성화

다음 `update-channel` 예제에서는 지정된 채널 ARN의 채널 구성을 업데이트하여 멀티트랙을 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' \
    --container-format 'FRAGMENTED_MP4' \
    --multitrack-input-configuration '{"enabled": true,"maximumResolution":
 "FULL_HD","policy": "ALLOW"}'
```

출력:

```
{
    "channel": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
        "containerFormat": "FRAGMENTED_MP4",
        "name": "test-channel-with-multitrack",
        "latencyMode": "LOW",
        "multitrackInputConfiguration": {
            "enabled": true,
            "maximumResolution": "FULL_HD",
            "policy": "ALLOW"
        },
        "type": "STANDARD",
        "playbackRestrictionPolicyArn": "",
        "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 Low-Latency 사용 설명서의 원치 않는 콘텐츠 및 뷰어를 참조하세요.

예제 7: 채널 구성을 업데이트하여 재생 제한을 비활성화

다음 `update-channel` 예제에서는 지정된 채널 ARN의 채널 구성을 업데이트하여 멀티트랙을 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' \
    --container-format 'TS' \
    --multitrack-input-configuration '{"enabled": false}'
```

출력:

```
{
    "channel": {
        ""arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
        "containerFormat": "TS",
        "name": "test-channel-with-multitrack",
        "latencyMode": "LOW",
        "multitrackInputConfiguration": {
            "enabled": false,
            "maximumResolution": "FULL_HD",
            "policy": "ALLOW"
        },
        "type": "STANDARD",
        "playbackRestrictionPolicyArn": "",
        "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 Low-Latency 사용 설명서의 원치 않는 콘텐츠 및 뷰어를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateChannel 섹션을 참조하세요.

update-playback-restriction-policy

다음 코드 예시에서는 update-playback-restriction-policy 코드를 사용하는 방법을 보여줍니다.

AWS CLI

재생 제한 정책 업데이트

다음 update-playback-restriction-policy 예시에서는 재생 제한 정책을 지정된 정책 ARN으로 업데이트하여 엄격한 출처 요구를 비활성화합니다. 이는 이 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 Low-Latency 사용 설명서의 원치 않는 콘텐츠 및 뷰어를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdatePlaybackRestrictionPolicy 섹션을 참조하세요.

AWS CLI를 사용한 Amazon IVS Chat 예시

다음 코드 예시에서는 Amazon IVS Chat과 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`create-chat-token`

다음 코드 예시에서는 `create-chat-token`의 사용 방법을 보여줍니다.

AWS CLI

채팅 토큰 생성

다음 `create-chat-token` 예시에서는 개별 WebSocket을 룸에 연결하는 데 사용되는 암호화된 채팅 토큰을 생성합니다. 토큰은 1분 동안 유효하며 토큰을 통한 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 사용자 안내서의 3단계: 채팅 클라이언트 인증 및 권한 부여를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateChatToken을 참조하세요.

`create-logging-configuration`

다음 코드 예시에서는 `create-logging-configuration`의 사용 방법을 보여줍니다.

AWS CLI

채팅 LoggingConfiguration 리소스 생성

다음 `create-logging-configuration` 예시에서는 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 사용자 안내서의 Amazon IVS Chat 시작하기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateLoggingConfiguration을 참조하세요.

`create-room`

다음 코드 예시에서는 `create-room`의 사용 방법을 보여줍니다.

AWS CLI

룸 생성

다음 `create-room` 예시에서는 새 룸을 만듭니다.

```
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 사용자 안내서의 2단계: 채팅룸 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateRoom을 참조하세요.

delete-logging-configuration

다음 코드 예시에서는 `delete-logging-configuration`의 사용 방법을 보여줍니다.

AWS CLI

채팅 LoggingConfiguration 리소스 삭제

다음 `delete-logging-configuration` 예시에서는 지정된 ARN에 대한 LoggingConfiguration 리소스를 삭제합니다.

```
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 사용자 안내서의 Amazon IVS Chat 시작하기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteLoggingConfiguration을 참조하세요.

delete-message

다음 코드 예시에서는 delete-message의 사용 방법을 보여줍니다.

AWS CLI

지정된 룸에서 메시지 삭제

다음 delete-message 예시에서는 지정된 룸으로 짝수를 보내 클라이언트가 지정된 메시지를 삭제하도록 합니다. 즉, 뷰에서 해당 메시지를 렌더링 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 사용자 안내서의 Amazon IVS Chat 시작하기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteMessage를 참조하세요.

delete-room

다음 코드 예시에서는 delete-room의 사용 방법을 보여줍니다.

AWS CLI

룸 삭제

다음 delete-room 예시에서는 지정된 룸을 삭제합니다. 연결된 클라이언트는 연결 해제됩니다. 성공하면 빈 응답 본문과 함께 HTTP 204가 반환됩니다.

```
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 사용자 안내서의 Amazon IVS Chat 시작하기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteRoom을 참조하세요.

`disconnect-user`

다음 코드 예시에서는 `disconnect-user`의 사용 방법을 보여줍니다.

AWS CLI

사용자를 룸에서 연결 해제

다음 `disconnect-user` 예시에서는 지정된 사용자의 모든 연결을 지정된 룸에서 연결 해제합니다. 성공하면 빈 응답 본문과 함께 HTTP 200이 반환됩니다.

```
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 사용자 안내서의 Amazon IVS Chat 시작하기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisconnectUser를 참조하세요.

`get-logging-configuration`

다음 코드 예시에서는 `get-logging-configuration`의 사용 방법을 보여줍니다.

AWS CLI

LoggingConfiguration 리소스 정보 가져오기

다음 `get-logging-configuration` 예시에서는 지정된 ARN의 LoggingConfiguration 리소스 정보를 가져옵니다.

```
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 사용자 안내서의 Amazon IVS Chat 시작하기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetLoggingConfiguration을 참조하세요.

get-room

다음 코드 예시에서는 `get-room`의 사용 방법을 보여줍니다.

AWS CLI

지정된 룸 가져오기

다음 `get-room` 예시에서는 지정된 룸의 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 사용자 안내서의 Amazon IVS Chat 시작하기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetRoom을 참조하세요.

list-logging-configurations

다음 코드 예시에서는 list-logging-configurations의 사용 방법을 보여줍니다.

AWS CLI

API 요청이 처리되는 AWS 리전의 사용자에 대한 모든 로깅 구성의 요약 정보 가져오기

다음 list-logging-configurations 예시에서는 API 요청이 처리되는 AWS 리전의 사용자에 대한 모든 LoggingConfiguration 리소스의 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 사용자 안내서의 Amazon IVS Chat 시작하기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListLoggingConfigurations를 참조하세요.

list-rooms

다음 코드 예시에서는 `list-rooms`의 사용 방법을 보여줍니다.

AWS CLI

현재 리전의 모든 룸에 대한 요약 정보 가져오기

다음 `list-rooms` 예시에서는 요청이 처리되는 AWS 리전의 모든 룸에 대한 요약 정보를 가져옵니다. 결과는 updateTime의 내림차순으로 정렬됩니다.

```
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 사용자 안내서의 Amazon IVS Chat 시작하기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListRooms를 참조하세요.

list-tags-for-resource

다음 코드 예시에서는 `list-tags-for-resource`의 사용 방법을 보여줍니다.

AWS CLI

AWS 리소스의 모든 태그 나열(예: 룸)

다음 `list-tags-for-resource` 예시에서는 지정된 리소스 Amazon 리소스 이름(ARN)의 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource를 참조하세요.

send-event

다음 코드 예시에서는 send-event의 사용 방법을 보여줍니다.

AWS CLI

룸으로 이벤트 보내기

다음 send-event 예시에서는 지정된 이벤트를 지정된 룸으로 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 사용자 안내서의 Amazon IVS Chat 시작하기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 SendEvent를 참조하세요.

tag-resource

다음 코드 예시에서는 tag-resource의 사용 방법을 보여줍니다.

AWS CLI

AWS 리소스의 태그를 추가하거나 업데이트(예: 룸)

다음 `tag-resource` 예시에서는 지정된 리소스 Amazon 리소스 이름(ARN)의 태그를 추가하거나 업데이트합니다. 성공하면 빈 응답 본문과 함께 HTTP 200이 반환됩니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

`untag-resource`

다음 코드 예시에서는 `untag-resource`의 사용 방법을 보여줍니다.

AWS CLI

AWS 리소스의 태그 제거(예: 룸)

다음 `untag-resource` 예시에서는 지정된 리소스 Amazon 리소스 이름(ARN)의 지정된 태그를 제거합니다. 성공하면 빈 응답 본문과 함께 HTTP 200이 반환됩니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

`update-logging-configuration`

다음 코드 예시에서는 `update-logging-configuration`의 사용 방법을 보여줍니다.

AWS CLI

룸의 로깅 구성 업데이트

다음 `update-logging-configuration` 예시에서는 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 사용자 안내서의 Amazon IVS Chat 시작하기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateLoggingConfiguration을 참조하세요.

`update-room`

다음 코드 예시에서는 `update-room`의 사용 방법을 보여줍니다.

AWS CLI

룸의 구성 업데이트

다음 `update-room` 예시에서는 지정된 룸의 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 사용자 안내서의 Amazon IVS Chat 시작하기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateRoom을 참조하세요.

AWS CLI를 사용한 Amazon IVS Real-Time Streaming 예시

다음 코드 예시에서는 Amazon IVS Real-Time Streaming에서 AWS Command Line Interface를 사용하여 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`create-encoder-configuration`

다음 코드 예시에서는 `create-encoder-configuration` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

구성 인코더 구성을 생성하는 방법

다음 `create-encoder-configuration` 예시에서는 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 사용자 안내서의 Amazon IVS 스트림에서 다중 호스트 활성화를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateEncoderConfiguration 섹션을 참조하세요.

`create-ingest-configuration`

다음 코드 예시는 `create-ingest-configuration`의 사용 방법을 보여 줍니다.

AWS CLI

수집 구성을 생성하려면

다음 `create-ingest-configuration` 예제에서는 RTMPS 프로토콜을 사용하여 수집 구성을 생성합니다.

```
aws ivs-realtime create-ingest-configuration \
    --name ingest1 \
    --ingest-protocol rtmps
```

출력:

```
{
    "ingestConfiguration": {
        "name": "ingest1",
        "arn": "arn:aws:ivs:us-west-2:123456789012:ingest-configuration/
AbCdEfGh1234",
        "ingestProtocol": "RTMPS",
        "streamKey": "rt_123456789012_us-
west-2_AbCdEfGh1234_abcd1234efgh5678ijkl9012MNOP34",
        "stageArn": "",
        "participantId": "xyZ654abC321",
        "state": "INACTIVE",
        "userId": "",
        "tags": {}
    }
}
```

자세한 내용은 Amazon Interactive Video Service 사용 설명서의 IVS Stream Ingest | 실시간 스트리밍을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateIngestConfiguration을 참조하세요.

`create-participant-token`

다음 코드 예시는 `create-participant-token`의 사용 방법을 보여 줍니다.

AWS CLI

스테이지 참가자 토큰 생성

다음 `create-participant-token` 예시에서는 지정된 스테이지의 참가자 토큰을 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 사용자 안내서의 Amazon IVS 스트림에서 다중 호스트 활성화를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateParticipantToken 섹션을 참조하세요.

`create-stage`

다음 코드 예시에서는 `create-stage` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

예시 1: 스테이지 생성

다음 `create-stage` 예시에서는 지정된 사용자의 스테이지 참가자 토큰 및 스테이지를 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",
        "autoParticipantRecordingConfiguration": {
            "storageConfigurationArn": "",
            "mediaTypes": [
                "AUDIO_VIDEO"
            ],
            "thumbnailConfiguration": {
                "targetIntervalSeconds": 60,
                "storage": [
                    "SEQUENTIAL"
                ],
                "recordingMode": "DISABLED"
            },
            "recordingReconnectWindowSeconds": 0,
            "hlsConfiguration": {
                "targetSegmentDurationSeconds": 6
            },
            "recordParticipantReplicas": true
        },
        "endpoints": {
            "events": "wss://global.events.live-video.net",
            "rtmp": "rtmp://9x0y8z7s6t5u.global-contribute-staging.live-video.net/
app/",
            "rtmps": "rtmps://9x0y8z7s6t5u.global-contribute-staging.live-
video.net:443/app/",
            "whip": "https://9x0y8z7s6t5u.global-bm.whip.live-video.net"
        },
        "name": "stage1",
        "tags": {}
    }
}
```

Amazon IVS Low-Latency Streaming 사용 설명서의 Amazon IVS 스트림에서 여러 호스트 활성화를 참조하세요.

예시 2: 스테이지를 생성하고 개별 참가자 레코딩을 구성하는 방법

다음 `create-stage` 예시에서는 스테이지를 생성하고 개별 참가자의 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",
 "recordingReconnectWindowSeconds": 100, \
        "hlsConfiguration": {"targetSegmentDurationSeconds": 5}}'
```

출력:

```
{
    "stage": {
        "activeSessionId": "st-a1b2c3d4e5f6g",
        "arn": "arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh",
        "autoParticipantRecordingConfiguration": {
            "storageConfigurationArn": "arn:aws:ivs:us-west-2:123456789012:storage-
configuration/abcdABCDefgh"
            "mediaTypes": [
                "AUDIO_VIDEO"
            ],
            "thumbnailConfiguration": {
                "targetIntervalSeconds": 60,
                "storage": [
                    "SEQUENTIAL"
                ],
                "recordingMode": "DISABLED"
            },
            "recordingReconnectWindowSeconds": 100,
            "hlsConfiguration": {
                "targetSegmentDurationSeconds": 5
            },
            "recordParticipantReplicas": true
        },
        "endpoints": {
            "events": "wss://global.events.live-video.net",
            "rtmp": "rtmp://9x0y8z7s6t5u.global-contribute-staging.live-video.net/
app/",
            "rtmps": "rtmps://9x0y8z7s6t5u.global-contribute-staging.live-
video.net:443/app/",
            "whip": "https://9x0y8z7s6t5u.global-bm.whip.live-video.net"
        },
        "name": "stage1",
        "tags": {}
```

```
    }
}
```

Amazon IVS Low-Latency Streaming 사용 설명서의 Amazon IVS 스트림에서 여러 호스트 활성화를 참조하세요.

예제 3: 단계를 만들고 썸네일 레코딩을 사용 설정하여 개별 참가자 레코딩 구성

다음 `create-stage` 예제에서는 단계를 만들고 썸네일 레코딩을 사용 설정하여 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", \
        "thumbnailConfiguration": {"recordingMode": "INTERVAL","storage":
 ["SEQUENTIAL"],"targetIntervalSeconds": 60}}'
```

출력:

```
{
    "stage": {
        "activeSessionId": "st-a1b2c3d4e5f6g",
        "arn": "arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh",
        "autoParticipantRecordingConfiguration": {
            "storageConfigurationArn": "arn:aws:ivs:us-west-2:123456789012:storage-
configuration/abcdABCDefgh",
            "mediaTypes": [
                "AUDIO_VIDEO"
            ],
            "thumbnailConfiguration": {
                "targetIntervalSeconds": 60,
                "storage": [
                    "SEQUENTIAL"
                ],
                "recordingMode": "INTERVAL"
            },
            "recordingReconnectWindowSeconds": 0,
            "hlsConfiguration": {
                "targetSegmentDurationSeconds": 6
            },
```

```
            "recordParticipantReplicas": true
        },
        "endpoints": {
            "events": "wss://global.events.live-video.net",
            "rtmp": "rtmp://9x0y8z7s6t5u.global-contribute-staging.live-video.net/
app/",
            "rtmps": "rtmps://9x0y8z7s6t5u.global-contribute-staging.live-
video.net:443/app/",
            "whip": "https://9x0y8z7s6t5u.global-bm.whip.live-video.net"
        },
        "name": "stage1",
        "tags": {}
    }
}
```

Amazon IVS Low-Latency Streaming 사용 설명서의 Amazon IVS 스트림에서 여러 호스트 활성화를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateStage 섹션을 참조하세요.

create-storage-configuration

다음 코드 예시에서는 `create-storage-configuration` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

구성 스토리지 구성을 생성하는 방법

다음 `create-storage-configuration` 예시에서는 지정된 속성을 사용하여 구성 스토리지 구성을 생성합니다.

```
aws ivs-realtime create-storage-configuration \
    --name "test-sc" --s3 "bucketName=amzn-s3-demo-bucket"
```

출력:

```
{
    "storageConfiguration": {
        "arn": "arn:aws:ivs:ap-northeast-1:123456789012:storage-configuration/
ABabCDcdEFef",
        "name": "test-sc",
```

```
            "s3": {
                "bucketName": "amzn-s3-demo-bucket"
            },
            "tags": {}
        }
    }
```

자세한 내용은 Amazon Interactive Video Service 사용자 안내서의 Amazon IVS 스트림에서 다중 호스트 활성화를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateStorageConfiguration 섹션을 참조하세요.

delete-encoder-configuration

다음 코드 예시에서는 delete-encoder-configuration 코드를 사용하는 방법을 보여줍니다.

AWS CLI

구성 인코더 구성을 삭제하는 방법

다음 delete-encoder-configuration은 지정된 Amazon 리소스 이름(ARN)에서 지정한 구성 인코더 구성을 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 사용자 안내서의 Amazon IVS 스트림에서 다중 호스트 활성화를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteEncoderConfiguration 섹션을 참조하세요.

delete-ingest-configuration

다음 코드 예시는 delete-ingest-configuration의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 비활성 수집 구성 삭제

다음 delete-ingest-configuration 예제에서는 지정된 수집 구성 ARN(Amazon Resource Name)에 대한 비활성 수집 구성을 삭제합니다.

```
aws ivs-realtime delete-ingest-configuration \
    --arn arn:aws:ivs:us-west-2:123456789012:ingest-configuration/AbCdEfGh1234
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Interactive Video Service 사용 설명서의 IVS Stream Ingest | 실시간 스트리밍을 참조하세요.

예제 2: 활성 수집 구성을 강제 삭제

다음 delete-ingest-configuration 예제에서는 지정된 수집 구성 ARN(Amazon Resource Name)에 대한 활성 수집 구성을 강제로 삭제합니다.

```
aws ivs-realtime delete-ingest-configuration \
    --arn arn:aws:ivs:us-west-2:123456789012:ingest-configuration/AbCdEfGh1234 \
    --force
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Interactive Video Service 사용 설명서의 IVS Stream Ingest | 실시간 스트리밍을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteIngestConfiguration을 참조하세요.

delete-public-key

다음 코드 예시는 delete-public-key의 사용 방법을 보여 줍니다.

AWS CLI

퍼블릭 키 삭제

다음 delete-public-key는 지정된 퍼블릭 키를 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 IVS Real-Time Streaming 사용자 안내서의 참가자 토큰 배포를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeletePublicKey 섹션을 참조하세요.

delete-stage

다음 코드 예시에서는 delete-stage 코드를 사용하는 방법을 보여줍니다.

AWS CLI

스테이지 삭제

다음 delete-stage 예시에서는 지정된 스테이지를 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 사용자 안내서의 Amazon IVS 스트림에서 다중 호스트 활성화를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteStage 섹션을 참조하세요.

delete-storage-configuration

다음 코드 예시에서는 delete-storage-configuration 코드를 사용하는 방법을 보여줍니다.

AWS CLI

구성 스토리지 구성을 삭제하는 방법

다음 delete-storage-configuration은 지정된 Amazon 리소스 이름(ARN)에서 지정한 구성 스토리지 구성을 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 사용자 안내서의 Amazon IVS 스트림에서 다중 호스트 활성화를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteStorageConfiguration 섹션을 참조하세요.

`disconnect-participant`

다음 코드 예시에서는 `disconnect-participant` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

스테이지 참가자 연결 해제

다음 `disconnect-participant` 예시에서는 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 사용자 안내서의 Amazon IVS 스트림에서 다중 호스트 활성화를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisconnectParticipant 섹션을 참조하세요.

`get-composition`

다음 코드 예시에서는 `get-composition` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

예시 1: 기본 레이아웃 설정을 사용하여 구성을 가져오는 방법

다음 `get-composition` 예시에서는 지정된 Amazon 리소스 이름(ARN)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",
                            "hlsConfiguration": {
                                "targetSegmentDurationSeconds": 2
                            }
                        },
                        "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 IVS Real-Time Streaming 사용 설명서의 IVS 복합 레코딩 | 실시간 스트리밍을 참조하세요.

예시 2: PiP 레이아웃을 사용하여 구성을 가져오는 방법

다음 `get-composition` 예시에서는 PiP 레이아웃을 사용하는 지정된 Amazon 리소스 이름(ARN)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",
                            "hlsConfiguration": {
                                "targetSegmentDurationSeconds": 2
                            }
                        },
                        "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 IVS Real-Time Streaming 사용 설명서의 IVS 복합 레코딩 | 실시간 스트리밍을 참조하세요.

예제 3: 썸네일 레코딩을 사용 설정하여 구성 가져오기

다음 get-composition 예제에서는 지정된 ARN(Amazon 리소스 이름)의 구성을 가져옵니다.이 ARN에는 기본 설정으로 썸네일 레코딩이 사용 설정되어 있습니다.

```
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",
                            "hlsConfiguration": {
                                "targetSegmentDurationSeconds": 2
                            }
                        },
                        "storageConfigurationArn": "arn:arn:aws:ivs:ap-
northeast-1:123456789012:storage-configuration/FefABabCDcdE",
                        "thumbnailConfigurations": [
                           {
                              "targetIntervalSeconds": 60,
                              "storage": [
                                  "SEQUENTIAL"
                              ],
                           }
                        ]
                    }
                },
                "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 IVS Real-Time Streaming 사용 설명서의 IVS 복합 레코딩 | 실시간 스트리밍을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetComposition 섹션을 참조하세요.

`get-encoder-configuration`

다음 코드 예시에서는 `get-encoder-configuration` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

구성 인코더 구성을 가져오는 방법

다음 `get-encoder-configuration` 예시에서는 지정된 Amazon 리소스 이름(ARN)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 사용자 안내서의 Amazon IVS 스트림에서 다중 호스트 활성화를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetEncoderConfiguration 섹션을 참조하세요.

get-ingest-configuration

다음 코드 예시는 get-ingest-configuration의 사용 방법을 보여 줍니다.

AWS CLI

수집 구성 정보를 가져오려면

다음 get-ingest-configuration 예제에서는 지정된 수집 구성 ARN(Amazon Resource Name)에 대한 수집 구성을 가져옵니다.

```
aws ivs-realtime get-ingest-configuration \
    --arn arn:aws:ivs:us-west-2:123456789012:ingest-configuration/AbCdEfGh1234
```

출력:

```
{
    "ingestConfiguration": {
        "name": "ingest1",
        "arn": "arn:aws:ivs:us-west-2:123456789012:ingest-configuration/
AbCdEfGh1234",
        "ingestProtocol": "RTMPS",
        "streamKey": "rt_123456789012_us-
west-2_AbCdEfGh1234_abcd1234efgh5678ijkl9012MNOP34",
        "stageArn": "",
        "participantId": "xyZ654abC321",
        "state": "INACTIVE",
        "userId": "",
        "tags": {}
    }
}
```

자세한 내용은 Amazon Interactive Video Service 사용 설명서의 IVS Stream Ingest | 실시간 스트리밍을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetIngestConfiguration을 참조하세요.

get-participant

다음 코드 예시는 get-participant의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 스테이지 참가자를 가져오려면

다음 get-participant 예시에서는 지정된 스테이지 Amazon 리소스 이름(ARN)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": ""
    }
}
```

Amazon IVS Low-Latency Streaming 사용 설명서의 Amazon IVS 스트림에서 여러 호스트 활성화를 참조하세요.

예제 2: 다른 스테이지에 복제된 스테이지 참가자를 가져오려면

다음 get-participant 예제에서는 참가자가 다른 스테이지에도 복제된 경우, 지정된 스테이지 ARN(Amazon 리소스 이름)에서 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",
        "replicationState": "ACTIVE",
        "replicationType": "SOURCE",
        "sdkVersion": "",
        "state": "CONNECTED",
        "userId": ""
    }
}
```

Amazon IVS Low-Latency Streaming 사용 설명서의 Amazon IVS 스트림에서 여러 호스트 활성화를 참조하세요.

예제 3: 다른 스테이지에서 복제된 스테이지 참가자를 가져오려면

다음 get-participant 예제에서는 참가자가 다른 스테이지에서 복제된 경우, 지정된 스테이지 ARN(Amazon 리소스 이름)에서 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",
        "replicationState": "ACTIVE",
        "replicationType": "REPLICA",
        "sdkVersion": "",
        "state": "CONNECTED",
        "userId": ""
    }
}
```

Amazon IVS Low-Latency Streaming 사용 설명서의 Amazon IVS 스트림에서 여러 호스트 활성화를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetParticipant 섹션을 참조하세요.

get-public-key

다음 코드 예시에서는 get-public-key 코드를 사용하는 방법을 보여줍니다.

AWS CLI

스테이지 참가자 토큰에 서명하는 데 사용되는 기존 퍼블릭 키를 가져오는 방법

다음 `get-public-key` 예시에서는 제공된 ARN에서 지정하고 스테이지 참가자 토큰의 서명에 사용할 퍼블릭 키를 가져옵니다.

```
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 IVS Real-Time Streaming 사용자 안내서의 참가자 토큰 배포를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetPublicKey 섹션을 참조하세요.

get-stage-session

다음 코드 예시에서는 `get-stage-session` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

스테이지 세션을 가져오는 방법

다음 `get-stage-session` 예시에서는 지정된 스테이지 Amazon 리소스 이름(ARN)의 지정된 세션 ID에 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 사용자 안내서의 Amazon IVS 스트림에서 다중 호스트 활성화를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetStageSession 섹션을 참조하세요.

get-stage

다음 코드 예시에서는 `get-stage` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

스테이지의 구성 정보를 가져오는 방법

다음 `get-stage` 예시에서는 지정된 스테이지 Amazon 리소스 이름(ARN)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": {
            "storageConfigurationArn": "",
            "mediaTypes": [
```

```
                "AUDIO_VIDEO"
            ],
            "thumbnailConfiguration": {
                "targetIntervalSeconds": 60,
                "storage": [
                    "SEQUENTIAL"
                ],
                "recordingMode": "DISABLED"
            },
            "recordingReconnectWindowSeconds": 0,
            "hlsConfiguration": {
                "targetSegmentDurationSeconds": 6
            },
            "recordParticipantReplicas": true
        },
        "endpoints": {
            "events": "wss://global.events.live-video.net",
            "rtmp": "rtmp://9x0y8z7s6t5u.global-contribute-staging.live-video.net/
app/",
            "rtmps": "rtmps://9x0y8z7s6t5u.global-contribute-staging.live-
video.net:443/app/",
            "whip": "https://9x0y8z7s6t5u.global-bm.whip.live-video.net"
        },
        "name": "test",
        "tags": {}
    }
}
```

Amazon IVS Low-Latency Streaming 사용 설명서의 Amazon IVS 스트림에서 여러 호스트 활성화를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetStage 섹션을 참조하세요.

get-storage-configuration

다음 코드 예시에서는 `get-storage-configuration` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

구성 스토리지 구성을 가져오는 방법

다음 `get-storage-configuration` 예시에서는 지정된 Amazon 리소스 이름(ARN)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": "amzn-s3-demo-bucket"
        },
        "tags": {}
    }
}
```

자세한 내용은 Amazon Interactive Video Service 사용자 안내서의 Amazon IVS 스트림에서 다중 호스트 활성화를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetStorageConfiguration 섹션을 참조하세요.

`import-public-key`

다음 코드 예시에서는 `import-public-key` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

스테이지 참가자 토큰에 서명하는 데 사용할 기존 퍼블릭 키를 가져오는 방법

다음 `import-public-key` 예시에서는 스테이지 참가자 토큰 서명에 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 IVS Real-Time Streaming 사용자 안내서의 참가자 토큰 배포를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ImportPublicKey 섹션을 참조하세요.

list-compositions

다음 코드 예시에서는 `list-compositions` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

구성 목록을 가져오는 방법

다음 `list-compositions`는 API 요청이 처리되는 AWS 리전 내 AWS 계정의 모든 구성을 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 사용자 안내서의 Amazon IVS 스트림에서 다중 호스트 활성화를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListCompositions 섹션을 참조하세요.

`list-encoder-configurations`

다음 코드 예시에서는 `list-encoder-configurations` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

구성 인코더 구성을 나열하는 방법

다음 `list-encoder-configurations`는 API 요청이 처리되는 AWS 리전 내 AWS 계정의 모든 구성 인코더 구성을 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 사용자 안내서의 [Amazon IVS 스트림에서 다중 호스트 활성화]를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ListEncoderConfigurations] 섹션을 참조하세요.

`list-ingest-configurations`

다음 코드 예시는 `list-ingest-configurations`의 사용 방법을 보여 줍니다.

AWS CLI

모든 수집 구성에 대한 요약 정보를 가져오려면

다음 `list-ingest-configurations` 예제에서는 API 요청이 처리되는 AWS 리전 내 AWS 계정의 모든 수집 구성을 나열합니다.

```
aws ivs-realtime list-ingest-configurations
```

출력:

```
{
    "ingestConfigurations": [
        {
            "name": "",
            "arn": "arn:aws:ivs:us-west-2:123456789012:ingest-configuration/
XYZuvwSt4567",
            "ingestProtocol": "RTMPS",
            "stageArn": "arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh",
            "participnatId": "abC789Xyz456",
            "state": "INACTIVE"
            "userId": "",
        }
    ]
}
```

자세한 내용은 Amazon Interactive Video Service 사용 설명서의 IVS Stream Ingest | 실시간 스트리밍을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListIngestConfigurations 섹션을 참조하세요.

list-participant-events

다음 코드 예시는 `list-participant-events`의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 스테이지 참가자 이벤트 목록을 가져오려면

다음 `list-participant-events` 예시에서는 지정된 스테이지 Amazon 리소스 이름(ARN)의 지정된 참가자 ID 및 세션 ID에 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"
        }
    ]
}
```

Amazon IVS Low-Latency Streaming 사용 설명서의 [Amazon IVS 스트림에서 여러 호스트 활성화](#)를 참조하세요.

예제 2: 참가자 복제 중지 및 시작을 포함한 스테이지 참가자 이벤트 목록을 가져오려면

다음 `list-participant-events` 예제에서는 지정된 스테이지 ARN(Amazon 리소스 이름)의 지정된 세션 ID에 대한 모든 참가자 이벤트를 나열합니다. 여기서 참가자는 다른 스테이지에 복제됩니다.

```
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": "2025-04-26T20:36:28+00:00",
            "name": "LEFT",
            "participantId": "abCDEf12GHIj"
        },
        {
            "eventTime": "2025-04-26T20:36:28+00:00",
            "name": "PUBLISH_STOPPED",
            "participantId": "abCDEf12GHIj"
        },
        {
            "eventTime": "2025-04-26T20:30:34+00:00",
            "name": "JOINED",
            "participantId": "abCDEf12GHIj"
        },
        {
            "eventTime": "2025-04-26T20:30:34+00:00",
            "name": "PUBLISH_STARTED",
            "participantId": "abCDEf12GHIj"
        },
        {
            "name": "REPLICATION_STARTED",
            "participantId": "abCDEf12GHIj",
            "eventTime": "2025-04-26T20:30:34+00:00",
            "destinationStageArn": "arn:aws:ivs:us-west-2:12345678901:stage/
ABCDabcdefgh",
            "destinationSessionId": "st-b1c2d3e4f5g6a"
        },
        {
            "name": "REPLICATION_STOPPED",
            "participantId": "abCDEf12GHIj",
            "eventTime": "2025-04-26T20:32:34+00:00",
            "destinationStageArn": "arn:aws:ivs:us-west-2:12345678901:stage/
ABCDabcdefgh",
            "destinationSessionId": "st-b1c2d3e4f5g6a"
        }
    ]
}
```

Amazon IVS Low-Latency Streaming 사용 설명서의 Amazon IVS 스트림에서 여러 호스트 활성화를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListParticipantEvents 섹션을 참조하세요.

list-participant-replicas

다음 코드 예시는 `list-participant-replicas`의 사용 방법을 보여 줍니다.

AWS CLI

스테이지 참가자 목록을 가져오려면

다음 `list-participant-replicas` 예제에서는 지정된 소스 스테이지 ARN(Amazon 리소스 이름)에서 다른 스테이지로 복제된 모든 스테이지 참가자를 나열합니다.

```
aws ivs-realtime list-participant-replicas \
    --source-stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \
    --participant-id abCDEf12GHIj
```

출력:

```
{
    "replicas": [
        {
            "sourceStageArn": "arn:aws:ivs:us-west-2:123456789012:stage/
abcdABCDefgh",
            "participantId": "abCDEf12GHIj",
            "sourceSessionId": "st-a1b2c3d4e5f6g",
            "destinationStageArn": "arn:aws:ivs:us-west-2:012345678901:stage/
ABCDabcdefgh",
            "destinationSessionId": "st-b1c2d3e4f5g6a",
            "replicationState": "ACTIVE"
        }
    ]
}
```

자세한 내용은 Amazon IVS Real-Time Streaming 사용자 안내서의 IVS Participant Replication을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListParticipantReplicas 섹션을 참조하세요.

list-participants

다음 코드 예시는 `list-participants`의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 스테이지 참가자 목록 가져오기

다음 `list-participants` 예시에서는 지정된 스테이지 Amazon 리소스 이름(ARN)의 지정된 세션 ID에 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": ""
        }
    ]
}
```

Amazon IVS Low-Latency Streaming 사용 설명서의 [Amazon IVS 스트림에서 여러 호스트 활성화]를 참조하세요.

예제 2: 참가자가 다른 스테이지에 복제된 경우 스테이지 참가자 목록 가져오기

다음 `list-participants` 예제에서는 지정된 스테이지 ARN(Amazon 리소스 이름)의 지정된 세션 ID에 대한 모든 참가자를 나열합니다. 여기서 참가자는 다른 스테이지에 복제되었습니다.

```
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": "",
            "replicationState": "ACTIVE",
            "replicationType": "SOURCE",
            "sourceStageArn": "",
            "sourceSessionId": ""
        }
    ]
}
```


Amazon IVS Low-Latency Streaming 사용 설명서의 Amazon IVS 스트림에서 여러 호스트 활성화를 참조하세요.

예제 3: 참가자가 다른 스테이지에서 복제된 경우 스테이지 참가자 목록 가져오기

다음 `list-participants` 예제에서는 지정된 스테이지 ARN(Amazon 리소스 이름)의 지정된 세션 ID에 대한 모든 참가자를 나열합니다. 여기서 참가자는 다른 스테이지에서 복제되었습니다.

```
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": "",
            "replicationState": "ACTIVE",
            "replicationType": "REPLICA",
            "sourceStageArn": "arn:aws:ivs:us-west-2:123456789012:stage/
abcdABCDefgh",
```

```
            "sourceSessionId": "st-a1b2c3d4e5f6g"
        }
    ]
}
```

Amazon IVS Low-Latency Streaming 사용 설명서의 Amazon IVS 스트림에서 여러 호스트 활성화를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListParticipants 섹션을 참조하세요.

`list-public-keys`

다음 코드 예시에서는 `list-public-keys` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

스테이지 참가자 토큰에 서명할 수 있는 기존 퍼블릭 키를 나열하는 방법

다음 `list-public-keys` 예시에서는 API 요청이 처리되는 AWS 리전에서 스테이지 참가자 토큰 서명에 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 IVS Real-Time Streaming 사용자 안내서의 참가자 토큰 배포를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ListPublicKeys](https://docs.aws.amazon.com/cli/latest/reference/ivs-realtime/list-public-keys.html) 섹션을 참조하세요.

`list-stage-sessions`

다음 코드 예시에서는 `list-stage-sessions` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

스테이지 세션 목록을 가져오는 방법

다음 `list-stage-sessions` 예시에서는 지정된 스테이지 Amazon 리소스 이름(ARN)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 사용자 안내서의 Amazon IVS 스트림에서 다중 호스트 활성화를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListStageSessions를 참조하세요.

`list-stages`

다음 코드 예시에서는 `list-stages` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

모든 단계에 대한 요약 정보를 가져오는 방법

다음 `list-stages` 예시에서는 API 요청이 처리되는 AWS 리전 내 AWS 계정의 모든 스테이지를 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 사용자 안내서의 Amazon IVS 스트림에서 다중 호스트 활성화를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListStages 섹션을 참조하세요.

`list-storage-configurations`

다음 코드 예시에서는 `list-storage-configurations` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

구성 스토리지 구성을 나열하는 방법

다음 `list-storage-configurations`는 API 요청이 처리되는 AWS 리전 내 AWS 계정의 모든 구성 스토리지 구성을 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": "amzn-s3-demo-bucket-1"
            },
            "tags": {}
        },
        {
            "arn": "arn:aws:ivs:ap-northeast-1:123456789012:storage-configuration/
ABCefgEFGabc",
            "name": "test-sc-2",
            "s3": {
                "bucketName": "amzn-s3-demo-bucket-2"
            },
            "tags": {}
        }
    ]
}
```

자세한 내용은 Amazon Interactive Video Service 사용자 안내서의 [Amazon IVS 스트림에서 다중 호스트 활성화]를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ListStorageConfigurations] 섹션을 참조하세요.

`start-composition`

다음 코드 예시에서는 `start-composition` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

예시 1: 기본 레이아웃 설정으로 구성을 시작하는 방법

다음 start-composition 예시에서는 지정된 위치로 스트리밍되는 지정된 스테이지의 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"], \
        "recordingConfiguration": {"hlsConfiguration":
 {"targetSegmentDurationSeconds": 5}}, \
        "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",
                            "hlsConfiguration": {
                                "targetSegmentDurationSeconds": 5
                            }
                        },
                        "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 IVS Real-Time Streaming 사용 설명서의 IVS 복합 레코딩 | 실시간 스트리밍을 참조하세요.

예시 2: PiP 레이아웃으로 구성을 시작하는 방법

다음 `start-composition` 예시에서는 PiP 레이아웃을 사용하여 지정된 위치로 스트리밍되는 지정된 스테이지의 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",
                            "hlsConfiguration": {
                                "targetSegmentDurationSeconds": 2
```

```
                        }
                    },
                    "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 IVS Real-Time Streaming 사용 설명서의 IVS 복합 레코딩 | 실시간 스트리밍을 참조하세요.

예시 3: 썸네일 레코딩을 활성화하여 구성 시작

다음 `start-composition` 예제에서는 썸네일 레코딩을 사용 설정하여 지정된 위치로 스트리밍되는 지정된 단계의 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", \
        "thumbnailConfigurations": [{"storage":
 ["SEQUENTIAL"],"targetIntervalSeconds": 60}]}}]'
```

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",
                            "hlsConfiguration": {
```

```
                                "targetSegmentDurationSeconds": 2
                            }
                        },
                        "storageConfigurationArn": "arn:arn:aws:ivs:ap-
northeast-1:123456789012:storage-configuration/FefABabCDcdE",
                        "thumbnailConfigurations": [
                           {
                              "targetIntervalSeconds": 60,
                              "storage": [
                                  "SEQUENTIAL"
                              ]
                           }
                        ]
                    }
                },
                "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 IVS Real-Time Streaming 사용 설명서의 IVS 복합 레코딩 | 실시간 스트리밍을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [StartComposition](https://docs.aws.amazon.com/goto/aws-cli/ivs-realtime-2020-07-14/StartComposition) 섹션을 참조하세요.

`start-participant-replication`

다음 코드 예시는 `start-participant-replication`의 사용 방법을 보여 줍니다.

AWS CLI

한 스테이지에서 다른 스테이지로 참가자 복제를 시작하려면

다음 `start-participant-replication` 예제에서는 각 스테이지가 ARN(Amazon 리소스 이름)별로 지정된 상태에서 소스 스테이지에서 대상 스테이지로 참가자를 복제합니다.

```
aws ivs-realtime start-participant-replication \
    --source-stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \
    --destination-stage-arn arn:aws:ivs:us-west-2:234567890123:stage/bcdABCDefghi \
    --participant-id abCDEf12GHIj
```

출력:

```
{
    "accessControlAllowOrigin": "*",
    "accessControlExposeHeaders": "Access-Control-Allow-Origin,Access-Control-
Expose-Headers,Cache-Control,Content-Length, \
    Content-Security-Policy,Content-Type,date,Strict-Transport-Security,x-amz-apigw-
id,x-amzn-errormessage,x-amzn-errortype, \
    x-amzn-requestid,x-amzn-trace-id,X-Content-Type-Options,X-Frame-Options",
    "cacheControl": "no-store, no-cache",
    "contentSecurityPolicy": "default-src 'self'; upgrade-insecure-requests;",
    "strictTransportSecurity": "max-age:47304000; includeSubDomains",
    "xContentTypeOptions": "nosniff",
    "xFrameOptions": "DENY"
}
```

자세한 내용은 Amazon IVS Real-Time Streaming 사용자 안내서의 [IVS Participant Replication](https://docs.aws.amazon.com/ivs/latest/RealTimeUserGuide/rt-participant-replication.html)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [StartParticipantReplication](https://docs.aws.amazon.com/goto/aws-cli/ivs-realtime-2020-07-14/StartParticipantReplication) 섹션을 참조하세요.

`stop-composition`

다음 코드 예시는 `stop-composition`의 사용 방법을 보여 줍니다.

AWS CLI

구성을 중지하는 방법

다음 `stop-composition`은 지정된 Amazon 리소스 이름(ARN)에서 지정한 구성을 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 사용자 안내서의 [Amazon IVS 스트림에서 다중 호스트 활성화](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [StopComposition](#) 섹션을 참조하세요.

`stop-participant-replication`

다음 코드 예시는 `stop-participant-replication`의 사용 방법을 보여 줍니다.

AWS CLI

한 스테이지에서 다른 스테이지로의 참가자 복제를 중지하려면

다음 `stop-participant-replication` 예제에서는 각 스테이지가 ARN(Amazon 리소스 이름)별로 지정된 상태에서 소스 스테이지에서 대상 스테이지로의 참가자 복제를 중지합니다.

```
aws ivs-realtime stop-participant-replication \
    --source-stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \
    --destination-stage-arn arn:aws:ivs:us-west-2:234567890123:stage/bcdABCDefghi \
    --participant-id abCDEf12GHIj
```

출력:

```
{
    "accessControlAllowOrigin": "*",
    "accessControlExposeHeaders": "Access-Control-Allow-Origin,Access-Control-Expose-Headers,Cache-Control,Content-Length, \
    Content-Security-Policy,Content-Type,date,Strict-Transport-Security,x-amz-apigw-id,x-amzn-errormessage,x-amzn-errortype, \
    x-amzn-requestid,x-amzn-trace-id,X-Content-Type-Options,X-Frame-Options",
```

```
    "cacheControl": "no-store, no-cache",
    "contentSecurityPolicy": "default-src 'self'; upgrade-insecure-requests;",
    "strictTransportSecurity": "max-age:47304000; includeSubDomains",
    "xContentTypeOptions": "nosniff",
    "xFrameOptions": "DENY"
}
```

자세한 내용은 Amazon IVS Real-Time Streaming 사용자 안내서의 IVS Participant Replication을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StopParticipantReplication 섹션을 참조하세요.

update-ingest-configuration

다음 코드 예시는 update-ingest-configuration의 사용 방법을 보여 줍니다.

AWS CLI

수집 구성을 업데이트하려면

다음 update-inegst-configuration 예제에서는 수집 구성을 업데이트하여 스테이지에 연결합니다.

```
aws ivs-realtime update-ingest-configuration \
    --arn arn:aws:ivs:us-west-2:123456789012:ingest-configuration/AbCdEfGh1234 \
    --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh
```

출력:

```
{
    "ingestConfiguration": {
        "name": "ingest1",
        "arn": "arn:aws:ivs:us-west-2:123456789012:ingest-configuration/
AbCdEfGh1234",
        "ingestProtocol": "RTMPS",
        "streamKey": "rt_123456789012_us-
west-2_AbCdEfGh1234_abcd1234efgh5678ijkl9012MNOP34",
        "stageArn": "arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh",
        "participantId": "xyZ654abC321",
        "state": "INACTIVE",
        "userId": "",
        "tags": {}
```

```
    }
}
```

자세한 내용은 Amazon Interactive Video Service 사용 설명서의 IVS Stream Ingest | 실시간 스트리밍을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateIngestConfiguration을 참조하세요.

update-stage

다음 코드 예시는 update-stage의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 스테이지의 구성 업데이트

다음 update-stage 예제에서는 썸네일 레코딩을 사용 설정하여 지정된 단계 ARN의 단계를 업데이트하여 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",
 "recordingReconnectWindowSeconds": 100, \
        "thumbnailConfiguration": {"recordingMode": "INTERVAL","storage":
 ["SEQUENTIAL"],"targetIntervalSeconds": 60}} \
        "hlsConfiguration": {"targetSegmentDurationSeconds": 5}}' \
    --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",
            "thumbnailConfiguration": {
```

```
                "targetIntervalSeconds": 60,
                "storage": [
                    "SEQUENTIAL"
                ],
                "recordingMode": "INTERVAL"
            },
            "recordingReconnectWindowSeconds": 100,
            "hlsConfiguration": {
                "targetSegmentDurationSeconds": 5
            },
            "recordParticipantReplicas": true
        },
        "endpoints": {
            "events": "wss://global.events.live-video.net",
            "rtmp": "rtmp://9x0y8z7s6t5u.global-contribute-staging.live-video.net/
app/",
            "rtmps": "rtmps://9x0y8z7s6t5u.global-contribute-staging.live-
video.net:443/app/",
            "whip": "https://1a2b3c4d5e6f.global-bm.whip.live-video.net"
        },
        "name": "stage1a",
        "tags": {}
    }
}
```

Amazon IVS Low-Latency Streaming 사용 설명서의 Amazon IVS 스트림에서 여러 호스트 활성화를 참조하세요.

예제 2: 참가자 복제본 레코딩 비활성화를 포함하여 스테이지의 구성 업데이트

다음 update-stage 예제에서는 썸네일 레코딩을 활성화하고 참가자 레코딩을 비활성화한 상태로 지정된 스테이지 ARN의 스테이지를 업데이트하여 스테이지 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",
 "recordingReconnectWindowSeconds": 100, \
        "thumbnailConfiguration": {"recordingMode": "INTERVAL","storage":
 ["SEQUENTIAL"],"targetIntervalSeconds": 60}, "recordParticipantReplicas":false} \
        "hlsConfiguration": {"targetSegmentDurationSeconds": 5}}' \
```

```
    --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",
            "thumbnailConfiguration": {
                "targetIntervalSeconds": 60,
                "storage": [
                    "SEQUENTIAL"
                ],
                "recordingMode": "INTERVAL"
            },
            "recordingReconnectWindowSeconds": 100,
            "hlsConfiguration": {
                "targetSegmentDurationSeconds": 5
            },
            "recordParticipantReplicas": false
        },
        "endpoints": {
            "events": "wss://global.events.live-video.net",
            "rtmp": "rtmp://9x0y8z7s6t5u.global-contribute-staging.live-video.net/
app/",
            "rtmps": "rtmps://9x0y8z7s6t5u.global-contribute-staging.live-
video.net:443/app/",
            "whip": "https://1a2b3c4d5e6f.global-bm.whip.live-video.net"
        },
        "name": "stage1a",
        "tags": {}
    }
}
```

Amazon IVS Low-Latency Streaming 사용 설명서의 Amazon IVS 스트림에서 여러 호스트 활성화를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateStage 섹션을 참조하세요.

AWS CLI를 사용한 Amazon Kendra 예시

다음 코드 예시에서는 Amazon Kendra와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`create-data-source`

다음 코드 예시에서는 `create-data-source`의 사용 방법을 보여줍니다.

AWS CLI

Amazon Kendra 데이터 소스 커넥터 생성

다음 `create-data-source`는 Amazon Kendra 데이터 소스 커넥터를 생성하고 구성합니다. `describe-data-source`를 사용하여 데이터 소스 커넥터의 상태를 확인하고, 상태가 데이터 소스 커넥터 '실패'로 표시되는 경우 오류 메시지를 읽고 생성을 완료할 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://amzn-s3-demo-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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateDataSource를 참조하세요.

create-index

다음 코드 예시에서는 create-index의 사용 방법을 보여줍니다.

AWS CLI

Amazon Kendra 인덱스 생성

다음 create-index는 Amazon Kendra 인덱스를 생성하고 구성합니다. describe-index를 사용하여 인덱스의 상태를 확인하고, 상태가 인덱스 '실패'로 표시되는 경우 오류 메시지를 읽고 생성을 완료할 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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateIndex를 참조하세요.

describe-data-source

다음 코드 예시에서는 `describe-data-source`의 사용 방법을 보여줍니다.

AWS CLI

Amazon Kendra 데이터 소스 커넥터의 정보 가져오기

다음 `describe-data-source`는 Amazon Kendra 데이터 소스 커넥터의 정보를 가져옵니다. 데이터 소스 커넥터의 구성을 확인하고, 상태가 데이터 소스 커넥터 '실패'로 표시되는 경우 오류 메시지를 읽고 생성을 완료할 수 있습니다.

```
aws kendra describe-data-source \
    --id exampledatasource1 \
    --index-id exampleindex1
```

출력:

```
{
    "Configuration": {
        "TemplateConfiguration": {
            "Template": {
                "connectionConfiguration": {
                    "repositoryEndpointMetadata": {
                        "BucketName": "amzn-s3-demo-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://amzn-s3-demo-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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDataSource를 참조하세요.

describe-index

다음 코드 예시에서는 describe-index의 사용 방법을 보여줍니다.

AWS CLI

Amazon Kendra 인덱스의 정보 가져오기

다음 describe-index는 Amazon Kendra 인덱스의 정보를 가져옵니다. 인덱스의 구성을 확인하고, 상태가 인덱스 '실패'로 표시되는 경우 오류 메시지를 읽고 생성을 완료할 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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeIndex를 참조하세요.

update-data-source

다음 코드 예시에서는 update-data-source의 사용 방법을 보여줍니다.

AWS CLI

Amazon Kendra 데이터 소스 커넥터 업데이트

다음 update-data-source는 Amazon Kendra 데이터 소스 커넥터의 구성을 업데이트합니다. 작업이 성공하면 서비스가 출력 없음, HTTP 상태 코드 200 또는 AWS CLI 반환 코드 0을 다시 보냅니다. describe-data-source를 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://amzn-s3-demo-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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateDataSource를 참조하세요.

update-index

다음 코드 예시에서는 update-index의 사용 방법을 보여줍니다.

AWS CLI

Amazon Kendra 인덱스 업데이트

다음 update-index는 Amazon Kendra 인덱스의 구성을 업데이트합니다. 작업이 성공하면 서비스가 출력 없음, HTTP 상태 코드 200 또는 AWS CLI 반환 코드 0을 다시 보냅니다. describe-index를 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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateIndex를 참조하세요.

AWS CLI를 사용한 Kinesis 예시

다음 코드 예시는 Kinesis와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

add-tags-to-stream

다음 코드 예시에서는 add-tags-to-stream의 사용 방법을 보여줍니다.

AWS CLI

데이터 스트림에 태그 추가

다음 add-tags-to-stream 예시에서는 키 samplekey 및 값 example이 있는 태그를 지정된 스트림에 할당합니다.

```
aws kinesis add-tags-to-stream \
    --stream-name samplestream \
    --tags samplekey=example
```

이 명령은 출력을 생성하지 않습니다.

자세한 설명은 Amazon Kinesis Data Streams 개발자 안내서의 스트림 태그 지정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AddTagsToStream을 참조하세요.

create-stream

다음 코드 예시에서는 create-stream의 사용 방법을 보여줍니다.

AWS CLI

데이터 스트림 생성

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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateStream을 참조하세요.

decrease-stream-retention-period

다음 코드 예시에서는 `decrease-stream-retention-period`의 사용 방법을 보여줍니다.

AWS CLI

데이터 스트림 보존 기간 줄이기

다음 `decrease-stream-retention-period` 예시에서는 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 설명은 Amazon Kinesis Data Streams 개발자 안내서의 데이터 보존 기간 변경을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DecreaseStreamRetentionPeriod를 참조하세요.

delete-stream

다음 코드 예시에서는 `delete-stream`의 사용 방법을 보여줍니다.

AWS CLI

데이터 스트림 삭제

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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteStream을 참조하세요.

deregister-stream-consumer

다음 코드 예시에서는 `deregister-stream-consumer`의 사용 방법을 보여줍니다.

AWS CLI

데이터 스트림 소비자 등록 취소

다음 `deregister-stream-consumer` 예시에서는 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 설명은 Amazon Kinesis Data Streams 개발자 안내서의 Kinesis Data Streams API를 사용하여 향상된 팬아웃으로 소비자 개발을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeregisterStreamConsumer를 참조하세요.

describe-limits

다음 코드 예시에서는 `describe-limits`의 사용 방법을 보여줍니다.

AWS CLI

샤드 제한 설명

다음 `describe-limits` 예시에서는 현재 AWS 계정의 샤드 제한 및 사용량을 표시합니다.

```
aws kinesis describe-limits
```

출력:

```
{
    "ShardLimit": 500,
    "OpenShardCount": 29
}
```

자세한 설명은 Amazon Kinesis Data Streams 개발자 안내서의 스트림 리샤딩을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeLimits를 참조하세요.

describe-stream-consumer

다음 코드 예시에서는 describe-stream-consumer의 사용 방법을 보여줍니다.

AWS CLI

데이터 스트림 소비자 설명

다음 describe-stream-consumer 예시에서는 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 설명은 Amazon Kinesis Data Streams 개발자 안내서의 Amazon Kinesis Data Streams에서 데이터 읽기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeStreamConsumer를 참조하세요.

describe-stream-summary

다음 코드 예시에서는 describe-stream-summary의 사용 방법을 보여줍니다.

AWS CLI

데이터 스트림 요약 설명

다음 describe-stream-summary 예시에서는 지정된 데이터 스트림에 대한 요약 설명을 제공합니다(샤드 목록 없음).

```
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 설명은 Amazon Kinesis Data Streams 개발자 안내서의 스트림 생성 및 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeStreamSummary를 참조하세요.

describe-stream

다음 코드 예시에서는 describe-stream의 사용 방법을 보여줍니다.

AWS CLI

데이터 스트림 설명

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":
 "49600871682957036442365024926191073437251060580128653314"
                }
            },
            {
                "ShardId": "shardId-000000000001",
                "HashKeyRange": {
                    "StartingHashKey": "113427455640312821154458202477256070485",
                    "EndingHashKey": "226854911280625642308916404954512140969"
                },
                "SequenceNumberRange": {
                    "StartingSequenceNumber":
 "49600871682979337187563555549332609155523708941634633746"
                }
            },
            {
                "ShardId": "shardId-000000000002",
                "HashKeyRange": {
```

```
                    "StartingHashKey": "226854911280625642308916404954512140970",
                    "EndingHashKey": "340282366920938463463374607431768211455"
                },
                "SequenceNumberRange": {
                    "StartingSequenceNumber":
 "49600871683001637932762086172474144873796357303140614178"
                }
            }
        ],
        "StreamARN": "arn:aws:kinesis:us-west-2:123456789012:stream/samplestream",
        "StreamName": "samplestream",
        "StreamStatus": "ACTIVE",
        "RetentionPeriodHours": 24,
        "EnhancedMonitoring": [
            {
                "ShardLevelMetrics": []
            }
        ],
        "EncryptionType": "NONE",
        "KeyId": null,
        "StreamCreationTimestamp": 1572297168.0
    }
}
```

자세한 설명은 Amazon Kinesis Data Streams 개발자 안내서의 스트림 생성 및 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeStream을 참조하세요.

disable-enhanced-monitoring

다음 코드 예시에서는 disable-enhanced-monitoring의 사용 방법을 보여줍니다.

AWS CLI

샤드 수준 지표에 대한 향상된 모니터링 비활성화

다음 disable-enhanced-monitoring 예시에서는 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 설명은 Amazon Kinesis Data Streams 개발자 안내서의 Amazon Kinesis Data Streams의 스트림 모니터링을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisableEnhancedMonitoring을 참조하세요.

enable-enhanced-monitoring

다음 코드 예시에서는 enable-enhanced-monitoring의 사용 방법을 보여줍니다.

AWS CLI

샤드 수준 지표에 대한 향상된 모니터링 활성화

다음 enable-enhanced-monitoring 예시에서는 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 설명은 Amazon Kinesis Data Streams 개발자 안내서의 Amazon Kinesis Data Streams의 스트림 모니터링을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 EnableEnhancedMonitoring을 참조하세요.

get-records

다음 코드 예시에서는 `get-records`의 사용 방법을 보여줍니다.

AWS CLI

샤드에서 레코드 가져오기

다음 `get-records` 예시에서는 지정된 샤드 반복자를 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 Kinesis Data Streams API 및 Java용 AWS SDK를 사용하여 소비자 개발을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetRecords를 참조하세요.

`get-shard-iterator`

다음 코드 예시에서는 `get-shard-iterator`의 사용 방법을 보여줍니다.

AWS CLI

샤드 반복자 가져오기

다음 `get-shard-iterator` 예시에서는 `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
+1UkA9BuN3xvATxMBGxRY3zYK05gqgvaIRn94O8SqeEqwhigwZxNWxID3Ej7YYYcxQi8Q/fIrCjGAy/
n2r5Z9G864YpWDfN9upNNQAR/iiOWKs"
}
```


자세한 내용은 Amazon Kinesis Data Streams 개발자 안내서의 [Kinesis Data Streams API 및 Java용 AWS SDK를 사용하여 소비자 개발]을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [GetShardIterator]를 참조하세요.

`increase-stream-retention-period`

다음 코드 예시에서는 `increase-stream-retention-period`의 사용 방법을 보여줍니다.

AWS CLI

데이터 스트림 보존 기간 늘리기

다음 `increase-stream-retention-period` 예시에서는 지정된 스트림의 보존 기간(데이터 레코드를 스트림에 추가한 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 설명은 Amazon Kinesis Data Streams 개발자 안내서의 데이터 보존 기간 변경을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 IncreaseStreamRetentionPeriod를 참조하세요.

list-shards

다음 코드 예시에서는 `list-shards`의 사용 방법을 보여줍니다.

AWS CLI

데이터 스트림의 샤드 나열

다음 `list-shards` 예시에서는 지정된 스트림에서 지정된 `shardId-000000000000`의 `exclusive-start-shard-id` 뒤에 ID가 바로 오는 샤드부터 시작하여 모든 샤드를 나열합니다.

```
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":
 "49600871682979337187563555549332609155523708941634633746"
            }
        },
```

```
        {
            "ShardId": "shardId-000000000002",
            "HashKeyRange": {
                "StartingHashKey": "226854911280625642308916404954512140970",
                "EndingHashKey": "340282366920938463463374607431768211455"
            },
            "SequenceNumberRange": {
                "StartingSequenceNumber":
 "49600871683001637932762086172474144873796357303140614178"
            }
        }
    ]
}
```

자세한 설명은 Amazon Kinesis Data Streams 개발자 안내서의 샤드 나열을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListShards를 참조하세요.

list-streams

다음 코드 예시에서는 `list-streams`의 사용 방법을 보여줍니다.

AWS CLI

데이터 스트림 나열

다음 `list-streams` 예시에서는 현재 계정 및 리전의 모든 활성 데이터 스트림을 나열합니다.

```
aws kinesis list-streams
```

출력:

```
{
    "StreamNames": [
        "samplestream",
        "samplestream1"
    ]
}
```

자세한 설명은 Amazon Kinesis Data Streams 개발자 안내서의 스트림 나열을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListStreams를 참조하세요.

list-tags-for-stream

다음 코드 예시에서는 list-tags-for-stream의 사용 방법을 보여줍니다.

AWS CLI

데이터 스트림의 태그 나열

다음 list-tags-for-stream 예시에서는 지정된 데이터 스트림에 연결된 태그를 나열합니다.

```
aws kinesis list-tags-for-stream \
    --stream-name samplestream
```

출력:

```
{
    "Tags": [
        {
            "Key": "samplekey",
            "Value": "example"
        }
    ],
    "HasMoreTags": false
}
```

자세한 설명은 Amazon Kinesis Data Streams 개발자 안내서의 스트림 태그 지정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForStream을 참조하세요.

merge-shards

다음 코드 예시에서는 merge-shards의 사용 방법을 보여줍니다.

AWS CLI

샤드 병합

다음 merge-shards 예시에서는 지정된 데이터 스트림에서 ID가 shardId-000000000000 및 shardId-000000000001인 2개의 인접 샤드를 병합하여 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 설명은 Amazon Kinesis Data Streams 개발자 가이드의 2개의 샤드 병합을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 MergeShards를 참조하세요.

put-record

다음 코드 예시에서는 put-record의 사용 방법을 보여줍니다.

AWS CLI

데이터 스트림에 레코드 쓰기

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 Kinesis Data Streams API 및 Java용 AWS SDK를 사용하여 생산자 개발을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutRecord를 참조하세요.

put-records

다음 코드 예시에서는 put-records의 사용 방법을 보여줍니다.

AWS CLI

스트림에 여러 레코드 쓰기

다음 put-records 예시에서는 단일 직접 호출에서 지정된 파티션 키를 사용하여 데이터 레코드를 쓰고 다른 파티션 키를 사용하여 또 하나의 데이터 레코드를 씁니다.

```
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 Kinesis Data Streams API 및 Java용 AWS SDK를 사용하여 생산자 개발을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutRecord를 참조하세요.

register-stream-consumer

다음 코드 예시에서는 register-stream-consumer의 사용 방법을 보여줍니다.

AWS CLI

데이터 스트림 소비자 등록

다음 `register-stream-consumer` 예시에서는 `KinesisConsumerApplication`이라는 소비자를 지정된 데이터 스트림에 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 설명은 Amazon Kinesis Data Streams 개발자 안내서의 Kinesis Data Streams API를 사용하여 향상된 팬아웃으로 소비자 개발을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RegisterStreamConsumer를 참조하세요.

`remove-tags-from-stream`

다음 코드 예시에서는 `remove-tags-from-stream`의 사용 방법을 보여줍니다.

AWS CLI

데이터 스트림에서 태그 제거

다음 `remove-tags-from-stream` 예시에서는 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 설명은 Amazon Kinesis Data Streams 개발자 안내서의 스트림 태그 지정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RemoveTagsFromStream을 참조하세요.

split-shard

다음 코드 예시에서는 split-shard의 사용 방법을 보여줍니다.

AWS CLI

샤드 분할

다음 split-shard 예시에서는 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 설명은 Amazon Kinesis Data Streams 개발자 안내서의 샤드 분할을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 SplitShard를 참조하세요.

start-stream-encryption

다음 코드 예시에서는 start-stream-encryption의 사용 방법을 보여줍니다.

AWS CLI

데이터 스트림 암호화 활성화

다음 start-stream-encryption 예시에서는 지정된 AWS KMS 키를 사용하여 지정된 스트림의 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 설명은 Amazon Kinesis Data Streams 개발자 안내서의 Amazon Kinesis Data Streams의 데이터 보호를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartStreamEncryption을 참조하세요.

stop-stream-encryption

다음 코드 예시에서는 stop-stream-encryption의 사용 방법을 보여줍니다.

AWS CLI

데이터 스트림 암호화 비활성화

다음 stop-stream-encryption 예시에서는 지정된 AWS KMS 키를 사용하여 지정된 스트림의 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 설명은 Amazon Kinesis Data Streams 개발자 안내서의 Amazon Kinesis Data Streams의 데이터 보호를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StopStreamEncryption을 참조하세요.

update-shard-count

다음 코드 예시에서는 update-shard-count의 사용 방법을 보여줍니다.

AWS CLI

데이터 스트림의 샤드 수 업데이트

다음 update-shard-count 예시에서는 지정된 데이터 스트림의 샤드 수를 6으로 업데이트합니다. 이 예시에서는 동일한 크기의 샤드를 생성하는 균일한 스케일링을 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 설명은 Amazon Kinesis Data Streams 개발자 안내서의 스트림 리샤딩을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateShardCount를 참조하세요.

AWS CLI를 사용한 AWS KMS 예시

다음 코드 예시에서는 AWS KMS에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`cancel-key-deletion`

다음 코드 예시에서는 `cancel-key-deletion`의 사용 방법을 보여줍니다.

AWS CLI

고객 관리형 KMS 키의 예약된 삭제 취소

다음 `cancel-key-deletion` 예시에서는 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. 하지만 KMS 키의 키 상태는 `Disabled`이므로 암호화 작업에서 KMS 키를 사용할 수 없습니다. 기능을 복원하는 방법 `enable-key` 명령 을 사용합니다.

자세한 내용은 AWS Key Management Service 개발자 안내서의 [Scheduling and canceling key deletion](#) 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [CancelKeyDeletion](#) 섹션을 참조하세요.

`connect-custom-key-store`

다음 코드 예시에서는 `connect-custom-key-store`의 사용 방법을 보여줍니다.

AWS CLI

사용자 지정 키 저장소 연결

다음 `connect-custom-key-store` 예시에서는 지정된 사용자 지정 키 저장소를 다시 연결합니다. 이와 같은 명령을 사용하여 사용자 지정 키 저장소를 처음 연결하거나 연결이 해제된 키 저장소를 다시 연결할 수 있습니다.

이 명령을 사용하여 AWS CloudHSM 키 저장소 또는 외부 키 저장소를 연결할 수 있습니다.

```
aws kms connect-custom-key-store \
    --custom-key-store-id cks-1234567890abcdef0
```

이 명령은 출력을 반환하지 않습니다. 명령이 적용되었는지 확인하려면 `describe-custom-key-stores` 명령을 사용합니다.

AWS CloudHSM 키 저장소를 연결하는 방법에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 [Connecting and disconnecting an AWS CloudHSM key store](#) 섹션을 참조하세요.

외부 키 저장소를 연결하는 방법에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 Connecting and disconnecting an external key store 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ConnectCustomKeyStore 섹션을 참조하세요.

`create-alias`

다음 코드 예시에서는 `create-alias`의 사용 방법을 보여줍니다.

AWS CLI

KMS 키의 별칭 생성

다음 `create-alias` 명령은 키 ID `1234abcd-12ab-34cd-56ef-1234567890ab`로 식별되는 KMS 키에 대해 `example-alias`라는 별칭을 만듭니다.

별칭 이름은 `alias/`로 시작해야 합니다. `alias/aws`로 시작하는 별칭 이름은 AWS에서 사용하도록 예약되어 있으므로 사용하지 마세요.

```
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 사용합니다.

자세한 내용은 AWS Key Management Service 개발자 안내서의 별칭 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateAlias를 참조하세요.

`create-custom-key-store`

다음 코드 예시에서는 `create-custom-key-store`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: AWS CloudHSM 키 저장소 생성

다음 `create-custom-key-store` 예시에서는 필수 파라미터를 사용하여 AWS CloudHSM 클러스터가 지원하는 AWS CloudHSM 키 저장소를 생성합니다. `custom-key-store-type``parameter with the default value: ``AWS_CLOUDHSM`도 사용할 수 있습니다.

AWS CLI에서 `trust-anchor-certificate` 명령에 대한 파일 입력을 지정하려면 `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 AWS Key Management Service 개발자 안내서의 [Creating an AWS CloudHSM key store](#) 섹션을 참조하세요.

예시 2: 퍼블릭 엔드포인트 연결이 있는 외부 키 저장소 생성

다음 `create-custom-key-store` 예시에서는 인터넷을 통해 AWS KMS와 통신하는 외부 키 저장소(XKS)를 생성합니다.

이 예시에서 `XksProxyUriPath`는 `example-prefix`라는 선택적 접두사를 사용합니다.

참고: AWS CLI 버전 1.0을 사용하는 경우 HTTP 또는 HTTPS 값을 포함하는 파라미터(예: XksProxyUriEndpoint 파라미터)를 지정하기 전에 다음 명령을 실행합니다.

```
aws configure set cli_follow_urlparam false
```

그러지 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 AWS Key Management Service 개발자 안내서의 [Creating an external key store](#) 섹션을 참조하세요.

예시 3: VPC 엔드포인트 서비스 연결이 있는 외부 키 저장소 생성

다음 `create-custom-key-store` 예시에서는 Amazon VPC 엔드포인트 서비스를 사용하여 AWS KMS와 통신하는 외부 키 스토어(XKS)를 생성합니다.

참고: AWS CLI 버전 1.0을 사용하는 경우 HTTP 또는 HTTPS 값을 포함하는 파라미터(예: XksProxyUriEndpoint 파라미터)를 지정하기 전에 다음 명령을 실행합니다.

```
aws configure set cli_follow_urlparam false
```

그러지 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 AWS Key Management Service 개발자 안내서의 [Creating an external key store](#) 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateCustomKeyStore 섹션을 참조하세요.

create-grant

다음 코드 예시에서는 create-grant의 사용 방법을 보여줍니다.

AWS CLI

권한 부여 생성

다음 create-grant 예시에서는 exampleUser 사용자가 1234abcd-12ab-34cd-56ef-1234567890ab 예시 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 사용합니다.

자세한 내용은 AWS Key Management Service 개발자 안내서의 AWS KMS에서의 권한 부여를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateGrant를 참조하세요.

create-key

다음 코드 예시에서는 create-key의 사용 방법을 보여줍니다.

AWS CLI

예시 1: AWS KMS에서 고객 관리형 KMS 키 생성

다음 `create-key` 예시에서는 대칭 암호화 KMS 키를 생성합니다.

대칭 암호화 키인 기본 KMS 키를 생성하기 위해 파라미터를 지정할 필요가 없습니다. 이러한 파라미터의 기본값은 대칭 암호화 키를 생성합니다.

이 명령은 키 정책을 지정하지 않으므로 KMS 키는 프로그래밍 방식으로 만든 KMS 키에 대한 기본 키 정책을 가져옵니다. 키 정책을 보려면 `get-key-policy` 명령을 사용합니다. 키 정책을 변경하려면 `put-key-policy` 명령을 사용합니다.

```
aws kms create-key
```

`create-key` 명령은 새 KMS 키의 키 ID 및 ARN을 포함한 키 메타데이터를 반환합니다. 이 값을 사용하여 다른 AWS KMS 작업에서 KMS 키를 식별할 수 있습니다. 출력에 태그가 포함되지 않습니다. KMS 키에 지정된 태그를 보려면 `list-resource-tags command`를 사용합니다.

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",
        "CurrentKeyMaterialId":
 "0b7fd7ddbac6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6",
        "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: `create-key` 명령으로는 별칭을 지정할 수 없습니다. 새 KMS 키의 별칭을 만들려면 `create-alias` 명령을 사용합니다.

자세한 내용은 AWS Key Management Service 개발자 안내서의 키 생성을 참조하세요.

예시 2: 암호화 및 복호화를 위한 비대칭 RSA KMS 키 생성

다음 `create-key` 예시에서는 암호화 및 복호화를 위한 비대칭 RSA 키 페어가 포함된 KMS 키를 생성합니다. 키가 생성된 후에는 키 사양 및 키 사용량을 변경할 수 없습니다.

```
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 AWS Key Management Service 개발자 안내서의 AWS KMS의 비대칭 키를 참조하세요.

예시 3: 서명 및 확인을 위한 비대칭 타원 곡선 KMS 키 생성

서명 및 확인을 위해 비대칭 타원 곡선(ECC) 키 페어가 포함된 비대칭 KMS 키를 생성합니다. ECC KMS 키에 사용할 수 있는 유일한 값이 SIGN_VERIFY이더라도 --key-usage 파라미터는 필수입니다. 키가 생성된 후에는 키 사양 및 키 사용량을 변경할 수 없습니다.

```
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 AWS Key Management Service 개발자 안내서의 AWS KMS의 비대칭 키를 참조하세요.

예제 4: 서명 및 확인을 위한 비대칭 ML-DSA KMS 키 만들기

이 예제에서는 서명 및 확인을 위한 모듈 격자 디지털 서명 알고리즘(ML-DSA) 키를 만듭니다. ML-DSA 키에 사용할 수 있는 유일한 값이 SIGN_VERIFY라도 key-usage 파라미터는 필수입니다.

```
aws kms create-key \
    --key-spec ML_DSA_65 \
    --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",
        "Description": "",
        "Enabled": true,
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "KeyManager": "CUSTOMER",
        "KeySpec": "ML_DSA_65",
        "KeyState": "Enabled",
        "KeyUsage": "SIGN_VERIFY",
        "MultiRegion": false,
        "Origin": "AWS_KMS",
        "SigningAlgorithms": [
            "ML_DSA_SHAKE_256"
        ]
    }
}
```

자세한 내용은 AWS Key Management Service 개발자 안내서의 AWS KMS의 비대칭 키를 참조하세요.

예제 5: HMAC KMS 키 만들기

다음 create-key 예시에서는 384비트 HMAC KMS 키를 생성합니다. HMAC KMS 키에 사용할 수 있는 유일한 값이더라도 --key-usage 파라미터의 GENERATE_VERIFY_MAC 값은 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 AWS Key Management Service 개발자 안내서의 AWS KMS의 HMAC 키를 참조하세요.

예제 6: 다중 리전 프라이머리 KMS 키 만들기

다음 `create-key` 예시에서는 다중 리전 프라이머리 대칭 암호화 키를 생성합니다. 모든 파라미터의 기본값이 대칭 암호화 키를 생성하므로 이 KMS 키에는 `--multi-region` 파라미터만 필요합니다. AWS CLI에서 부울 파라미터가 true임을 나타내려면 파라미터 이름만 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",
        "CurrentKeyMaterialId":
 "0b7fd7ddbac6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6",
        "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 AWS KMS의 비대칭 키를 참조하세요.

예제 7: 가져온 키 자료에 대한 KMS 키 만들기

다음 `create-key` 예시에서는 키 구성 요소 없이 KMS 키를 생성합니다. 작업이 완료되면 자체 키 구성 요소를 KMS 키로 가져올 수 있습니다. 이 KMS 키를 생성하려면 `--origin` 파라미터를 EXTERNAL로 설정하세요.

```
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 AWS Key Management Service 개발자 안내서의 [AWS KMS 키에서 키 구성 요소 가져오기]를 참조하세요.

예시 6: AWS CloudHSM 키 저장소에 KMS 키 생성

다음 `create-key` 예시에서는 지정된 AWS CloudHSM 키 저장소에 KMS 키를 생성합니다. 이 작업은 AWS KMS에 KMS 키와 해당 메타데이터를 생성하고 사용자 지정 키 저장소와 연결된 AWS CloudHSM 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 AWS CloudHSM 키 저장소를 참조하세요.

예제 8: 외부 키 저장소에 KMS 키 만들기

다음 `create-key` 예시에서는 지정된 외부 키 저장소에 KMS 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 `EXTERNAL_KEY_STORE`여야 합니다. `custom-key-store-id` 파라미터는 외부 키 저장소 프록시에 연결된 외부 키 저장소를 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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateKey를 참조하세요.

decrypt

다음 코드 예시에서는 decrypt의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 대칭 KMS 키를 사용하여 암호화된 메시지 복호화(Linux 및 macOS)

다음 decrypt 명령 예시에서는 AWS CLI를 사용하여 데이터를 복호화하는 권장 방법을 보여줍니다. 이 버전은 대칭 KMS 키로 데이터를 복호화하는 방법을 보여줍니다.

파일에 사이퍼텍스트를 입력합니다. --ciphertext-blob 파라미터 값에는 바이너리 파일에서 데이터를 읽도록 CLI에 지시하는 fileb:// 접두사를 사용합니다. 파일이 현재 디렉터리에 없는 경우 파일의 전체 경로를 입력합니다. 파일에서 AWS CLI 파라미터 값을 읽는 방법에 대한 자세한

내용은 AWS Command Line Interface 사용자 안내서의 파일에서 AWS CLI 파라미터 로드<https://docs.aws.amazon.com/cli/latest/userguide/cli-usage-parameters-file.html> 및 AWS Command Line Tool 블로그의 로컬 파일 파라미터의 모범 사례<https://aws.amazon.com/blogs/developer/best-practices-for-local-file-parameters/>를 참조하세요. 사이퍼텍스트를 복호화할 KMS 키를 지정합니다. 대칭 KMS 키로 복호화할 때는 `--key-id` 파라미터가 필요하지 않습니다. AWS KMS는 사이퍼텍스트 내 메타데이터에서 데이터를 암호화하는 데 사용된 KMS 키의 키 ID를 가져올 수 있습니다. 그러나 사용 중인 KMS 키를 지정하는 것이 항상 좋습니다. 이렇게 하면 의도한 KMS 키를 사용할 수 있으며 신뢰하지 않는 KMS 키를 사용하여 사이퍼텍스트를 실수로 복호화하는 것을 방지할 수 있습니다. 일반 텍스트 출력을 텍스트 값으로 요청하세요. `--query` 파라미터는 CLI에 출력에서 `Plaintext` 필드 값만 가져오도록 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 예시 키 ID를 AWS 계정의 유효한 키 ID로 바꾸세요.

```
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 Key Management Service API 참조의 [암호화 해제](#)를 참조하세요.

예시 2: 대칭 KMS 키를 사용하여 암호화된 메시지 복호화(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 예시 키 ID를 AWS 계정의 유효한 키 ID로 바꾸세요.

```
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 Key Management Service API 참조의 암호화 해제를 참조하세요.

예시 3: 비대칭 KMS 키를 사용하여 암호화된 메시지 복호화(Linux 및 macOS)

다음 decrypt 명령 예시에서는 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 예시 키 ID를 AWS 계정의 유효한 키 ID로 바꾸세요.

```
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 AWS Key Management Service 개발자 안내서의 AWS KMS의 비대칭 키를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 Decrypt를 참조하세요.

delete-alias

다음 코드 예시에서는 delete-alias의 사용 방법을 보여줍니다.

AWS CLI

AWS KMS 별칭 삭제

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 사용합니다.

자세한 내용은 AWS Key Management Service 개발자 안내서의 별칭 삭제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteAlias를 참조하세요.

delete-custom-key-store

다음 코드 예시에서는 delete-custom-key-store의 사용 방법을 보여줍니다.

AWS CLI

사용자 지정 키 저장소 삭제

다음 delete-custom-key-store 예시에서는 지정된 사용자 지정 키 저장소를 삭제합니다.

AWS CloudHSM 키 저장소를 삭제해도 연결된 CloudHSM 클러스터에는 영향을 주지 않습니다. 외부 키 저장소를 삭제해도 연결된 외부 키 저장소 프록시, 외부 키 관리자 또는 외부 키에는 영향을 주지 않습니다.

참고: 사용자 지정 키 저장소를 삭제하려면 먼저 사용자 지정 키 저장소의 모든 KMS 키 삭제를 예약한 다음 해당 KMS 키가 삭제될 때까지 기다려야 합니다. 그런 다음 사용자 지정 키 저장소의 연결을 해제해야 합니다. 사용자 지정 키 저장소에서 KMS 키를 찾는 방법에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 Delete an AWS CloudHSM key store (API) 섹션을 참조하세요.

```
delete-custom-key-store \
    --custom-key-store-id cks-1234567890abcdef0
```

이 명령은 출력을 반환하지 않습니다. 사용자 지정 키 저장소가 삭제되는지 확인하려면 describe-custom-key-stores 명령을 사용합니다.

AWS CloudHSM 키 스토어 삭제에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 Deleting an AWS CloudHSM key store 섹션을 참조하세요.

외부 키 저장소를 삭제하는 방법에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 Deleting an external key store 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteCustomKeyStore 섹션을 참조하세요.

`delete-imported-key-material`

다음 코드 예시에서는 `delete-imported-key-material`의 사용 방법을 보여줍니다.

AWS CLI

KMS 키에서 가져온 키 구성 요소를 삭제하는 방법

다음 `delete-imported-key-material` 예시에서는 KMS 키로 가져온 키 구성 요소를 삭제합니다.

```
aws kms delete-imported-key-material \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

출력:

```
{
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyMaterialId":
 "0b7fd7ddbac6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6"
}
```

자세한 내용은 AWS Key Management Service 개발자 안내서의Deleting imported key material을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteImportedKeyMaterial 섹션을 참조하세요.

`derive-shared-secret`

다음 코드 예시에서는 `derive-shared-secret`의 사용 방법을 보여줍니다.

AWS CLI

공유 보안 암호 파생

다음 derive-shared-secret 예시에서는 키 계약 알고리즘을 사용하여 공유 보안 암호를 도출합니다.

DeriveSharedSecret을 직접적으로 호출하려면 KEY_AGREEMENT의 KeyUsage 값을 가진 비대칭 NIST 권장 타원 곡선(ECC) 또는 SM2(중국 리전만 해당) KMS 키 페어를 사용해야 합니다.

```
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 AWS Key Management Service API 참조의 DeriveSharedSecret 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeriveSharedSecret 섹션을 참조하세요.

describe-custom-key-stores

다음 코드 예시에서는 describe-custom-key-stores의 사용 방법을 보여줍니다.

AWS CLI

예시 1: AWS CloudHSM 키 저장소에 대한 세부 정보를 가져오기

다음 describe-custom-key-store 예시에서는 지정된 AWS CloudHSM 키 저장소의 세부 정보를 표시합니다. 명령은 모든 유형의 사용자 지정 키 저장소에서 동일하지만 출력은 키 저장소 유형에 따라 다르고 외부 키 저장소의 경우 연결 옵션에 따라서도 다릅니다.

기본적으로 이 명령은 계정 및 리전 내 모든 사용자 지정 키 저장소의 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(ConnectionState)를 포함하여 AWS CloudHSM 키 저장소에 대한 유용한 세부 정보가 포함됩니다. 연결 상태가 FAILED인 경우 출력에는 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 AWS Key Management Service 개발자 안내서의 Viewing an AWS CloudHSM key store 섹션을 참조하세요.

예시 2: 퍼블릭 엔드포인트 연결이 있는 외부 키 저장소의 세부 정보 가져오기

다음 describe-custom-key-store 예시에서는 지정된 외부 키 저장소의 세부 정보를 표시합니다. 명령은 모든 유형의 사용자 지정 키 저장소에서 동일하지만 출력은 키 저장소 유형에 따라 다르고 외부 키 저장소의 경우 연결 옵션에 따라서도 다릅니다.

기본적으로 이 명령은 계정 및 리전 내 모든 사용자 지정 키 저장소의 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(ConnectionState)를 포함하여 외부 키 저장소에 대한 유용한 세부 정보가 포함됩니다. 연결 상태가 FAILED인 경우 출력에는 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 AWS Key Management Service 개발자 안내서의 [Viewing an external key store](#) 섹션을 참조하세요.

예시 3: VPC 엔드포인트 서비스 연결이 있는 외부 키 저장소의 세부 정보 가져오기

다음 describe-custom-key-store 예시에서는 지정된 외부 키 저장소의 세부 정보를 표시합니다. 명령은 모든 유형의 사용자 지정 키 저장소에서 동일하지만 출력은 키 저장소 유형에 따라 다르고 외부 키 저장소의 경우 연결 옵션에 따라서도 다릅니다.

기본적으로 이 명령은 계정 및 리전 내 모든 사용자 지정 키 저장소의 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(ConnectionState)를 포함하여 외부 키 저장소에 대한 유용한 세부 정보가 포함됩니다. 연결 상태가 FAILED인 경우 출력에는 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 AWS Key Management Service 개발자 안내서의 Viewing an external key store 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeCustomKeyStores 섹션을 참조하세요.

describe-key

다음 코드 예시에서는 describe-key의 사용 방법을 보여줍니다.

AWS CLI

예시 1: KMS 키의 세부 정보 찾기

다음 describe-key 예시에서는 예시 계정 및 리전에서 Amazon S3의 AWS 관리형 키에 대한 세부 정보를 가져옵니다. 이 명령을 사용하여 AWS 관리형 키 및 고객 관리형 키의 세부 정보를 찾을 수 있습니다.

KMS 키를 지정하려면 key-id 파라미터를 사용합니다. 이 예시에서는 별칭 이름 값을 사용하지만 이 명령에는 키 ID, 키 ARN, 별칭 이름 또는 별칭 ARN을 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",
        "CurrentKeyMaterialId":
 "0b7fd7ddbac6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6",
        "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.

예시 2: RSA 비대칭 KMS 키의 세부 정보 가져오기

다음 `describe-key` 예시에서는 서명 및 확인에 사용되는 비대칭 RSA KMS 키의 세부 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"
        ]
    }
}
```

예시 3: 다중 리전 복제본 키의 세부 정보 가져오기

다음 describe-key 예시에서는 다중 리전 복제본 키의 메타데이터를 가져옵니다. 이 다중 리전 키는 대칭 암호화 키입니다. 모든 다중 리전 키에 대한 describe-key 명령 출력은 프라이머리 키와 모든 해당 복제본의 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",
        "CurrentKeyMaterialId":
 "0b7fd7ddbac6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6",
```

```
        "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"
                }
            ]
        }
    }
}
```

예시 4: HMAC KMS 키의 세부 정보 가져오기

다음 `describe-key` 예시에서는 HMAC KMS 키의 세부 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
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeKey를 참조하세요.

disable-key-rotation

다음 코드 예시에서는 disable-key-rotation의 사용 방법을 보여줍니다.

AWS CLI

KMS 키의 자동 교체를 비활성화하는 방법

다음 disable-key-rotation 예시에서는 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 AWS Key Management Service 개발자 안내서의 Rotating keys 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisableKeyRotation 섹션을 참조하세요.

disable-key

다음 코드 예시에서는 `disable-key`의 사용 방법을 보여줍니다.

AWS CLI

KMS 키를 일시적으로 비활성화

다음 `disable-key` 명령은 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.

- API 세부 정보는 AWS CLI 명령 참조의 DisableKey를 참조하세요.

disconnect-custom-key-store

다음 코드 예시에서는 `disconnect-custom-key-store`의 사용 방법을 보여줍니다.

AWS CLI

사용자 지정 키 저장소 연결 해제

다음 `disconnect-custom-key-store` 예시에서는 AWS CloudHSM 클러스터에서 사용자 지정 키 저장소의 연결을 해제합니다. 문제를 해결하거나 설정을 업데이트하거나 키 저장소의 KMS 키가 암호화 작업에 사용되지 않도록 키 저장소의 연결을 해제할 수 있습니다.

이 명령은 AWS CloudHSM 키 저장소 및 외부 키 저장소를 포함한 모든 사용자 지정 키 저장소에서 동일합니다.

이 명령을 실행하기 앞서 예시에 나온 사용자 지정 키 저장소 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 저장소 연결 해제에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 AWS CloudHSM 키 저장소 연결 및 연결 해제를 참조하세요.

외부 키 저장소 연결 해제에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 외부 키 저장소 연결 및 연결 해제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisconnectCustomKeyStore 섹션을 참조하세요.

`enable-key-rotation`

다음 코드 예시에서는 `enable-key-rotation`의 사용 방법을 보여줍니다.

AWS CLI

KMS 키의 자동 교체 활성화

다음 `enable-key-rotation` 예시에서는 고객 관리형 KMS 키의 자동 교체를 180일의 교체 기간으로 활성화합니다. KMS 키는 이 명령이 완료된 날짜로부터 1년(약 365일) 후, 그 이후에는 매년 교체됩니다.

`--key-id` 파라미터는 KMS 키를 식별합니다. 이 예시에서는 키 ARN 값을 사용하지만 KMS 키의 키 ID 또는 ARN을 사용할 수 있습니다. `--rotation-period-in-days` 파라미터는 각 교체 날짜 사이의 일 수를 지정합니다. 90~2560의 값을 지정하세요. 값을 지정하지 않을 경우 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 활성화되었는지 확인하려면 `get-key-rotation-status` 명령을 사용합니다.

자세한 내용은 AWS Key Management Service 개발자 안내서의 Rotating keys 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 EnableKeyRotation 섹션을 참조하세요.

enable-key

다음 코드 예시에서는 enable-key의 사용 방법을 보여줍니다.

AWS CLI

KMS 키 활성화

다음 enable-key 예시에서는 고객 관리형 키를 활성화합니다. disable-key 명령을 사용하여 일시적으로 비활성화한 KMS 키를 활성화하려면 이와 같은 명령을 사용할 수 있습니다. 또한 삭제 일정이 잡혀 있다가 삭제가 취소되어 사용 중지된 KMS 키를 활성화하는 데도 사용할 수 있습니다.

KMS 키를 지정하려면 key-id 파라미터를 사용합니다. 이 예시에서는 키 ID 값을 사용하지만 이 명령에는 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.

- API 세부 정보는 AWS CLI 명령 참조의 EnableKey를 참조하세요.

encrypt

다음 코드 예시에서는 encrypt의 사용 방법을 보여줍니다.

AWS CLI

예시 1: Linux 또는 macOS에서 파일 콘텐츠 암호화

다음 encrypt 명령은 AWS CLI를 사용하여 데이터를 암호화하는 권장 방법을 보여줍니다.

```
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. `plaintext` 파라미터 값은 base64로 인코딩되거나 AWS CLI에 파일에서 바이너리 데이터를 읽도록 지시하는 `fileb://` 접두사를 사용해야 합니다. 파일이 현재 디렉터리에 없는 경우 파일의 전체 경로를 입력하세요. 예: `fileb:///var/tmp/ExamplePlaintextFile` 또는 `fileb://C:\Temp\ExamplePlaintextFile`. 파일에서 AWS CLI 파라미터 값을 읽는 방법에 대한 자세한 내용은 AWS Command Line Interface 사용자 안내서의 파일에서 파라미터 로드 및 AWS Command Line Tool 블로그의 로컬 파일 매개변수의 모범 사례를 참조하세요. `--output` 및 `--query` 파라미터를 사용하여 명령 출력을 제어합니다. 이러한 파라미터는 명령 출력에서 사이퍼텍스트라는 암호화된 데이터를 추출합니다. 출력을 제어하는 방법에 대한 자세한 내용은 AWS Command Line Interface 사용자 안내서의 명령 출력 제어를 참조하세요. `base64` 유틸리티를 사용하여 추출된 출력을 바이너리 데이터로 디코딩합니다. 성공적인 `encrypt` 명령에서 반환되는 사이퍼텍스트는 base64로 인코딩된 텍스트입니다. AWS CLI를 사용하여 복호화하려면 먼저 이 텍스트를 디코딩해야 합니다. 바이너리 사이퍼텍스트를 파일에 저장합니다. 명령의 마지막 부분(`> ExampleEncryptedFile`)은 복호화를 쉽게 하기 위해 바이너리 사이퍼텍스트를 파일에 저장합니다. AWS CLI를 사용하여 데이터를 복호화하는 예시 명령은 복호화 예시를 참조하세요.

예시 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
```

예 3: 비대칭 KMS 키를 사용한 암호화

다음 encrypt 명령은 비대칭 KMS 키를 사용하여 일반 텍스트를 암호화하는 방법을 보여줍니다. --encryption-algorithm 파라미터가 필요합니다. 모든 encrypt CLI 명령에서와 같이 plaintext 파라미터는 base64로 인코딩되거나 파일에서 바이너리 데이터를 읽도록 AWS CLI에 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 세부 정보는 AWS CLI 명령 참조의 Encrypt를 참조하세요.

generate-data-key-pair-without-plaintext

다음 코드 예시에서는 generate-data-key-pair-without-plaintext의 사용 방법을 보여줍니다.

AWS CLI

ECC NIST P384 비대칭 데이터 키 페어를 생성하는 방법

다음 generate-data-key-pair-without-plaintext 예시에서는 AWS 외부에서 사용할 ECC NIST P384 키 페어를 요청합니다.

명령은 지정된 KMS 키로 암호화된 프라이빗 키의 복사본과 일반 텍스트 퍼블릭 키를 반환합니다. 일반 텍스트 프라이빗 키는 반환하지 않습니다. 암호화된 프라이빗 키를 암호화된 데이터와 함께 안전하게 저장한 후 프라이빗 키를 사용해야 할 때 AWS KMS를 직접적으로 호출하여 복호화할 수 있습니다.

ECC NIST P384 비대칭 데이터 키 페어를 요청하려면 값이 ECC_NIST_P384인 key-pair-spec 파라미터를 사용합니다.

지정하는 KMS 키는 대칭 암호화 KMS 키, 즉 KeySpec 값이 SYMMETRIC_DEFAULT인 KMS 키여야 합니다.

참고: 이 예시의 출력에 있는 값은 잘려서 표시됩니다.

```
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",
    "KeyMaterialId":
 "0b7fd7ddbac6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6",
    "KeyPairSpec": "ECC_NIST_P384"
}
```


`PublicKey` 및 `PrivateKeyCiphertextBlob`은 base64 인코딩 형식으로 반환됩니다.

자세한 내용은 AWS Key Management Service 개발자 안내서의 Data key pairs 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GenerateDataKeyPairWithoutPlaintext 섹션을 참조하세요.

`generate-data-key-pair`

다음 코드 예시에서는 `generate-data-key-pair`의 사용 방법을 보여줍니다.

AWS CLI

2048비트 RSA 비대칭 데이터 키 페어를 생성하는 방법

다음 `generate-data-key-pair` 예시에서는 AWS 외부에서 사용할 2048비트 RSA 비대칭 데이터 키 페어를 요청합니다. 명령은 즉시 사용 및 삭제할 수 있는 일반 텍스트 프라이빗 키와 지정된 KMS 키로 암호화된 프라이빗 키의 복사본을 반환합니다. 암호화된 프라이빗 키를 암호화된 데이터와 함께 안전하게 저장할 수 있습니다.

2048비트 RSA 비대칭 데이터 키 페어를 요청하려면 값이 `RSA_2048`인 `key-pair-spec` 파라미터를 사용합니다.

지정하는 KMS 키는 대칭 암호화 KMS 키, 즉 KeySpec 값이 SYMMETRIC_DEFAULT인 KMS 키여야 합니다.

참고: 이 예시의 출력에 있는 값은 잘려서 표시됩니다.

```
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",
    "KeyMaterialId":
 "0b7fd7ddbac6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6"
    "KeyPairSpec": "RSA_2048"
}
```


PublicKey, PrivateKeyPlaintext 및 PrivateKeyCiphertextBlob는 base64 인코딩 형식으로 반환됩니다.

자세한 내용은 AWS Key Management Service 개발자 안내서의 Data key pairs 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GenerateDataKeyPair 섹션을 참조하세요.

generate-data-key-without-plaintext

다음 코드 예시에서는 generate-data-key-without-plaintext의 사용 방법을 보여줍니다.

AWS CLI

일반 텍스트 키 없이 256비트 대칭 데이터 키 생성

다음 `generate-data-key-without-plaintext` 예시에서는 AWS 외부에서 사용할 256비트 대칭 데이터 키의 암호화된 사본을 요청합니다. 사용할 준비가 되면 AWS KMS를 직접 호출하여 데이터 키를 복호화할 수 있습니다.

256비트 데이터 키를 요청하려면 값이 `AES_256`인 `key-spec` 파라미터를 사용합니다. 128비트 데이터 키를 요청하려면 값이 `AES_128`인 `key-spec` 파라미터를 사용합니다. 다른 모든 데이터 키 길이에는 `number-of-bytes` 파라미터를 사용합니다.

지정하는 KMS 키는 대칭 암호화 KMS 키, 즉 키 사양 값이 SYMMETRIC_DEFAULT인 KMS 키여야 합니다.

```
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",
    "KeyMaterialId":
 "0b7fd7ddbac6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6"
}
```

`CiphertextBlob`(암호화된 데이터 키)은 base64로 인코딩된 형식으로 반환됩니다.

자세한 내용은 AWS Key Management Service 개발자 안내서의 [데이터 키](https://docs.aws.amazon.com/kms/latest/developerguide/concepts.html#data-keys)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [GenerateDataKeyWithoutPlaintext](https://awscli.amazonaws.com/v2/documentation/api/latest/reference/kms/generate-data-key-without-plaintext.html)를 참조하세요.

`generate-data-key`

다음 코드 예시에서는 `generate-data-key`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 256비트 대칭 데이터 키 생성

다음 generate-data-key 예시에서는 AWS 외부에서 사용할 256비트 대칭 데이터 키를 요청합니다. 명령은 즉시 사용 및 삭제할 수 있는 일반 텍스트 데이터 키와 지정된 KMS 키로 암호화된 해당 데이터 키의 사본을 반환합니다. 암호화된 데이터 키를 암호화된 데이터와 함께 안전하게 저장할 수 있습니다.

256비트 데이터 키를 요청하려면 값이 AES_256인 key-spec 파라미터를 사용합니다. 128비트 데이터 키를 요청하려면 값이 AES_128인 key-spec 파라미터를 사용합니다. 다른 모든 데이터 키 길이에는 number-of-bytes 파라미터를 사용합니다.

지정하는 KMS 키는 대칭 암호화 KMS 키, 즉 키 사양 값이 SYMMETRIC_DEFAULT인 KMS 키여야 합니다.

```
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",
    "KeyMaterialId":
 "0b7fd7ddbac6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6",
    "CiphertextBlob":
 "AQEDAHjRYf5WytIc0C857tFSnBaPn2F8DgfmThbJlGfR8P3WlwAAAH4wfAYJKoZIhvcNAQcGoG8wbQIBADBoBgkqhk
+YdhV8MrkBQPeac0ReRVNDt9qleAt+SHgIRF8P0H+7U="
}
```

Plaintext(일반 텍스트 데이터 키) 및 CiphertextBlob(암호화된 데이터 키)은 base64로 인코딩된 형식으로 반환됩니다.

자세한 내용은 AWS Key Management Service 개발자 안내서의 데이터 키를 참조하세요. 예시 2: 512비트 대칭 데이터 키 생성

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 KMS 키로 암호화된 해당 데이터 키의 사본을 반환합니다. 암호화된 데이터 키를 암호화된 데이터와 함께 안전하게 저장할 수 있습니다.

128비트 또는 256비트가 아닌 키 길이를 요청하려면 `number-of-bytes` 파라미터를 사용합니다. 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_DEFAULT인 KMS 키여야 합니다.

참고: 이 예시의 출력에 있는 값은 잘려서 표시됩니다.

```
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",
    "KeyMaterialId":
 "0b7fd7ddbac6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6"
}
```

`Plaintext`(일반 텍스트 데이터 키) 및 `CiphertextBlob`(암호화된 데이터 키)은 base64로 인코딩된 형식으로 반환됩니다.

자세한 내용은 AWS Key Management Service 개발자 안내서의 데이터 키를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GenerateDataKey를 참조하세요.

generate-mac

다음 코드 예시는 `generate-mac`의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 메시지에 대한 HMAC 생성

다음 `generate-mac` 명령은 메시지에 대한 HMAC, HMAC KMS 키 및 MAC 알고리즘을 생성합니다. 알고리즘은 지정된 HMAC KMS 키에서 지원되어야 합니다.

AWS CLI v2에서 `message` 파라미터 값은 Base64로 인코딩되어야 합니다. 또는 메시지를 파일에 저장하고 AWS CLI에 파일에서 바이너리 데이터를 읽도록 지시하는 `fileb://` 접두사를 사용할 수 있습니다.

이 명령을 실행하기 전에 예시 키 ID를 AWS 계정의 유효한 키 ID로 바꾸세요. 키 ID는 키 사용량이 `GENERATE_VERIFY_MAC`인 HMAC KMS 키를 나타내야 합니다.

```
msg=(echo 'Hello World' | base64)

aws kms generate-mac \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --message fileb://Message \
    --mac-algorithm HMAC_SHA_384
```

출력:

```
{
    "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "Mac": "<HMAC_TAG>",
    "MacAlgorithm": "HMAC_SHA_384"
}
```

AWS KMS에서 HMAC KMS 키를 사용하는 방법에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 [HMAC keys in AWS KMS](https://docs.aws.amazon.com/kms/latest/developerguide/hmac.html)를 참조하세요.

예제 2: HMAC를 파일에 저장(Linux 및 macOS)

다음 `generate-mac` 예제에서는 로컬 파일에 저장된 짧은 메시지에 대한 HMAC를 생성합니다. 또한 명령은 응답에서 `Mac` 속성을 가져오고 Base64로 디코딩하여 ExampleMac 파일에 저장합니다. MAC를 확인하는 `verify-mac` 명령에서 MAC 파일을 사용할 수 있습니다.

`generate-mac` 명령에는 Base64로 인코딩된 메시지와 HMAC KMS 키가 지원하는 MAC 알고리즘이 필요합니다. KMS 키가 지원하는 MAC 알고리즘을 가져오려면 `describe-key` 명령을 사용합니다.

이 명령을 실행하기 전에 예시 키 ID를 AWS 계정의 유효한 키 ID로 바꾸세요. 키 ID는 GENERATE_VERIFY_MAC라는 키가 사용된 비대칭 KMS 키를 나타내야 합니다.

```
echo 'hello world' | base64 > EncodedMessage

aws kms generate-mac \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --message fileb://EncodedMessage \
    --mac-algorithm HMAC_SHA_384 \
    --output text \
    --query Mac | base64 --decode > ExampleMac
```

이 명령은 출력을 생성하지 않습니다. 이 예시에서는 출력의 `Mac` 속성을 추출하여 파일에 저장합니다.

AWS KMS에서 HMAC KMS 키를 사용하는 방법에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 [HMAC keys in AWS KMS](https://docs.aws.amazon.com/kms/latest/developerguide/hmac.html)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GenerateMac을 참조하세요.

`generate-random`

다음 코드 예시는 `generate-random`의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 256비트 무작위 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. 무작위 바이트 문자열은 어떤 KMS 키와도 관련이 없습니다.

기본적으로 AWS KMS는 무작위 수를 생성합니다. 그러나 사용자 지정 키 스토어를 지정하면 사용자 지정 키 스토어와 연결된 AWS CloudHSM 클러스터에 무작위 바이트 문자열이 생성됩니다.

이 예시에서는 다음 파라미터와 값을 사용합니다.

값이 32인 필수 `--number-of-bytes` 파라미터를 사용하여 32바이트(256비트) 문자열을 요청합니다. 값이 `text`인 `--output` 파라미터를 사용하여 AWS CLI가 출력을 JSON 대신 텍스트로 반환

하도록 지시합니다. 응답에서 Plaintext 속성 값을 추출하는 데 --query parameter를 사용합니다. base64 유틸리티에 명령 출력을 파이프(|)로 구분하고, 이 유틸리티는 추출된 출력을 디코딩합니다. 리디렉션 연산자(>)를 사용하여 디코딩된 바이트 문자열을 ExampleRandom 파일에 저장합니다. 리디렉션 연산자(>)를 사용 바이너리 사이퍼텍스트를 파일에 저장합니다.

```
aws kms generate-random \
    --number-of-bytes 32 \
    --output text \
    --query Plaintext | base64 --decode > ExampleRandom
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Key Management Service API 참조의 GenerateRandom을 참조하세요.

예시 2: 256비트 무작위 수 생성(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 AWS Key Management Service API 참조의 GenerateRandom을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GenerateRandom을 참조하세요.

get-key-policy

다음 코드 예시에서는 `get-key-policy`의 사용 방법을 보여줍니다.

AWS CLI

한 KMS 키에서 다른 KMS 키로 키 정책 복사

다음 `get-key-policy` 예시에서는 한 KMS 키에서 키 정책을 가져와 텍스트 파일에 저장합니다. 그런 다음 텍스트 파일을 정책 입력으로 사용하여 다른 KMS 키의 정책을 대체합니다.

`put-key-policy`의 `--policy` 파라미터에는 문자열이 필요하므로 출력을 JSON 대신 텍스트 문자열로 반환하려면 `--output text` 옵션을 사용해야 합니다.

```
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 AWS KMS API 참조의 PutKeyPolicy를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetKeyPolicy를 참조하세요.

get-key-rotation-status

다음 코드 예시에서는 `get-key-rotation-status`의 사용 방법을 보여줍니다.

AWS CLI

KMS 키의 교체 상태를 검색하는 방법

다음 `get-key-rotation-status` 예시에서는 자동 교체 활성화 여부, 교체 기간 및 다음 예정된 교체 날짜를 포함하여 지정된 KMS 키의 교체 상태 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 AWS Key Management Service 개발자 안내서의 Rotating keys 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetKeyRotationStatus 섹션을 참조하세요.

`get-parameters-for-import`

다음 코드 예시에서는 `get-parameters-for-import`의 사용 방법을 보여줍니다.

AWS CLI

키 구성 요소를 KMS 키로 가져오는 데 필요한 항목을 가져오는 방법

다음 `get-parameters-for-import` 예시에서는 KMS 키로 키 구성 요소를 가져오는 데 필요한 퍼블릭 키와 가져오기 토큰을 가져옵니다. `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 예시에서는 키 ID 값을 사용하지만 이 명령에는 키 ID 또는 키 ARN을 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 Download the public key and import token 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetParametersForImport 섹션을 참조하세요.

get-public-key

다음 코드 예시에서는 `get-public-key`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 비대칭 KMS 키의 퍼블릭 키를 다운로드

다음 `get-public-key` 예시에서는 비대칭 KMS 키의 퍼블릭 키를 다운로드합니다.

출력에는 퍼블릭 키 반환 외에도 키 사용 및 지원되는 암호화 알고리즘을 포함하여 퍼블릭 키를 AWS KMS 외부에서 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"
    ]
}
```

AWS KMS에서 비대칭 KMS 키를 사용하는 방법에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 AWS KMS에서의 비대칭 키를 참조하세요. 예시 2: 퍼블릭 키를 DER 형식으로 변환(Linux 및 macOS)

다음 `get-public-key` 예시에서는 비대칭 KMS 키의 퍼블릭 키를 다운로드하여 DER 파일에 저장합니다.

AWS CLI에서 `get-public-key` 명령을 사용하면 Base64로 인코딩된 DER 인코딩 X.509 퍼블릭 키를 반환합니다. 이 예시에서는 `PublicKey` 속성 값을 텍스트로 가져옵니다. `PublicKey`를 Base64로 디코딩하고 `public_key.der` 파일에 저장합니다. `output` 파라미터는 출력을 JSON 대신 텍스트로 반환합니다. `--query` 파라미터는 AWS KMS 외부에서 퍼블릭 키를 안전하게 사용하는 데 필요한 속성이 아닌 `PublicKey` 속성만 가져옵니다.

이 명령을 실행하기 전에 예시 키 ID를 AWS 계정의 유효한 키 ID로 바꾸세요.

```
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.

AWS KMS에서 비대칭 KMS 키를 사용하는 방법에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 AWS KMS에서의 비대칭 키를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetPublicKey 섹션을 참조하세요.

`import-key-material`

다음 코드 예시에서는 `import-key-material`의 사용 방법을 보여줍니다.

AWS CLI

키 구성 요소 정보를 KMS 키로 가져오기

다음 `import-key-material` 예시에서는 키 구성 요소 없이 생성된 KMS 키에 키 구성 요소를 업로드합니다. KMS 키의 키 상태가 `PendingImport`여야 합니다.

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 만료되면 AWS KMS는 키 구성 요소를 삭제하고 KMS 키의 키 상태가 `Pending import`로 변경되며 KMS 키가 사용할 수 없게 됩니다. 키를 복원하려면 같은 키 구성 요소를 다시 가져와야 합니다. 다른 키 구성 요소를 사용하려면 새 KMS 키를 생성해야 합니다.

이 명령을 실행하기 전에 예시 키 ID를 AWS 계정의 유효한 키 ID 또는 키 ARN으로 바꾸세요.

```
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
```

출력:

```
{
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyMaterialId":
  "0b7fd7ddbac6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6"
}
```

자세한 내용은 AWS Key Management Service 개발자 안내서의 [키 구성 요소 가져오기](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ImportKeyMaterial](#) 섹션을 참조하세요.

list-aliases

다음 코드 예시에서는 `list-aliases`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: AWS 계정 및 리전의 모든 별칭 나열

다음 예시에서는 `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"
        },
        ...
    ]
}
```

예시 2: 특정 KMS 키의 모든 별칭 나열

다음 예시에서는 `list-aliases` 명령과 해당 `key-id` 파라미터를 사용하여 특정 KMS 키와 연결된 별칭을 모두 나열합니다.

각 별칭은 단 하나의 KMS 키와 연결되지만 KMS 키는 여러 개의 별칭을 가질 수 있습니다. AWS KMS 콘솔에는 각 KMS 키에 대해 별칭이 하나만 나열되므로 이 명령은 매우 유용합니다. KMS 키의 모든 별칭을 찾으려면 `list-aliases` 명령을 사용해야 합니다.

이 예시에서는 `--key-id` 파라미터에 KMS 키의 키 ID를 사용하지만 이 명령에는 키 ID, 키 ARN, 별칭 이름 또는 별칭 ARN을 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 작업]을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ListAliases]를 참조하세요.

`list-grants`

다음 코드 예시에서는 `list-grants`의 사용 방법을 보여줍니다.

AWS CLI

AWS KMS 키에 대한 권한 부여 보기

다음 `list-grants` 예시에서는 계정의 Amazon DynamoDB에 대한 지정된 AWS 관리형 KMS 키에 대한 모든 권한 부여를 표시합니다. 이 권한 부여를 사용하면 DynamoDB가 사용자 대신 KMS 키를 사용하여 DynamoDB 테이블을 디스크에 쓰기 전에 암호화할 수 있습니다. 이와 같은 명령을 사

용하여 AWS 계정 및 리전의 AWS 관리형 KMS 키와 고객 관리형 KMS 키에 대한 권한 부여를 확인할 수 있습니다.

이 명령은 키 ID가 있는 key-id 파라미터를 사용하여 KMS 키를 식별합니다. 키 ID 또는 키 ARN을 사용하여 KMS 키를 식별할 수 있습니다. AWS 관리형 KMS 키의 키 ID 또는 키 ARN을 가져오려면 list-keys 또는 list-aliases 명령을 사용합니다.

```
aws kms list-grants \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

출력은 권한 부여가 Amazon DynamoDB에 KMS 키를 암호화 작업에 사용할 권한을 부여하고 KMS 키(DescribeKey)에 대한 세부 정보를 보고 권한 부여를 사용 중지(RetireGrant)할 수 있는 권한을 부여함을 보여줍니다. 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 AWS KMS에서의 권한 부여를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListGrants를 참조하세요.

`list-key-policies`

다음 코드 예시에서는 `list-key-policies`의 사용 방법을 보여줍니다.

AWS CLI

KMS 키의 키 정책 이름 가져오기

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 키에 대한 키 정책의 이름을 찾을 수 있습니다.

유효한 키 정책 이름은 `default`뿐이므로 이 명령은 유용하지 않습니다.

KMS 키를 지정하려면 `key-id` 파라미터를 사용합니다. 이 예시에서는 키 ID 값을 사용하지만 이 명령에는 키 ID 또는 키 ARN을 사용할 수 있습니다.

```
aws kms list-key-policies \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

출력:

```
{
    "PolicyNames": [
    "default"
    ]
}
```

AWS KMS 키 정책에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 AWS KMS에서 키 정책 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListKeyPolicies를 참조하세요.

list-key-rotations

다음 코드 예시에서는 list-key-rotations의 사용 방법을 보여줍니다.

AWS CLI

완료된 모든 키 재료 교체에 대한 정보를 검색하는 방법

다음 list-key-rotations 예시에서는 지정된 KMS 키에 대해 완료된 모든 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 AWS Key Management Service 개발자 안내서의 Rotating keys 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListKeyRotations 섹션을 참조하세요.

list-keys

다음 코드 예시에서는 list-keys의 사용 방법을 보여줍니다.

AWS CLI

계정 및 리전의 KMS 키 가져오기

다음 `list-keys` 예시에서는 계정과 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListKeys를 참조하세요.

`list-resource-tags`

다음 코드 예시에서는 `list-resource-tags`의 사용 방법을 보여줍니다.

AWS CLI

KMS 키에서 태그를 가져오는 방법

다음 `list-resource-tags` 예시에서는 KMS 키의 태그를 가져옵니다. KMS 키의 태그를 추가 또는 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 예시에서는 키 ID 값을 사용하지만 이 명령에는 키 ID 또는 키 ARN을 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
}
```

AWS KMS에서 태그를 사용하는 방법에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 키 태그 지정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListResourceTags 섹션을 참조하세요.

list-retirable-grants

다음 코드 예시에서는 list-retirable-grants의 사용 방법을 보여줍니다.

AWS CLI

보안 주체가 사용할 수 있는 권한 부여를 보는 방법

다음 list-retirable-grants 예시에서는 ExampleAdmin 사용자가 AWS 계정 및 리전의 KMS 키에서 사용 중지할 수 있는 모든 권한 부여를 표시합니다. 이와 같은 명령을 사용하여 계정 위탁자가 AWS 계정 및 리전의 KMS 키에서 사용 중지할 수 있는 권한 부여를 볼 수 있습니다.

필수 retiring-principal 파라미터의 값은 계정, 사용자 또는 역할의 Amazon 리소스 이름(ARN)이어야 합니다.

서비스가 사용 중지 위탁자일 수 있더라도 이 명령에서 retiring-principal의 값에 대한 서비스를 지정할 수 없습니다. 특정 서비스가 사용 중지 보안 주체인 권한을 찾으려면 list-grants 명령을 사용합니다.

출력에 따르면 ExampleAdmin 사용자에게는 계정 및 리전의 서로 다른 두 KMS 키에 대한 권한 부여를 사용 중지할 수 있는 권한이 있습니다. 사용 중지 위탁자 외에도 계정에는 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 AWS KMS에서의 권한 부여를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListRetirableGrants 섹션을 참조하세요.

put-key-policy

다음 코드 예시에서는 put-key-policy의 사용 방법을 보여줍니다.

AWS CLI

KMS 키의 키 정책 변경

다음 put-key-policy 예시에서는 고객 관리형 키의 키 정책을 변경합니다.

시작하려면 키 정책을 생성하고 로컬 JSON 파일로 저장합니다. 이 예시에서 파일은 key_policy.json입니다. 키 정책을 policy 파라미터의 문자열 값으로 지정할 수도 있습니다.

이 키 정책의 첫 번째 문은 IAM 정책을 사용하여 KMS 키에 대한 액세스를 제어할 권한을 AWS 계정에 부여합니다. 두 번째 문은 test-user 사용자에게 KMS 키에서 describe-key 및 list-keys 명령을 실행할 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 예시에서는 키 ID를 사용하지만 키 ARN을 사용할 수도 있습니다. 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 `default`인 `policy-name` 파라미터를 사용합니다. 정책 이름을 지정하지 않을 경우 기본값은 `default`입니다. 유일한 유효 값은 `default`입니다.

```
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 적용되었는지 확인하려면 `get-key-policy` 명령을 사용합니다. 다음 예시 명령은 동일한 KMS 키에 대한 키 정책을 가져옵니다. 값이 `text`인 `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.

- API 세부 정보는 AWS CLI 명령 참조의 PutKeyPolicy를 참조하세요.

re-encrypt

다음 코드 예시에서는 re-encrypt의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 다른 대칭 KMS 키를 사용하여 암호화된 메시지 다시 암호화(Linux 및 macOS)

다음 re-encrypt 명령 예시에서는 AWS CLI를 사용하여 데이터를 다시 암호화하는 권장 방법을 보여줍니다.

파일에 사이퍼텍스트를 입력합니다. `--ciphertext-blob` 파라미터 값에는 바이너리 파일에서 데이터를 읽도록 CLI에 지시하는 `fileb://` 접두사를 사용합니다. 파일이 현재 디렉터리에 없는 경우 파일의 전체 경로를 입력합니다. 파일에서 AWS CLI 파라미터 값을 읽는 방법에 대한 자세한 내용은 AWS Command Line Interface 사용 설명서의 Loading AWS CLI parameters from a file 및 AWS Command Line Tool Blog의 Best Practices for Local File Parameters를 참조하세요. 사이퍼텍스트를 복호화하는 소스 KMS 키를 지정하세요. 대칭 암호화 KMS 키를 사용하여 복호화할 때는 `--source-key-id` 파라미터가 필요하지 않습니다. AWS KMS는 사이퍼텍스트 블롭 내 메타데이터에서 데이터를 암호화하는 데 사용된 KMS 키를 가져올 수 있습니다. 그러나 사용 중인 KMS 키를 지정하는 것이 항상 좋습니다. 이렇게 하면 의도한 KMS 키를 사용할 수 있으며 신뢰하지 않는 KMS 키를 사용하여 사이퍼텍스트를 실수로 복호화하는 것을 방지할 수 있습니다. 데이터를 다시 암호화하는 대상 KMS 키를 지정하세요. `--destination-key-id` 파라미터는 항상 필요합니다. 이 예시에서는 키 ARN을 사용하지만 모든 유효한 키 식별자를 사용할 수 있습니다. 일반 텍스트 출력을 텍스트 값으로 요청하세요. `--query` 파라미터는 출력에서 `Plaintext` 필드 값만 가져오도록 CLI에 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 예시 키 ID를 AWS 계정의 유효한 키 ID로 바꾸세요.

```
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 ReEncrypt를 참조하세요.

예시 2: 다른 대칭 KMS 키를 사용하여 암호화된 메시지 다시 암호화(Windows 명령 프롬프트)

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 예시 키 ID를 AWS 계정의 유효한 키 ID로 바꾸세요.

```
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 사용합니다.

```
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 ReEncrypt를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ReEncrypt를 참조하세요.

`retire-grant`

다음 코드 예시에서는 `retire-grant`의 사용 방법을 보여줍니다.

AWS CLI

고객 마스터 키에 대한 권한 부여 사용 중지

다음 `retire-grant` 예시에서는 KMS 키에서 권한 부여를 삭제합니다.

다음 예시 명령은 `grant-id` 및 `key-id` 파라미터를 지정합니다. `key-id` 파라미터 값은 KMS 키의 키 ARN이어야 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 사용합니다.

자세한 내용은 AWS Key Management Service 개발자 안내서의 권한 부여 사용 중지 및 취소를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RetireGrant를 참조하세요.

revoke-grant

다음 코드 예시에서는 revoke-grant의 사용 방법을 보여줍니다.

AWS CLI

고객 마스터 키에 대한 권한 부여 사용 중지

다음 revoke-grant 예시에서는 KMS 키에서 권한 부여를 삭제합니다. 다음 예시 명령은 grant-id 및 key-id 파라미터를 지정합니다. key-id 파라미터 값은 KMS 키의 키 ID 또는 키 ARN일 수 있습니다.

```
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 사용합니다.

자세한 내용은 AWS Key Management Service 개발자 안내서의 권한 부여 사용 중지 및 취소를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RevokeGrant를 참조하세요.

rotate-key-on-demand

다음 코드 예시에서는 rotate-key-on-demand의 사용 방법을 보여줍니다.

AWS CLI

KMS 키의 온디맨드 교체를 수행하는 방법

다음 rotate-key-on-demand 예시에서는 지정된 KMS 키의 키 구성 요소를 즉시 교체하기 시작합니다.

```
aws kms rotate-key-on-demand \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

출력:

```
{
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
}
```

자세한 내용은 AWS Key Management Service 개발자 안내서의 How to perform on-demand key rotation 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RotateKeyOnDemand 섹션을 참조하세요.

`schedule-key-deletion`

다음 코드 예시에서는 `schedule-key-deletion`의 사용 방법을 보여줍니다.

AWS CLI

고객 관리형 KMS 키 삭제 예약

다음 `schedule-key-deletion` 예시에서는 지정된 고객 관리형 KMS 키가 15일 후에 삭제되도록 예약합니다.

`--key-id` 파라미터는 KMS 키를 식별합니다. 이 예시에서는 키 ARN 값을 사용하지만 KMS 키의 키 ID 또는 ARN을 사용할 수 있습니다. `--pending-window-in-days` 파라미터는 7~30일의 대기 기간을 지정합니다. 기본 대기 기간은 30일입니다. 이 예시에서는 값을 15로 지정합니다. 이 값은 명령이 완료되고 15일 후에 KMS 키를 영구 삭제하도록 AWS에 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`), 삭제 날짜(Unix 시간)이 포함됩니다. 삭제 날짜를 현지 시간으로 보려면 AWS KMS 콘솔을 사용합니다. `PendingDeletion` 키 상태의 KMS 키는 암호화 작업에 사용될 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.

- API 세부 정보는 AWS CLI 명령 참조의 ScheduleKeyDeletion을 참조하세요.

sign

다음 코드 예시에서는 sign의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 메시지에 대한 디지털 서명을 생성하는 방법

다음 sign 예시에서는 짧은 메시지에 대한 암호화 서명을 생성합니다. 명령 출력에는 verify 명령을 사용하여 확인할 수 있는 base-64로 인코딩된 Signature 필드가 포함됩니다.

서명할 메시지와 비대칭 KMS 키가 지원하는 서명 알고리즘을 지정해야 합니다. KMS 키의 서명 알고리즘을 가져오려면 describe-key 명령을 사용합니다.

AWS CLI v2에서 message 파라미터 값은 Base64로 인코딩되어야 합니다. 또는 메시지를 파일에 저장하고 AWS CLI에 파일에서 바이너리 데이터를 읽도록 지시하는 fileb:// 접두사를 사용할 수 있습니다.

이 명령을 실행하기 전에 예시 키 ID를 AWS 계정의 유효한 키 ID로 바꾸세요. 키 ID는 SIGN_VERIFY라는 키가 사용된 비대칭 KMS 키를 나타내야 합니다.

```
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"
}
```

AWS KMS에서 비대칭 KMS 키를 사용하는 방법에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 AWS KMS에서의 비대칭 키를 참조하세요.

예시 2: 디지털 서명을 파일에 저장(Linux 및 macOS)

다음 `sign` 예시에서는 로컬 파일에 저장된 짧은 메시지에 대한 암호화 서명을 생성합니다. 또한 명령은 응답에서 `Signature` 속성을 가져오고 Base64로 디코딩하여 ExampleSignature 파일에 저장합니다. 서명을 확인하는 `verify` 명령에서 서명 파일을 사용할 수 있습니다.

`sign` 명령에는 Base64로 인코딩된 메시지와 비대칭 KMS 키가 지원하는 서명 알고리즘이 필요합니다. KMS 키가 지원하는 서명 알고리즘을 가져오려면 `describe-key` 명령을 사용합니다.

이 명령을 실행하기 전에 예시 키 ID를 AWS 계정의 유효한 키 ID로 바꾸세요. 키 ID는 SIGN_VERIFY라는 키가 사용된 비대칭 KMS 키를 나타내야 합니다.

```
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 예시에서는 출력의 `Signature` 속성을 추출하여 파일에 저장합니다.

AWS KMS에서 비대칭 KMS 키를 사용하는 방법에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 AWS KMS에서의 비대칭 키를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 Sign 섹션을 참조하세요.

`tag-resource`

다음 코드 예시에서는 `tag-resource`의 사용 방법을 보여줍니다.

AWS CLI

KMS 키에 태그를 추가하는 방법

다음 `tag-resource` 예시에서는 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 예시에서는 키 ID 값을 사용하지만 이 명령에는 키 ID 또는 키 ARN을 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. AWS KMS 키의 태그를 보려면 `list-resource-tags` 명령을 사용합니다.

AWS KMS에서 태그를 사용하는 방법에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 키 태그 지정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

`untag-resource`

다음 코드 예시에서는 `untag-resource`의 사용 방법을 보여줍니다.

AWS CLI

KMS 키에서 태그 삭제

다음 `untag-resource` 예시에서는 고객 관리형 KMS 키에서 `"Purpose"` 키가 있는 태그를 삭제합니다.

KMS 키를 지정하려면 `key-id` 파라미터를 사용합니다. 이 예시에서는 키 ID 값을 사용하지만 이 명령에는 키 ID 또는 키 ARN을 사용할 수 있습니다. 이 명령을 실행하기 전에 예시 키 ID를 AWS 계정의 유효한 키 ID로 바꾸세요.

```
aws kms untag-resource \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
```

```
    --tag-key 'Purpose'
```

이 명령은 출력을 생성하지 않습니다. AWS KMS 키의 태그를 보려면 `list-resource-tags` 명령을 사용합니다.

AWS KMS에서 태그를 사용하는 방법에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 키 태그 지정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

`update-alias`

다음 코드 예시에서는 `update-alias`의 사용 방법을 보여줍니다.

AWS CLI

별칭을 다른 KMS 키에 연결

다음 `update-alias` 예시에서는 별칭 `alias/test-key`를 다른 KMS 키와 연결합니다.

`--alias-name` 파라미터는 별칭을 지정합니다. 별칭 이름 값은 `alias/`로 시작해야 합니다. `--target-key-id` 파라미터는 별칭과 연결할 KMS 키를 지정합니다. 별칭에 대한 현재 KMS 키를 지정할 필요는 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 사용합니다.

자세한 내용은 AWS Key Management Service 개발자 안내서의 별칭 업데이트를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateAlias를 참조하세요.

`update-custom-key-store`

다음 코드 예시에서는 `update-custom-key-store`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 사용자 지정 키 저장소의 표시 이름 편집

다음 update-custom-key-store 예시에서는 사용자 지정 키 저장소의 이름을 변경합니다. 이 예시는 AWS CloudHSM 키 저장소 또는 외부 키 저장소에서 작동합니다.

custom-key-store-id를 사용하여 키 저장소를 식별합니다. new-custom-key-store-name 파라미터를 사용하여 새로운 기억하기 쉬운 이름을 지정합니다.

AWS CloudHSM 키 스토어의 표시 이름을 업데이트하려면 먼저 disconnect-custom-key-store 명령을 사용하여 키 스토어의 연결을 해제해야 합니다. 외부 키 저장소가 연결되거나 연결 해제된 상태에서 해당 저장소의 기억하기 쉬운 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 출력을 반환하지 않습니다. 명령이 작동하는지 확인하려면 describe-custom-key-stores 명령을 사용합니다.

AWS CloudHSM 키 저장소를 업데이트하는 방법에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 AWS CloudHSM 키 저장소 설정 편집을 참조하세요.

외부 키 저장소를 업데이트하는 방법에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 외부 키 저장소 속성 편집을 참조하세요.

예시 2: AWS CloudHSM 키 저장소의 kmsuser 암호 편집

다음 update-custom-key-store 예시에서는 지정된 키 저장소와 연결된 CloudHSM 클러스터에서 kmsuser의 kmsuser 암호 값을 현재 암호로 업데이트합니다. 이 명령은 클러스터의 kmsuser 암호를 변경하지 않습니다. 단지 AWS KMS에 현재 암호를 알려줍니다. KMS에 현재 kmsuser 암호가 없는 경우 AWS CloudHSM 키 저장소에 연결할 수 없습니다.

참고: AWS CloudHSM 키 저장소를 업데이트하기 전에 연결을 해제해야 합니다. disconnect-custom-key-store 명령을 사용합니다. 명령이 완료되면 AWS CloudHSM 키 저장소를 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. 변경이 적용되었는지 확인하려면 describe-custom-key-stores 명령을 사용합니다.

AWS CloudHSM 키 저장소를 업데이트하는 방법에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 AWS CloudHSM 키 저장소 설정 편집을 참조하세요.

예시 3: AWS CloudHSM 키 저장소의 AWS CloudHSM 클러스터 편집

다음 예시에서는 AWS CloudHSM 키 저장소에 연결된 AWS CloudHSM 클러스터를 동일한 클러스터의 다른 백업과 같은 관련 클러스터로 변경합니다.

참고: AWS CloudHSM 키 저장소를 업데이트하기 전에 연결을 해제해야 합니다. `disconnect-custom-key-store` 명령을 사용합니다. 명령이 완료되면 AWS CloudHSM 키 저장소를 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. 변경이 적용되었는지 확인하려면 `describe-custom-key-stores` 명령을 사용합니다.

AWS CloudHSM 키 저장소를 업데이트하는 방법에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 AWS CloudHSM 키 저장소 설정 편집을 참조하세요.

예시 4: 외부 키 저장소의 프록시 인증 자격 증명 편집

다음 예시에서는 외부 키 저장소의 프록시 인증 자격 증명을 업데이트합니다. 값 중 하나만 변경하더라도 `raw-secret-access-key` 및 `access-key-id`를 모두 지정해야 합니다. 이 특성을 사용하여 잘못된 자격 증명을 수정하거나 외부 키 저장소 프록시가 자격 증명을 교체할 때 해당 자격 증명을 변경할 수 있습니다.

외부 키 저장소에서 AWS KMS의 프록시 인증 자격 증명을 설정합니다. 그런 다음 이 명령을 사용하여 AWS KMS에 자격 증명을 제공합니다. AWS KMS는 이 자격 증명을 사용하여 외부 키 저장소 프록시에 대한 요청에 서명합니다.

외부 키 저장소가 연결되거나 연결 해제된 상태에서 프록시 인증 자격 증명을 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. 변경이 적용되었는지 확인하려면 describe-custom-key-stores 명령을 사용합니다.

외부 키 저장소를 업데이트하는 방법에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 외부 키 저장소 속성 편집을 참조하세요.

예시 5: 외부 키 저장소의 프록시 연결 편집

다음 예시에서는 외부 키 스토어 프록시 연결 옵션을 퍼블릭 엔드포인트 연결에서 VPC 엔드포인트 서비스 연결로 변경합니다. xks-proxy-connectivity 값을 변경하는 것 외에도 VPC 엔드포인트 서비스에 연결된 프라이빗 DNS 이름을 반영하도록 xks-proxy-uri-endpoint 값을 변경해야 합니다. 또한 xks-proxy-vpc-endpoint-service-name 값을 추가해야 합니다.

참고: 외부 저장소의 프록시 연결을 업데이트하기 전에 연결을 해제해야 합니다. disconnect-custom-key-store 명령을 사용합니다. 명령이 완료되면 connect-custom-key-store 명령을 사용하여 외부 키 저장소를 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. 변경이 적용되었는지 확인하려면 describe-custom-key-stores 명령을 사용합니다.

외부 키 저장소를 업데이트하는 방법에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 외부 키 저장소 속성 편집을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateCustomKeyStore 섹션을 참조하세요.

update-key-description

다음 코드 예시에서는 update-key-description의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 고객 관리형 KMS 키에 설명을 추가하거나 변경

다음 update-key-description 예시에서는 고객 관리형 KMS 키에 설명을 추가합니다. 동일한 명령을 사용하여 기존 설명을 변경할 수 있습니다.

`--key-id` 파라미터는 명령에서 KMS 키를 식별합니다. 이 예시에서는 키 ARN 값을 사용하지만 KMS 키의 키 ID 또는 ARN을 사용할 수 있습니다. `--description` 파라미터는 새로운 설명을 지정합니다. 이 파라미터의 값은 있는 경우 KMS 키의 현재 설명을 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 AWS Key Management Service API 참조의 [UpdateKeyDescription] 섹션을 참조하세요.

예시 2: 고객 관리형 KMS 키의 설명 삭제

다음 `update-key-description` 예시에서는 고객 관리형 KMS 키의 설명을 삭제합니다.

`--key-id` 파라미터는 명령에서 KMS 키를 식별합니다. 이 예시에서는 키 ID 값을 사용하지만 KMS 키의 키 ID 또는 키 ARN을 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 AWS Key Management Service API 참조의 [UpdateKeyDescription] 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [UpdateKeyDescription] 섹션을 참조하세요.

`verify-mac`

다음 코드 예시는 `verify-mac`의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: HMAC 확인

다음 verify-mac 명령은 특정 메시지에 대한 HMAC, HMAC KMS 키 및 MAC 알고리즘을 확인합니다. 응답의 MacValid 값에서 'true' 값은 HMAC가 유효함을 나타냅니다.

AWS CLI v2에서 message 파라미터 값은 Base64로 인코딩되어야 합니다. 또는 메시지를 파일에 저장하고 AWS CLI에 파일에서 바이너리 데이터를 읽도록 지시하는 fileb:// 접두사를 사용할 수 있습니다.

지정하는 MAC는 base64로 인코딩할 수 없습니다. generate-mac 명령이 반환하는 MAC를 디코딩하는 데 도움이 필요하면 generate-mac 명령 예제를 참조하세요.

이 명령을 실행하기 전에 예시 키 ID를 AWS 계정의 유효한 키 ID로 바꾸세요. 키 ID는 키 사용량이 GENERATE_VERIFY_MAC인 HMAC KMS 키를 나타내야 합니다.

```
msg=(echo 'Hello World' | base64)

aws kms verify-mac \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --message fileb://Message \
    --mac-algorithm HMAC_SHA_384 \
    --mac fileb://ExampleMac
```

출력:

```
{
    "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "MacValid": true,
    "MacAlgorithm": "HMAC_SHA_384"
}
```

AWS KMS에서 HMAC KMS 키를 사용하는 방법에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 HMAC keys in AWS KMS를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 VerifyMac을 참조하세요.

verify

다음 코드 예시는 verify의 사용 방법을 보여 줍니다.

AWS CLI

디지털 서명을 확인하는 방법

다음 verify 명령은 Base64로 인코딩된 짧은 메시지에 대한 암호화 서명을 확인합니다. 키 ID, 메시지, 메시지 유형 및 서명 알고리즘은 메시지 서명에 사용된 것과 동일해야 합니다.

AWS CLI v2에서 message 파라미터 값은 Base64로 인코딩되어야 합니다. 또는 메시지를 파일에 저장하고 AWS CLI에 파일에서 바이너리 데이터를 읽도록 지시하는 fileb:// 접두사를 사용할 수 있습니다.

지정하는 서명은 base64로 인코딩할 수 없습니다. sign 명령이 반환하는 서명을 디코딩하는 데 도움이 필요하면 sign 명령 예시를 참조하세요.

명령의 출력에는 서명이 확인되었음을 나타내는 부울 SignatureValid 필드가 포함됩니다. 서명 검증에 실패하면 verify 명령도 실패합니다.

이 명령을 실행하기 전에 예시 키 ID를 AWS 계정의 유효한 키 ID로 바꾸세요.

```
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"
}
```

AWS KMS에서 비대칭 KMS 키를 사용하는 방법에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 비대칭 키 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 Verify 섹션을 참조하세요.

AWS CLI를 사용한 Lake Formation 예시

다음 코드 예시에서는 Lake Formation에서 AWS Command Line Interface를 사용하여 일반적인 시나리오를 구현하고 개별 서비스 작업을 수행하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

add-lf-tags-to-resource

다음 코드 예시에서는 add-lf-tags-to-resource 코드를 사용하는 방법을 보여줍니다.

AWS CLI

기존 리소스에 하나 이상의 LF 태그를 연결하는 방법

다음 add-lf-tags-to-resource 예시에서는 주어진 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 [Assigning LF-Tags to Data Catalog resources](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [AddLfTagsToResource](#) 섹션을 참조하세요.

`batch-grant-permissions`

다음 코드 예시에서는 `batch-grant-permissions` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

보안 주체에 리소스에 대한 권한을 대량 부여하는 방법

다음 `batch-grant-permissions` 예시에서는 보안 주체에게 지정된 리소스에 대한 액세스 권한을 대량 부여합니다.

```
aws lakeformation batch-grant-permissions \
    --cli-input-json file://input.json
```

`input.json`의 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 Granting and revoking permissions on Data Catalog resources를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 BatchGrantPermissions 섹션을 참조하세요.

`batch-revoke-permissions`

다음 코드 예시에서는 `batch-revoke-permissions` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

보안 주체의 리소스에 대한 권한 대량 취소

다음 batch-revoke-permissions 예시에서는 보안 주체로부터 지정된 리소스에 대한 액세스를 대량으로 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 Granting and revoking permissions on Data Catalog resources를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 BatchRevokePermissions 섹션을 참조하세요.

`cancel-transaction`

다음 코드 예시에서는 `cancel-transaction` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

트랜잭션을 취소하는 방법

다음 `cancel-transaction` 예시에서는 트랜잭션을 취소합니다.

```
aws lakeformation cancel-transaction \
    --transaction-id='b014d972ca8347b89825e33c5774aec4'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Lake Formation 개발자 안내서의 Reading from and writing to the data lake within transactions를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [Command Reference](#) 섹션을 참조하세요.

commit-transaction

다음 코드 예시에서는 commit-transaction 코드를 사용하는 방법을 보여줍니다.

AWS CLI

트랜잭션 커밋

다음 commit-transaction 예시에서는 트랜잭션을 커밋합니다.

```
aws lakeformation commit-transaction \
    --transaction-id='b014d972ca8347b89825e33c5774aec4'
```

출력:

```
{
    "TransactionStatus": "committed"
}
```

자세한 내용은 AWS Lake Formation 개발자 안내서의 [Reading from and writing to the data lake within transactions](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [CommitTransaction](#) 섹션을 참조하세요.

create-data-cells-filter

다음 코드 예시에서는 create-data-cells-filter 코드를 사용하는 방법을 보여줍니다.

AWS CLI

예시 1: 데이터 셀 필터를 생성하는 방법

다음 create-data-cells-filter 예시에서는 행 조건에 따라 특정 열에 대한 액세스 권한을 부여할 수 있는 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 AWS Lake Formation 개발자 안내서의 Data filtering and cell-level security in Lake Formation을 참조하세요.

예시 2: 열 필터를 생성하는 방법

다음 `create-data-cells-filter` 예시에서는 특정 열에 대한 액세스 권한을 부여할 수 있는 데이터 필터를 만듭니다.

```
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 AWS Lake Formation 개발자 안내서의 [Data filtering and cell-level security in Lake Formation](#)을 참조하세요.

예시 3: 제외 열을 사용하여 데이터 필터를 생성하는 방법

다음 `create-data-cells-filter` 예시에서는 언급된 열을 제외한 모든 열에 대한 액세스 권한을 허용하는 데이터 필터를 생성합니다.

```
aws lakeformation create-data-cells-filter \
    --cli-input-json file://input.json
```

`input.json`의 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 AWS Lake Formation 개발자 안내서의 [Data filtering and cell-level security in Lake Formation](#)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [CreateDataCellsFilter](#) 섹션을 참조하세요.

`create-lf-tag`

다음 코드 예시에서는 `create-lf-tag` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

LF 태그 생성

다음 `create-lf-tag` 예시에서는 지정된 이름과 값을 가진 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 Managing LF-Tags for metadata access control을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateLfTag 섹션을 참조하세요.

delete-data-cells-filter

다음 코드 예시에서는 `delete-data-cells-filter` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

데이터 셀 필터를 삭제하는 방법

다음 `delete-data-cells-filter` 예시에서는 지정된 데이터 셀 필터를 삭제합니다.

```
aws lakeformation delete-data-cells-filter \
    --cli-input-json file://input.json
```

`input.json`의 콘텐츠:

```
{
    "TableCatalogId": "123456789111",
    "DatabaseName": "tpc",
    "TableName": "dl_tpc_promotion",
    "Name": "developer_promotion"
}
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Lake Formation 개발자 안내서의 Data filtering and cell-level security in Lake Formation을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteDataCellsFilter 섹션을 참조하세요.

delete-lf-tag

다음 코드 예시에서는 delete-lf-tag 코드를 사용하는 방법을 보여줍니다.

AWS CLI

LF 태그 정의를 삭제하는 방법

다음 delete-lf-tag 예시에서는 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 Managing LF-Tags for metadata access control을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteLfTag 섹션을 참조하세요.

delete-objects-on-cancel

다음 코드 예시에서는 delete-objects-on-cancel 코드를 사용하는 방법을 보여줍니다.

AWS CLI

트랜잭션이 취소될 때 객체를 삭제하는 방법

다음 delete-objects-on-cancel 예시에서는 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 Reading from and writing to the data lake within transactions를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteObjectsOnCancel 섹션을 참조하세요.

deregister-resource

다음 코드 예시에서는 deregister-resource 코드를 사용하는 방법을 보여줍니다.

AWS CLI

데이터 레이크 스토리지 등록을 취소하는 방법

다음 deregister-resource 예시에서는 리소스를 Lake Formation에서 관리하는 것으로 등록을 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 Adding an Amazon S3 location to your data lake를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeregisterResource 섹션을 참조하세요.

describe-transaction

다음 코드 예시에서는 describe-transaction 코드를 사용하는 방법을 보여줍니다.

AWS CLI

트랜잭션 세부 정보를 검색하는 방법

다음 describe-transaction 예시에서는 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 Reading from and writing to the data lake within transactions를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeTransaction 섹션을 참조하세요.

extend-transaction

다음 코드 예시에서는 extend-transaction 코드를 사용하는 방법을 보여줍니다.

AWS CLI

트랜잭션을 확장하는 방법

다음 extend-transaction 예시는 트랜잭션을 확장합니다.

```
aws lakeformation extend-transaction \
    --transaction-id='8cb4b1a7cc8d486fbaca9a64e7d9f5ce'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Lake Formation 개발자 안내서의 Reading from and writing to the data lake within transactions를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ExtendTransaction 섹션을 참조하세요.

get-data-lake-settings

다음 코드 예시에서는 get-data-lake-settings 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS Lake Formation 관리형 데이터 레이크 설정을 검색하는 방법

다음 get-data-lake-settings 예시에서는 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 Changing the default security settings for your data lake를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetDataLakeSettings 섹션을 참조하세요.

get-effective-permissions-for-path

다음 코드 예시에서는 `get-effective-permissions-for-path` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

특정 경로에 있는 리소스에 대한 권한을 검색하는 방법

다음 `get-effective-permissions-for-path` 예시에서는 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.json`의 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 AWS Lake Formation 개발자 안내서의 Managing Lake Formation permissions를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetEffectivePermissionsForPath 섹션을 참조하세요.

get-lf-tag

다음 코드 예시에서는 `get-lf-tag` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

LF 태그 정의를 검색하는 방법

다음 `get-lf-tag` 예시에서는 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 Managing LF-Tags for metadata access control을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetLfTag 섹션을 참조하세요.

get-query-state

다음 코드 예시에서는 get-query-state 코드를 사용하는 방법을 보여줍니다.

AWS CLI

제출된 쿼리의 상태를 검색하는 방법

다음 get-query-state 예시에서는 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 Transactional data operations를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetQueryState 섹션을 참조하세요.

get-query-statistics

다음 코드 예시에서는 get-query-statistics 코드를 사용하는 방법을 보여줍니다.

AWS CLI

쿼리 통계를 검색하는 방법

다음 get-query-statistics 예시에서는 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 Transactional data operations를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetQueryStatistics 섹션을 참조하세요.

get-resource-lf-tags

다음 코드 예시에서는 get-resource-lf-tags 코드를 사용하는 방법을 보여줍니다.

AWS CLI

LF 태그 나열

다음 list-lf-tags 예시에서는 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 Managing LF-Tags for metadata access control을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetResourceLfTags 섹션을 참조하세요.

get-table-objects

다음 코드 예시에서는 `get-table-objects` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

관리 테이블의 객체를 나열하는 방법

다음 `get-table-objects` 예시에서는 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 Reading from and writing to the data lake within transactions를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetTableObjects 섹션을 참조하세요.

get-work-unit-results

다음 코드 예시에서는 get-work-unit-results 코드를 사용하는 방법을 보여줍니다.

AWS CLI

지정된 쿼리의 작업 단위를 검색하는 방법

다음 get-work-unit-results 예시에서는 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 [Transactional data operations](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [GetWorkUnitResults](#) 섹션을 참조하세요.

`get-work-units`

다음 코드 예시에서는 `get-work-units` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

작업 단위를 검색하는 방법

다음 `get-work-units` 예시에서는 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 Transactional data operations를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetWorkUnits 섹션을 참조하세요.

grant-permissions

다음 코드 예시에서는 grant-permissions 코드를 사용하는 방법을 보여줍니다.

AWS CLI

예시 1: LF 태그를 사용하여 리소스에 대한 보안 주체에 권한 부여

다음 grant-permissions 예시에서는 LF 태그 정책과 일치하는 데이터베이스 리소스의 보안 주체에게 모든 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 [Granting and revoking permissions on Data Catalog resources](#)를 참조하세요.

예시 2: 보안 주체에 열 수준 권한을 부여하는 방법

다음 `grant-permissions` 예시에서는 보안 주체에게 특정 열을 선택할 수 있는 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": {
        "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 [Granting and revoking permissions on Data Catalog resources](#)를 참조하세요.

예시 3: 보안 주체에 테이블 권한을 부여하는 방법

다음 `grant-permissions` 예시에서는 지정된 데이터베이스의 모든 테이블에 대한 선택 권한을 보안 주체에게 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": {
        "Table": {
            "CatalogId": "123456789111",
            "DatabaseName": "tpc",
            "TableWildcard": {}
        }
    },
    "Permissions": [
        "SELECT"
    ],
    "PermissionsWithGrantOption": []
}
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Lake Formation 개발자 안내서의 [Granting and revoking permissions on Data Catalog resources]를 참조하세요.

예시 4: 보안 주체에게 LF 태그에 대한 권한을 부여하는 방법

다음 `grant-permissions` 예시에서는 보안 주체에게 LF 태그에 대한 연결 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": {
        "LFTag": {
            "CatalogId": "123456789111",
            "TagKey": "category",
            "TagValues": [
                "private", "public"
            ]
        }

    },
    "Permissions": [
        "ASSOCIATE"
    ],
    "PermissionsWithGrantOption": []
}
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Lake Formation 개발자 안내서의 Granting and revoking permissions on Data Catalog resources를 참조하세요.

예시 5: 보안 주체에게 데이터 위치에 대한 권한을 부여하는 방법

다음 `grant-permissions` 예시에서는 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 Granting and revoking permissions on Data Catalog resources를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GrantPermissions 섹션을 참조하세요.

list-data-cells-filter

다음 코드 예시에서는 `list-data-cells-filter` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

데이터 셀 필터를 나열하는 방법

다음 `list-data-cells-filter` 예시에서는 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 AWS Lake Formation 개발자 안내서의 [Data filtering and cell-level security in Lake Formation](#)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ListDataCellsFilter](#) 섹션을 참조하세요.

`list-permissions`

다음 코드 예시에서는 `list-permissions` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

예시 1: 리소스에 대한 보안 주체 권한 목록을 검색하는 방법

다음 `list-permissions` 예시에서는 데이터베이스 리소스에 대한 보안 주체 권한 목록을 반환합니다.

```
aws lakeformation list-permissions \
    --cli-input-json file://input.json
```

`input.json`의 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 AWS Lake Formation 개발자 안내서의 Managing Lake Formation permissions를 참조하세요.

예시 2: 데이터 필터를 사용하여 테이블의 보안 주체 권한 목록을 검색하는 방법

다음 `list-permissions` 예시에서는 보안 주체에게 부여된 관련 데이터 필터와 함께 테이블의 권한을 나열합니다.

```
aws lakeformation list-permissions \
    --cli-input-json file://input.json
```

`input.json`의 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 AWS Lake Formation 개발자 안내서의 Managing Lake Formation permissions를 참조하세요.

예시 3: LF 태그에 대한 보안 주체 권한 목록을 검색하는 방법

다음 `list-permissions` 예시에서는 보안 주체에게 부여된 LF 태그에 대한 권한을 나열합니다.

```
aws lakeformation list-permissions \
    --cli-input-json file://input.json
```

`input.json`의 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": "EJwY21GMGFXOXVJanA3SW50cm1pc3Npb25zIjpmYWxzZX0="
}
```

자세한 내용은 AWS Lake Formation 개발자 안내서의 Managing Lake Formation permissions를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListPermissions 섹션을 참조하세요.

list-resources

다음 코드 예시에서는 `list-resources` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Lake Formation에서 관리하는 리소스를 나열하는 방법

다음 `list-resources` 예시에서는 Lake Formation에서 관리하는 조건과 일치하는 리소스를 나열합니다.

```
aws lakeformation list-resources \
    --cli-input-json file://input.json
```

`input.json`의 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 AWS Lake Formation 개발자 안내서의 Managing Lake Formation permissions를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListResources 섹션을 참조하세요.

`list-transactions`

다음 코드 예시에서는 `list-transactions` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

모든 트랜잭션 세부 정보를 나열하는 방법

다음 `list-transactions` 예시에서는 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 Reading from and writing to the data lake within transactions를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTransactions 섹션을 참조하세요.

put-data-lake-settings

다음 코드 예시에서는 put-data-lake-settings 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS Lake Formation 관리형 데이터 레이크 설정을 설정하는 방법

다음 put-data-lake-settings 예시에서는 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 Changing the default security settings for your data lake를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutDataLakeSettings 섹션을 참조하세요.

register-resource

다음 코드 예시에서는 register-resource 코드를 사용하는 방법을 보여줍니다.

AWS CLI

예시 1: Service Linked Role을 사용하여 데이터 레이크 스토리지 등록

다음 register-resource 예시에서는 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 Adding an Amazon S3 location to your data lake를 참조하세요.

예시 2: 사용자 지정 역할을 사용하여 데이터 레이크 스토리지 등록

다음 `register-resource` 예시에서는 사용자 지정 역할을 사용하여 Lake Formation에서 관리하는 대로 리소스를 등록합니다.

```
aws lakeformation register-resource \
    --cli-input-json file://input.json
```

`input.json`의 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 Adding an Amazon S3 location to your data lake를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RegisterResource 섹션을 참조하세요.

`remove-lf-tags-from-resource`

다음 코드 예시에서는 `remove-lf-tags-from-resource` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

리소스에서 LF 태그 제거

다음 `remove-lf-tags-from-resource` 예시에서는 테이블 리소스와의 LF 태그 연결을 제거합니다.

```
aws lakeformation remove-lf-tags-from-resource \
    --cli-input-json file://input.json
```

`input.json`의 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 Assigning LF-Tags to Data Catalog resources를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RemoveLfTagsFromResource 섹션을 참조하세요.

`revoke-permissions`

다음 코드 예시에서는 `revoke-permissions` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

보안 주체의 리소스에 대한 권한을 취소하는 방법

다음 `revoke-permissions` 예시에서는 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 Granting and revoking permissions on Data Catalog resources를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RevokePermissions 섹션을 참조하세요.

`search-databases-by-lf-tags`

다음 코드 예시에서는 `search-databases-by-lf-tags` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

LFTags로 데이터베이스 리소스를 검색하는 방법

다음 `search-databases-by-lf-tags` 예시는 LFTag 표현식과 일치하는 데이터베이스 리소스를 검색합니다.

```
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 Viewing the resources that a LF-Tag is assigned to를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 SearchDatabasesByLfTags 섹션을 참조하세요.

search-tables-by-lf-tags

다음 코드 예시에서는 search-tables-by-lf-tags 코드를 사용하는 방법을 보여줍니다.

AWS CLI

LFTags로 테이블 리소스를 검색하는 방법

다음 search-tables-by-lf-tags 예시에서는 LFTag 표현식과 일치하는 테이블 리소스를 검색합니다.

```
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 Viewing the resources that a LF-Tag is assigned to를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 SearchTablesByLfTags 섹션을 참조하세요.

start-query-planning

다음 코드 예시에서는 start-query-planning 코드를 사용하는 방법을 보여줍니다.

AWS CLI

쿼리 문을 처리하는 방법

다음 start-query-planning 예시에서는 쿼리 문을 처리하기 위한 요청을 제출합니다.

```
aws lakeformation start-query-planning \
    --cli-input-json file://input.json
```

input.json의 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 Reading from and writing to the data lake within transactions를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartQueryPlanning 섹션을 참조하세요.

start-transaction

다음 코드 예시에서는 `start-transaction` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

새 트랜잭션 시작

다음 `start-transaction` 예시에서는 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 Reading from and writing to the data lake within transactions를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartTransaction 섹션을 참조하세요.

update-lf-tag

다음 코드 예시에서는 `update-lf-tag` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

LF 태그 정의를 업데이트하는 방법

다음 `update-lf-tag` 예시에서는 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 Managing LF-Tags for metadata access control을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateLfTag 섹션을 참조하세요.

update-table-objects

다음 코드 예시에서는 `update-table-objects` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

관리 테이블의 객체를 수정하는 방법

다음 `update-table-objects` 예시에서는 제공된 S3 객체를 지정된 관리 테이블에 추가합니다.

```
aws lakeformation update-table-objects \
    --cli-input-json file://input.json
```

`input.json`의 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 Reading from and writing to the data lake within transactions를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateTableObjects 섹션을 참조하세요.

AWS CLI를 사용한 Lambda 예시

다음 코드 예시는 Lambda와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

add-layer-version-permission

다음 코드 예시에서는 `add-layer-version-permission` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

계층 버전에 권한을 추가하는 방법

다음 `add-layer-version-permission` 예시에서는 지정된 계정이 계층 `my-layer`의 버전 1을 사용할 수 있는 권한을 부여합니다.

```
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 AWS Lambda Layers를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AddLayerVersionPermission 섹션을 참조하세요.

add-permission

다음 코드 예시에서는 add-permission 코드를 사용하는 방법을 보여줍니다.

AWS CLI

기존 Lambda 함수에서 권한 추가

다음 add-permission 예시에서는 Amazon SNS 서비스에 my-function 함수를 호출할 수 있는 권한을 부여합니다.

```
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 AWS Lambda에 리소스 기반 정책 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AddPermission 섹션을 참조하세요.

create-alias

다음 코드 예시에서는 `create-alias` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Lambda 함수의 별칭 생성

다음 `create-alias` 예시에서는 `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 AWS Lambda 함수 별칭 구성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateAlias를 참조하세요.

create-event-source-mapping

다음 코드 예시에서는 `create-event-source-mapping` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

이벤트 소스와 AWS Lambda 함수 간에 매핑 생성

다음 `create-event-source-mapping` 예시에서는 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 AWS Lambda Event Source Mapping을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateEventSourceMapping 섹션을 참조하세요.

create-function

다음 코드 예시에서는 `create-function` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Lambda 함수 생성

다음 `create-function` 예시에서는 `my-function`이라는 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.zip`의 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 내용은 AWS Lambda 개발자 안내서의 AWS Lambda 함수 구성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateFunction을 참조하세요.

delete-alias

다음 코드 예시에서는 `delete-alias` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Lambda 함수의 별칭 삭제

다음 `delete-alias` 예시에서는 `my-function` Lambda 함수에서 `LIVE`라는 별칭을 삭제합니다.

```
aws lambda delete-alias \
    --function-name my-function \
    --name LIVE
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Lambda 개발자 안내서의 AWS Lambda 함수 별칭 구성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteAlias를 참조하세요.

delete-event-source-mapping

다음 코드 예시에서는 `delete-event-source-mapping` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

이벤트 소스와 AWS Lambda 함수 간의 매핑을 삭제하는 방법

다음 `delete-event-source-mapping` 예시에서는 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 AWS Lambda Event Source Mapping을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteEventSourceMapping 섹션을 참조하세요.

delete-function-concurrency

다음 코드 예시에서는 delete-function-concurrency 코드를 사용하는 방법을 보여줍니다.

AWS CLI

예약된 동시 실행 한도를 함수에서 제거

다음 delete-function-concurrency 예시에서는 예약된 동시 실행 한도를 my-function 함수에서 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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteFunctionConcurrency를 참조하세요.

delete-function-event-invoke-config

다음 코드 예시에서는 delete-function-event-invoke-config 코드를 사용하는 방법을 보여줍니다.

AWS CLI

비동기식 간접 호출 구성 삭제

다음 delete-function-event-invoke-config 예시에서는 지정된 함수의 GREEN 별칭에 대한 비동기식 간접 호출 구성을 삭제합니다.

```
aws lambda delete-function-event-invoke-config --function-name my-function:GREEN
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteFunctionEventInvokeConfig 섹션을 참조하세요.

delete-function

다음 코드 예시에서는 delete-function 코드를 사용하는 방법을 보여줍니다.

AWS CLI

예시 1: 함수 이름을 기준으로 Lambda 함수 삭제

다음 delete-function 예시에서는 함수 이름을 지정하여 my-function이라는 Lambda 함수를 삭제합니다.

```
aws lambda delete-function \
    --function-name my-function
```

이 명령은 출력을 생성하지 않습니다.

예시 2: 함수 ARN을 기준으로 Lambda 함수 삭제

다음 delete-function 예시에서는 함수 ARN을 지정하여 my-function이라는 Lambda 함수를 삭제합니다.

```
aws lambda delete-function \
    --function-name arn:aws:lambda:us-west-2:123456789012:function:my-function
```

이 명령은 출력을 생성하지 않습니다.

예시 3: 함수의 부분 ARN을 기준으로 Lambda 함수 삭제

다음 delete-function 예시에서는 함수의 부분 ARN을 지정하여 my-function이라는 Lambda 함수를 삭제합니다.

```
aws lambda delete-function \
    --function-name 123456789012:function:my-function
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Lambda 개발자 안내서의 AWS Lambda 함수 구성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteFunction을 참조하세요.

delete-layer-version

다음 코드 예시에서는 delete-layer-version 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Lambda 계층의 버전을 삭제하는 방법

다음 delete-layer-version 예시에서는 my-layer 계층의 버전 2를 삭제합니다.

```
aws lambda delete-layer-version \
    --layer-name my-layer \
    --version-number 2
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Lambda 개발자 안내서의 AWS Lambda Layers를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteLayerVersion 섹션을 참조하세요.

delete-provisioned-concurrency-config

다음 코드 예시에서는 delete-provisioned-concurrency-config 코드를 사용하는 방법을 보여줍니다.

AWS CLI

프로비저닝된 동시성 구성 삭제

다음 delete-provisioned-concurrency-config 예시에서는 지정된 함수의 GREEN 별칭에 대해 프로비저닝된 동시성 구성을 삭제합니다.

```
aws lambda delete-provisioned-concurrency-config \
    --function-name my-function \
    --qualifier GREEN
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteProvisionedConcurrencyConfig를 참조하세요.

`get-account-settings`

다음 코드 예시에서는 `get-account-settings` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS 리전 내 계정의 세부 정보 가져오기

다음 `get-account-settings` 예시는 계정의 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 한도를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetAccountSettings를 참조하세요.

`get-alias`

다음 코드 예시에서는 `get-alias` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

함수 별칭의 세부 정보 가져오기

다음 `get-alias` 예시에서는 `my-function` Lambda 함수에서 `LIVE`라는 별칭의 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 AWS Lambda 함수 별칭 구성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetAlias를 참조하세요.

get-event-source-mapping

다음 코드 예시에서는 get-event-source-mapping 코드를 사용하는 방법을 보여줍니다.

AWS CLI

이벤트 소스 매핑에 대한 세부 정보 검색

다음 get-event-source-mapping 예시에서는 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 AWS Lambda Event Source Mapping을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetEventSourceMapping 섹션을 참조하세요.

get-function-concurrency

다음 코드 예시에서는 `get-function-concurrency` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

함수에 대한 예약된 동시성 설정 보기

다음 `get-function-concurrency` 예시에서는 지정된 함수에 대한 예약된 동시성 설정을 가져옵니다.

```
aws lambda get-function-concurrency \
    --function-name my-function
```

출력:

```
{
    "ReservedConcurrentExecutions": 250
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetFunctionConcurrency를 참조하세요.

get-function-configuration

다음 코드 예시에서는 `get-function-configuration` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Lambda 함수의 버전별 설정 가져오기

다음 `get-function-configuration` 예시에서는 `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 내용은 AWS Lambda 개발자 안내서의 AWS Lambda 함수 구성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetFunctionConfiguration을 참조하세요.

get-function-event-invoke-config

다음 코드 예시에서는 `get-function-event-invoke-config` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

비동기식 간접 호출 구성 보기

다음 `get-function-event-invoke-config` 예시에서는 지정된 함수의 BLUE 별칭에 대한 비동기식 간접 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"
        }
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetFunctionEventInvokeConfig 섹션을 참조하세요.

get-function

다음 코드 예시에서는 get-function 코드를 사용하는 방법을 보여줍니다.

AWS CLI

함수 정보 가져오기

다음 get-function 예시에서는 my-function 함수의 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 내용은 AWS Lambda 개발자 안내서의 AWS Lambda 함수 구성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetFunction을 참조하세요.

get-layer-version-by-arn

다음 코드 예시에서는 `get-layer-version-by-arn` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Lambda 계층 버전에 대한 정보를 검색하는 방법

다음 `get-layer-version-by-arn` 예시에서는 지정된 Amazon 리소스 이름(ARN)을 사용하여 계층 버전에 대한 정보를 표시합니다.

```
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 AWS Lambda Layers를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetLayerVersionByArn 섹션을 참조하세요.

get-layer-version-policy

다음 코드 예시에서는 get-layer-version-policy 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Lambda 계층 버전에 대한 권한 정책을 검색하는 방법

다음 get-layer-version-policy 예시에서는 my-layer 계층의 버전 1에 대한 정책 정보를 표시합니다.

```
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 AWS Lambda Layers를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetLayerVersionPolicy 섹션을 참조하세요.

get-layer-version

다음 코드 예시에서는 get-layer-version 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Lambda 계층 버전에 대한 정보를 검색하는 방법

다음 get-layer-version 예시에서는 my-layer 계층의 버전 1에 대한 정보를 표시합니다.

```
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 AWS Lambda Layers를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetLayerVersion 섹션을 참조하세요.

get-policy

다음 코드 예시에서는 `get-policy` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

함수, 버전 또는 별칭에 대한 리소스 기반 IAM 정책 가져오기

다음 `get-policy` 예시에서는 `my-function` Lambda 함수의 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 AWS Lambda에 리소스 기반 정책 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetPolicy를 참조하세요.

`get-provisioned-concurrency-config`

다음 코드 예시에서는 `get-provisioned-concurrency-config` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

프로비저닝된 동시성 구성 보기

다음 `get-provisioned-concurrency-config` 예시에서는 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetProvisionedConcurrencyConfig를 참조하세요.

invoke

다음 코드 예시에서는 `invoke` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

예시 1: Lambda 함수를 동기식으로 간접 호출

다음 `invoke` 예시에서는 `my-function` 함수를 동기식으로 간접 호출합니다. `cli-binary-format` 옵션은 AWS CLI 버전 2를 사용할 때 필요합니다. 자세한 내용은 AWS Command Line Interface 사용자 안내서의 AWS CLI에서 지원되는 전역 명령줄 옵션을 참조하세요.

```
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 Lambda 개발자 안내서의 Lambda 함수의 동기식 간접 호출을 참조하세요.

예시 2: Lambda 함수를 비동기식으로 간접 호출

다음 invoke 예시에서는 my-function 함수를 비동기식으로 간접 호출합니다. cli-binary-format 옵션은 AWS CLI 버전 2를 사용할 때 필요합니다. 자세한 내용은 AWS Command Line Interface 사용자 안내서의 AWS CLI에서 지원되는 전역 명령줄 옵션을 참조하세요.

```
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 Lambda 개발자 안내서의 Lambda 함수를 비동기 호출합니다.를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 Invoke를 참조하세요.

list-aliases

다음 코드 예시에서는 list-aliases 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Lambda 함수에 대한 태그 목록 검색

다음 list-aliases 예시에서는 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 AWS Lambda 함수 별칭 구성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListAliases를 참조하세요.

`list-event-source-mappings`

다음 코드 예시에서는 `list-event-source-mappings` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

함수에 대한 이벤트 소스 매핑을 나열하는 방법

다음 `list-event-source-mappings` 예시에서는 `my-function` Lambda 함수의 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 AWS Lambda Event Source Mapping을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListEventSourceMappings 섹션을 참조하세요.

`list-function-event-invoke-configs`

다음 코드 예시에서는 `list-function-event-invoke-configs` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

비동기 호출 구성 목록을 보는 방법

다음 `list-function-event-invoke-configs` 예시는 지정된 함수의 비동기식 간접 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
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListFunctionEventInvokeConfigs 섹션을 참조하세요.

list-functions

다음 코드 예시에서는 `list-functions` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Lambda 함수 목록 가져오기

다음 `list-functions` 예시에서는 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 내용은 AWS Lambda 개발자 안내서의 AWS Lambda 함수 구성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListFunctions를 참조하세요.

list-layer-versions

다음 코드 예시에서는 `list-layer-versions` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS Lambda 계층의 버전을 나열하는 방법

다음 `list-layers-versions` 예시에서는 `my-layer` 계층의 버전에 대한 정보를 표시합니다.

```
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 AWS Lambda Layers를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListLayerVersions 섹션을 참조하세요.

`list-layers`

다음 코드 예시에서는 `list-layers` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

함수의 런타임과 호환되는 레이어 나열

다음 `list-layers` 예시에서는 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 AWS Lambda Layers를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListLayers 섹션을 참조하세요.

`list-provisioned-concurrency-configs`

다음 코드 예시에서는 `list-provisioned-concurrency-configs` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

프로비저닝된 동시성 구성의 목록 가져오기

다음 `list-provisioned-concurrency-configs` 예시에서는 지정된 함수의 프로비저닝된 동시성 구성을 나열합니다.

```
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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListProvisionedConcurrencyConfigs를 참조하세요.

list-tags

다음 코드 예시에서는 `list-tags` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Lambda 함수의 태그 목록 가져오기

다음 `list-tags` 예시에서는 `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.

- API 세부 정보는 AWS CLI 명령 참조의 ListTags를 참조하세요.

list-versions-by-function

다음 코드 예시에서는 `list-versions-by-function` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

함수 버전 목록 가져오기

다음 `list-versions-by-function` 예시에서는 `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 AWS Lambda 함수 별칭 구성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListVersionsByFunction을 참조하세요.

publish-layer-version

다음 코드 예시에서는 publish-layer-version 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Lambda 계층 버전을 생성하는 방법

다음 publish-layer-version 예시에서는 새 Python 라이브러리 계층 버전을 생성합니다. 명령은 지정된 S3 버킷에 layer.zip 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 AWS Lambda Layers를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PublishLayerVersion 섹션을 참조하세요.

publish-version

다음 코드 예시에서는 publish-version 코드를 사용하는 방법을 보여줍니다.

AWS CLI

함수의 새 버전 게시

다음 publish-version 예시에서는 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 AWS Lambda 함수 별칭 구성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PublishVersion을 참조하세요.

put-function-concurrency

다음 코드 예시에서는 put-function-concurrency 코드를 사용하는 방법을 보여줍니다.

AWS CLI

함수에 대해 예약된 동시성 한도 구성

다음 put-function-concurrency 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 PutFunctionConcurrency를 참조하세요.

put-function-event-invoke-config

다음 코드 예시에서는 put-function-event-invoke-config 코드를 사용하는 방법을 보여줍니다.

AWS CLI

비동기식 간접 호출에 대한 오류 처리 구성

다음 put-function-event-invoke-config 예시에서는 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": {}
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 PutFunctionEventInvokeConfig 섹션을 참조하세요.

put-provisioned-concurrency-config

다음 코드 예시에서는 put-provisioned-concurrency-config 코드를 사용하는 방법을 보여줍니다.

AWS CLI

프로비저닝된 동시성 할당

다음 put-provisioned-concurrency-config 예시에서는 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 PutProvisionedConcurrencyConfig를 참조하세요.

`remove-layer-version-permission`

다음 코드 예시에서는 `remove-layer-version-permission` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

계층 버전 권한을 삭제하는 방법

다음 `remove-layer-version-permission` 예시에서는 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 AWS Lambda Layers를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RemoveLayerVersionPermission 섹션을 참조하세요.

`remove-permission`

다음 코드 예시에서는 `remove-permission` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

기존 Lambda 함수에서 권한 제거

다음 `remove-permission` 예시에서는 `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 AWS Lambda에 리소스 기반 정책 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RemovePermission을 참조하세요.

`tag-resource`

다음 코드 예시에서는 `tag-resource` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

기존 Lambda 함수에 태그 추가

다음 `tag-resource` 예시는 키 이름 `DEPARTMENT` 및 값 `Department A`가 있는 태그를 지정된 Lambda 함수에 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.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

`untag-resource`

다음 코드 예시에서는 `untag-resource` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

기존 Lambda 함수에서 태그 제거

다음 `untag-resource` 예시에서는 `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.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

update-alias

다음 코드 예시에서는 `update-alias` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

함수 별칭 업데이트

다음 `update-alias` 예시에서는 `my-function` Lambda 함수의 버전 3을 가리키도록 `LIVE`라는 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 AWS Lambda 함수 별칭 구성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateAlias를 참조하세요.

update-event-source-mapping

다음 코드 예시에서는 `update-event-source-mapping` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

이벤트 소스와 AWS Lambda 함수 간의 매핑을 업데이트하는 방법

다음 update-event-source-mapping 예시에서는 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 AWS Lambda Event Source Mapping을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateEventSourceMapping 섹션을 참조하세요.

update-function-code

다음 코드 예시에서는 update-function-code 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Lambda 함수 코드 업데이트

다음 update-function-code 예시에서는 my-function 함수의 게시되지 않은 ($LATEST) 버전 코드를 지정된 zip 파일의 콘텐츠로 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 내용은 AWS Lambda 개발자 안내서의 AWS Lambda 함수 구성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateFunctionCode를 참조하세요.

update-function-configuration

다음 코드 예시에서는 `update-function-configuration` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

함수 구성 수정

다음 `update-function-configuration` 예시에서는 `my-function` 함수의 게시되지 않은 ($LATEST) 버전에서 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 내용은 AWS Lambda 개발자 안내서의 AWS Lambda 함수 구성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateFunctionConfiguration을 참조하세요.

update-function-event-invoke-config

다음 코드 예시에서는 update-function-event-invoke-config 코드를 사용하는 방법을 보여줍니다.

AWS CLI

비동기식 간접 호출 구성 업데이트

다음 update-function-event-invoke-config 예시에서는 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"
        }
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateFunctionEventInvokeConfig 섹션을 참조하세요.

AWS CLI를 사용한 License Manager 예시

다음 코드 예시는 License Manager와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`create-license-configuration`

다음 코드 예시에서는 `create-license-configuration`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 라이선스 구성 생성

다음 `create-license-configuration` 예시에서는 코어 10개의 하드 제한으로 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"
}
```

예시 2: 라이선스 구성 생성

다음 `create-license-configuration` 예시에서는 vCPU 100개의 소프트 제한으로 라이선스 구성을 생성합니다. 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"
```

```
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateLicenseConfiguration을 참조하세요.

delete-license-configuration

다음 코드 예시에서는 delete-license-configuration의 사용 방법을 보여줍니다.

AWS CLI

라이선스 구성 삭제

다음 delete-license-configuration 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteLicenseConfiguration을 참조하세요.

get-license-configuration

다음 코드 예시에서는 get-license-configuration의 사용 방법을 보여줍니다.

AWS CLI

라이선스 구성 정보 가져오기

다음 get-license-configuration 예시에서는 지정된 라이선스 구성의 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
        }
    ]
}
```


- API 세부 정보는 AWS CLI 명령 참조의 [GetLicenseConfiguration](#)을 참조하세요.

`get-service-settings`

다음 코드 예시에서는 `get-service-settings`의 사용 방법을 보여줍니다.

AWS CLI

License Manager 설정 가져오기

다음 `get-service-settings` 예시에서는 현재 리전의 License Manager에 대한 서비스 설정을 표시합니다.

```
aws license-manager get-service-settings
```

다음은 교차 계정 리소스 검색이 비활성화된 경우의 예시 출력입니다.

```
{
    "OrganizationConfiguration": {
        "EnableIntegration": false
    },
    "EnableCrossAccountsDiscovery": false
}
```

다음은 교차 계정 리소스 검색이 활성화된 경우의 예시 출력입니다.

```
{
    "S3BucketArn": "arn:aws:s3:::aws-license-manager-service-c22d6279-35c4-47c4-bb",
    "OrganizationConfiguration": {
        "EnableIntegration": true
    },
    "EnableCrossAccountsDiscovery": true
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [GetServiceSettings](https://docs.aws.amazon.com/goto/aws-cli/license-manager-2018-08-01/GetServiceSettings)를 참조하세요.

`list-associations-for-license-configuration`

다음 코드 예시에서는 `list-associations-for-license-configuration`의 사용 방법을 보여줍니다.

AWS CLI

라이선스 구성에 대한 연결 가져오기

다음 list-associations-for-license-configuration 예시에서는 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
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListAssociationsForLicenseConfiguration을 참조하세요.

list-license-configurations

다음 코드 예시에서는 list-license-configurations의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 모든 라이선스 구성 나열

다음 list-license-configurations 예시에서는 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
                    }
                ]
            },
            {
                ...
            }
    ]
}
```

예시 2: 특정 라이선스 구성 나열

다음 `list-license-configurations` 예시에서는 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
```

- API 세부 정보는 AWS CLI 명령 참조의 ListLicenseConfigurations를 참조하세요.

list-license-specifications-for-resource

다음 코드 예시에서는 `list-license-specifications-for-resource`의 사용 방법을 보여줍니다.

AWS CLI

리소스의 라이선스 구성 나열

다음 `list-license-specifications-for-resource` 예시에서는 지정된 Amazon Machine Image(AMI)에 연결된 라이선스 구성을 나열합니다.

```
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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [ListLicenseSpecificationsForResource]를 참조하세요.

`list-resource-inventory`

다음 코드 예시에서는 `list-resource-inventory`의 사용 방법을 보여줍니다.

AWS CLI

리소스 인벤토리의 리소스 나열

다음 `list-resource-inventory` 예시에서는 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [ListResourceInventory](#)를 참조하세요.

list-tags-for-resource

다음 코드 예시에서는 `list-tags-for-resource`의 사용 방법을 보여줍니다.

AWS CLI

라이선스 구성의 태그 나열

다음 `list-tags-for-resource` 예시는 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [ListTagsForResource](#) 섹션을 참조하세요.

list-usage-for-license-configuration

다음 코드 예시에서는 `list-usage-for-license-configuration`의 사용 방법을 보여줍니다.

AWS CLI

라이선스 구성에 사용 중인 라이선스 나열

다음 list-usage-for-license-configuration 예시에서는 지정된 라이선스 구성에 대한 라이선스를 사용하는 리소스의 정보를 나열합니다. 예를 들어 라이선스 유형이 vCPU인 경우 모든 인스턴스는 vCPU당 하나의 라이선스를 사용합니다.

```
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
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListUsageForLicenseConfiguration을 참조하세요.

tag-resource

다음 코드 예시에서는 tag-resource의 사용 방법을 보여줍니다.

AWS CLI

라이선스 구성에 태그 추가

다음 tag-resource 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 [TagResource](#)를 참조하세요.

`untag-resource`

다음 코드 예시에서는 `untag-resource`의 사용 방법을 보여줍니다.

AWS CLI

라이선스 구성에서 태그 제거

다음 `untag-resource` 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 [UntagResource](#)를 참조하세요.

`update-license-configuration`

다음 코드 예시에서는 `update-license-configuration`의 사용 방법을 보여줍니다.

AWS CLI

라이선스 구성 업데이트

다음 `update-license-configuration` 예시에서는 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 예시에서는 지정된 라이선스 구성을 업데이트하여 상태를 DISABLED로 변경합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateLicenseConfiguration을 참조하세요.

update-license-specifications-for-resource

다음 코드 예시에서는 update-license-specifications-for-resource의 사용 방법을 보여줍니다.

AWS CLI

리소스의 라이선스 구성 업데이트

다음 update-license-specifications-for-resource 예시는 하나의 라이선스 구성을 제거하고 다른 구성을 추가하여 지정된 Amazon Machine Image(AMI)에 연결된 라이선스 구성을 바꿉니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateLicenseSpecificationsForResource를 참조하세요.

update-service-settings

다음 코드 예시에서는 update-service-settings의 사용 방법을 보여줍니다.

AWS CLI

License Manager 설정 업데이트

다음 `update-service-settings` 예시에서는 현재 AWS 리전의 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.

- API 세부 정보는 AWS CLI 명령 참조의 [UpdateServiceSettings](https://docs.aws.amazon.com/goto/aws-cli/license-manager-2018-08-01/UpdateServiceSettings)를 참조하세요.

AWS CLI를 사용한 Lightsail 예시

다음 코드 예시에서는 Lightsail에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`allocate-static-ip`

다음 코드 예시에서는 `allocate-static-ip`의 사용 방법을 보여줍니다.

AWS CLI

고정 IP 생성

다음 allocate-static-ip 예시에서는 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
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 AllocateStaticIp 섹션을 참조하세요.

attach-disk

다음 코드 예시에서는 attach-disk의 사용 방법을 보여줍니다.

AWS CLI

블록 스토리지 디스크를 인스턴스에 연결하는 방법

다음 attach-disk 예시에서는 /dev/xvdf의 디스크 경로를 사용하여 인스턴스 WordPress_Multisite-1에 디스크 Disk-1을 연결합니다.

```
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
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 AttachDisk 섹션을 참조하세요.

attach-instances-to-load-balancer

다음 코드 예시에서는 attach-instances-to-load-balancer의 사용 방법을 보여줍니다.

AWS CLI

로드 밸런서에 인스턴스 연결

다음 attach-instances-to-load-balancer 예시에서는 인스턴스 MEAN-1, MEAN-2 및 MEAN-3을 로드 밸런서 LoadBalancer-1에 연결합니다.

```
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
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 AttachInstancesToLoadBalancer 섹션을 참조하세요.

attach-load-balancer-tls-certificate

다음 코드 예시에서는 attach-load-balancer-tls-certificate의 사용 방법을 보여줍니다.

AWS CLI

로드 밸런서에 TLS 인증서 연결

다음 attach-load-balancer-tls-certificate 예시에서는 로드 밸런서 TLS 인증서 Certificate2를 로드 밸런서 LoadBalancer-1에 연결합니다.

```
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
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 AttachLoadBalancerTlsCertificate 섹션을 참조하세요.

attach-static-ip

다음 코드 예시에서는 attach-static-ip의 사용 방법을 보여줍니다.

AWS CLI

인스턴스에 고정 IP 연결

다음 `attach-static-ip` 예시에서는 정적 IP `StaticIp-1`을 인스턴스 `MEAN-1`에 연결합니다.

```
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
```

```
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [AttachStaticIp](https://docs.aws.amazon.com/goto/aws-cli/lightsail-2016-11-28/AttachStaticIp) 섹션을 참조하세요.

`close-instance-public-ports`

다음 코드 예시에서는 `close-instance-public-ports`의 사용 방법을 보여줍니다.

AWS CLI

인스턴스의 방화벽 포트를 닫으려면

다음 `close-instance-public-ports` 예시에서는 인스턴스 `MEAN-2`의 TCP 포트 `22`를 닫습니다.

```
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
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [CloseInstancePublicPorts](https://docs.aws.amazon.com/goto/aws-cli/lightsail-2016-11-28/CloseInstancePublicPorts) 섹션을 참조하세요.

copy-snapshot

다음 코드 예시에서는 copy-snapshot의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 동일한 AWS 리전 내에서 스냅샷 복사

다음 copy-snapshot 예시에서는 인스턴스 스냅샷 MEAN-1-1571075291을 동일한 AWS 리전 us-west-2 내에서 인스턴스 스냅샷 MEAN-1-Copy로서 복사합니다.

```
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 개발자 안내서의 [Copying snapshots from one AWS Region to another in Amazon Lightsail](https://docs.aws.amazon.com/lightsail/latest/userguide/amazon-lightsail-copying-snapshots-from-one-region-to-another.html) 섹션을 참조하세요.

예시 2: 한 리전에서 다른 AWS 리전으로 스냅샷 복사

다음 copy-snapshot 예시에서는 인스턴스 스냅샷 MEAN-1-1571075291을 AWS 리전 us-west-2에서 us-east-1로 인스턴스 스냅샷 MEAN-1-1571075291-Copy로서 복사합니다.

```
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 개발자 안내서의 Copying snapshots from one AWS Region to another in Amazon Lightsail 섹션을 참조하세요.

예시 3: 동일한 AWS 리전 내에서 자동 스냅샷을 복사하는 방법

다음 copy-snapshot 예시에서는 인스턴스 WordPress-1의 자동 스냅샷 2019-10-14를 AWS 리전 us-west-2에서 수동 스냅샷 WordPress-1-10142019로서 복사합니다.

```
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 개발자 안내서의 [Keeping automatic snapshots of instances or disks in Amazon Lightsail](#) 섹션을 참조하세요.

예시 4: 한 리전에서 다른 AWS 리전으로 자동 스냅샷 복사

다음 `copy-snapshot` 예시에서는 인스턴스 `WordPress-1`의 자동 스냅샷 `2019-10-14`를 AWS 리전 `us-west-2`에서 `us-east-1`로 수동 스냅샷 `WordPress-1-10142019`로서 복사합니다.

```
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 개발자 안내서의 Keeping automatic snapshots of instances or disks in Amazon Lightsail 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CopySnapshot 섹션을 참조하세요.

create-disk-from-snapshot

다음 코드 예시에서는 `create-disk-from-snapshot`의 사용 방법을 보여줍니다.

AWS CLI

디스크 스냅샷에서 디스크를 생성하는 방법

다음 `create-disk-from-snapshot` 예시에서는 지정된 블록 스토리지 디스크 스냅샷에서 `Disk-2`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 Creating a block storage disk from a snapshot in Amazon Lightsail 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateDiskFromSnapshot 섹션을 참조하세요.

create-disk-snapshot

다음 코드 예시에서는 create-disk-snapshot의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 디스크 스냅샷 생성

다음 create-disk-snapshot 예시에서는 지정된 블록 스토리지 디스크의 DiskSnapshot-1이라는 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
        }
    ]
}
```

예시 2: 인스턴스의 시스템 디스크 스냅샷 생성

다음 `create-disk-snapshot` 예시에서는 지정된 인스턴스 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 Snapshots in Amazon Lightsail 섹션 및 Creating a snapshot of an instance root volume in Amazon Lightsail 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateDiskSnapshot 섹션을 참조하세요.

`create-disk`

다음 코드 예시에서는 `create-disk`의 사용 방법을 보여줍니다.

AWS CLI

블록 스토리지 디스크를 생성하는 방법

다음 `create-disk` 예시에서는 32GB의 스토리지 공간이 있는 지정된 AWS 리전 및 가용 영역에 블록 스토리지 디스크 `Disk-1`을 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
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [CreateDisk](https://docs.aws.amazon.com/goto/aws-cli/lightsail-2016-11-28/CreateDisk) 섹션을 참조하세요.

`create-domain-entry`

다음 코드 예시에서는 `create-domain-entry`의 사용 방법을 보여줍니다.

AWS CLI

도메인 항목을 생성하는 방법(DNS 레코드)

다음 `create-domain-entry` 예시에서는 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. 그러지 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 Lightsail 개발자 안내서의 DNS in Amazon Lightsail 섹션 및 Creating a DNS zone to manage your domain's DNS records in Amazon Lightsail 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateDomainEntry 섹션을 참조하세요.

create-domain

다음 코드 예시에서는 create-domain의 사용 방법을 보여줍니다.

AWS CLI

도메인(DNS 영역)을 생성하는 방법

다음 create-domain 예시에서는 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. 그러지 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 Lightsail 개발자 안내서의 DNS in Amazon Lightsail 섹션 및 Creating a DNS zone to manage your domain's DNS records in Amazon Lightsail 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateDomain 섹션을 참조하세요.

`create-instance-snapshot`

다음 코드 예시에서는 `create-instance-snapshot`의 사용 방법을 보여줍니다.

AWS CLI

인스턴스의 스냅샷 생성

다음 `create-instance-snapshot` 예시에서는 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
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateInstanceSnapshot 섹션을 참조하세요.

create-instances-from-snapshot

다음 코드 예시에서는 `create-instances-from-snapshot`의 사용 방법을 보여줍니다.

AWS CLI

스냅샷에서 인스턴스 생성

다음 `create-instances-from-snapshot` 예시에서는 12 USD 번들을 사용하여 지정된 AWS 리전 및 가용 영역의 지정된 인스턴스 스냅샷에서 인스턴스를 생성합니다.

참고: 지정된 번들은 스냅샷을 생성하는 데 사용된 원본 소스 인스턴스의 번들과 사양이 같거나 높아야 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
        }
    ]
}
```


- API 세부 정보는 AWS CLI 명령 참조의 [CreateInstancesFromSnapshot](#) 섹션을 참조하세요.

`create-instances`

다음 코드 예시에서는 `create-instances`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 단일 인스턴스 생성

다음 `create-instances` 예시에서는 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
        }
    ]
}
```

예시 2: 한 번에 여러 인스턴스 생성

다음 `create-instances` 예시에서는 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
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateInstances 섹션을 참조하세요.

create-key-pair

다음 코드 예시에서는 create-key-pair의 사용 방법을 보여줍니다.

AWS CLI

키 페어 생성

다음 create-key-pair 예시에서는 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 가져올 수 없으므로 프라이빗 키 base64 값을 안전한 위치에 복사하여 붙여 넣습니다.

```
{
    "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
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateKeyPair를 참조하세요.

create-load-balancer-tls-certificate

다음 코드 예시에서는 create-load-balancer-tls-certificate의 사용 방법을 보여줍니다.

AWS CLI

로드 밸런서의 TLS 인증서 생성

다음 create-load-balancer-tls-certificate 예시에서는 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
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateLoadBalancerTlsCertificate 섹션을 참조하세요.

create-load-balancer

다음 코드 예시에서는 `create-load-balancer`의 사용 방법을 보여줍니다.

AWS CLI

로드 밸런서 생성

다음 `create-load-balancer` 예시에서는 TLS 인증서를 사용하여 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 load balancers 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateLoadBalancer를 참조하세요.

`create-relational-database-from-snapshot`

다음 코드 예시에서는 `create-relational-database-from-snapshot`의 사용 방법을 보여줍니다.

AWS CLI

스냅샷에서 관리형 데이터베이스 생성

다음 `create-relational-database-from-snapshot` 예시에서는 15 USD 표준 데이터베이스 번들을 사용하여 지정된 AWS 리전 및 가용 영역의 지정된 스냅샷에서 관리형 데이터베이스를 생성합니다. 참고: 지정된 번들은 스냅샷을 생성하는 데 사용된 원본 소스 데이터베이스의 번들과 사양이 같거나 높아야 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
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateRelationalDatabaseFromSnapshot 섹션을 참조하세요.

`create-relational-database-snapshot`

다음 코드 예시에서는 `create-relational-database-snapshot`의 사용 방법을 보여줍니다.

AWS CLI

관리형 데이터베이스의 스냅샷 생성

다음 `create-relational-database-snapshot` 예시에서는 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
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [CreateRelationalDatabaseSnapshot] 섹션을 참조하세요.

create-relational-database

다음 코드 예시에서는 `create-relational-database`의 사용 방법을 보여줍니다.

AWS CLI

관리형 데이터베이스를 생성하는 방법

다음 `create-relational-database` 예시에서는 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
        }
    ]
}
```


- API 세부 정보는 AWS CLI 명령 참조의 CreateRelationalDatabase 섹션을 참조하세요.

`delete-auto-snapshot`

다음 코드 예시에서는 `delete-auto-snapshot`의 사용 방법을 보여줍니다.

AWS CLI

자동 스냅샷 삭제

다음 `delete-auto-snapshot` 예시에서는 인스턴스 `WordPress-1`의 자동 스냅샷 `2019-10-10`을 삭제합니다.

```
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 개발자 안내서의 [Amazon Lightsail에서 인스턴스 또는 디스크의 자동 스냅샷 삭제]를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteAutoSnapshot] 섹션을 참조하세요.

delete-disk-snapshot

다음 코드 예시에서는 `delete-disk-snapshot`의 사용 방법을 보여줍니다.

AWS CLI

블록 스토리지 디스크의 스냅샷을 삭제하는 방법

다음 `delete-disk-snapshot` 예시에서는 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
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteDiskSnapshot 섹션을 참조하세요.

delete-disk

다음 코드 예시에서는 delete-disk의 사용 방법을 보여줍니다.

AWS CLI

블록 스토리지 디스크를 삭제하는 방법

다음 delete-disk 예시에서는 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
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteDisk](#) 섹션을 참조하세요.

`delete-domain-entry`

다음 코드 예시에서는 `delete-domain-entry`의 사용 방법을 보여줍니다.

AWS CLI

도메인 항목을 삭제하는 방법(DNS 레코드)

다음 `delete-domain-entry` 예시에서는 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. 그러지 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
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteDomainEntry 섹션을 참조하세요.

delete-domain

다음 코드 예시에서는 delete-domain의 사용 방법을 보여줍니다.

AWS CLI

도메인(DNS 영역)을 삭제하는 방법

다음 delete-domain 예시에서는 지정된 도메인과 도메인의 모든 항목(DNS 레코드)을 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. 그러지 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
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteDomain 섹션을 참조하세요.

delete-instance-snapshot

다음 코드 예시에서는 delete-instance-snapshot의 사용 방법을 보여줍니다.

AWS CLI

제목

다음 delete-instance-snapshot 예시에서는 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
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteInstanceSnapshot 섹션을 참조하세요.

delete-instance

다음 코드 예시에서는 delete-instance의 사용 방법을 보여줍니다.

AWS CLI

인스턴스 삭제

다음 delete-instance 예시에서는 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
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteInstance 섹션을 참조하세요.

delete-key-pair

다음 코드 예시에서는 delete-key-pair의 사용 방법을 보여줍니다.

AWS CLI

키 페어 삭제

다음 delete-key-pair 예시에서는 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
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteKeyPair를 참조하세요.

delete-known-host-keys

다음 코드 예시에서는 delete-known-host-keys의 사용 방법을 보여줍니다.

AWS CLI

인스턴스에서 알려진 호스트 키를 삭제하는 방법

다음 delete-known-host-keys 예시에서는 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 안내서의 Troubleshooting connection issues with the Amazon Lightsail browser-based SSH or RDP client 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteKnownHostKeys 섹션을 참조하세요.

`delete-load-balancer-tls-certificate`

다음 코드 예시에서는 `delete-load-balancer-tls-certificate`의 사용 방법을 보여줍니다.

AWS CLI

로드 밸런서의 TLS 인증서 삭제

다음 `delete-load-balancer-tls-certificate` 예시에서는 지정된 로드 밸런서에서 지정된 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
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteLoadBalancerTlsCertificate 섹션을 참조하세요.

delete-load-balancer

다음 코드 예시에서는 delete-load-balancer의 사용 방법을 보여줍니다.

AWS CLI

로드 밸런서 삭제

다음 delete-load-balancer 예시에서는 지정된 로드 밸런서와 연결된 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 안내서의 제목을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteLoadBalancer를 참조하세요.

delete-relational-database-snapshot

다음 코드 예시에서는 delete-relational-database-snapshot의 사용 방법을 보여줍니다.

AWS CLI

관리형 데이터베이스의 스냅샷을 삭제하는 방법

다음 delete-relational-database-snapshot 예시에서는 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
        }
    ]
```

```
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteRelationalDatabaseSnapshot 섹션을 참조하세요.

delete-relational-database

다음 코드 예시에서는 `delete-relational-database`의 사용 방법을 보여줍니다.

AWS CLI

관리형 데이터베이스를 삭제하는 방법

다음 `delete-relational-database` 예시에서는 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
        }
    ]
}
```


- API 세부 정보는 AWS CLI 명령 참조의 DeleteRelationalDatabase 섹션을 참조하세요.

detach-static-ip

다음 코드 예시에서는 detach-static-ip의 사용 방법을 보여줍니다.

AWS CLI

인스턴스에서 정적 IP를 분리하는 방법

다음 detach-static-ip 예시에서는 연결된 인스턴스에서 정적 IP StaticIp-1을 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
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DetachStaticIp 섹션을 참조하세요.

`get-active-names`

다음 코드 예시에서는 `get-active-names`의 사용 방법을 보여줍니다.

AWS CLI

활성 리소스 이름을 가져오는 방법

다음 get-active-names 예시에서는 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"
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetActiveNames 섹션을 참조하세요.

get-auto-snapshots

다음 코드 예시에서는 get-auto-snapshots의 사용 방법을 보여줍니다.

AWS CLI

인스턴스에 대해 사용 가능한 자동 스냅샷을 가져오는 방법

다음 get-auto-snapshots 예시에서는 인스턴스 WordPress-1에 사용 가능한 자동 스냅샷을 반환합니다.

```
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 개발자 안내서의 Keeping automatic snapshots of instances or disks in Amazon Lightsail 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetAutoSnapshots 섹션을 참조하세요.

get-blueprints

다음 코드 예시에서는 `get-blueprints`의 사용 방법을 보여줍니다.

AWS CLI

새 인스턴스의 청사진을 가져오는 방법

다음 `get-blueprints` 예시에서는 Amazon Lightsail에서 새 인스턴스를 생성하는 데 사용할 수 있는 모든 사용 가능한 블루프린트의 세부 정보를 표시합니다.

```
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"
        },
        ...
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetBlueprints 섹션을 참조하세요.

get-bundles

다음 코드 예시에서는 get-bundles의 사용 방법을 보여줍니다.

AWS CLI

새 인스턴스에 대한 번들을 가져오는 방법

다음 get-bundles 예시에서는 Amazon Lightsail에서 새 인스턴스를 생성하는 데 사용할 수 있는 모든 사용 가능한 번들의 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"
                ]
            },
            ...
            }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetBundles 섹션을 참조하세요.

get-cloud-formation-stack-records

다음 코드 예시에서는 `get-cloud-formation-stack-records`의 사용 방법을 보여줍니다.

AWS CLI

CloudFormation 스택 레코드 및 관련 스택을 가져오는 방법

다음 `get-cloud-formation-stack-records` 예시에서는 내보낸 Amazon Lightsail 스냅샷에서 Amazon EC2 리소스를 생성하는 데 사용되는 CloudFormation 스택 레코드 및 관련 스택의 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"
            }
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetCloudFormationStackRecords 섹션을 참조하세요.

get-disk-snapshot

다음 코드 예시에서는 get-disk-snapshot의 사용 방법을 보여줍니다.

AWS CLI

디스크 스냅샷의 정보 가져오기

다음 get-disk-snapshot 예시에서는 디스크 스냅샷 Disk-1-1566839161의 세부 정보를 표시합니다.

```
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 안내서의 제목을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetDiskSnapshot 섹션을 참조하세요.

get-disk-snapshots

다음 코드 예시에서는 get-disk-snapshots의 사용 방법을 보여줍니다.

AWS CLI

모든 디스크 스냅샷의 정보 가져오기

다음 get-disk-snapshots 예시에서는 구성된 AWS 리전 내 모든 디스크 스냅샷의 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
        }
    ]
```

```
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetDiskSnapshots 섹션을 참조하세요.

get-disk

다음 코드 예시에서는 get-disk의 사용 방법을 보여줍니다.

AWS CLI

블록 스토리지 디스크에 대한 정보를 가져오는 방법

다음 get-disk 예시에서는 디스크 Disk-1의 세부 정보를 표시합니다.

```
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 안내서의 제목을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetDisk 섹션을 참조하세요.

get-disks

다음 코드 예시에서는 get-disks의 사용 방법을 보여줍니다.

AWS CLI

모든 블록 스토리지 디스크에 대한 정보를 가져오는 방법

다음 get-disks 예시에서는 구성된 AWS 리전 내 모든 디스크의 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"
            }
        ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetDisks 섹션을 참조하세요.

get-domain

다음 코드 예시에서는 get-domain의 사용 방법을 보여줍니다.

AWS CLI

도메인 정보 가져오기

다음 get-domain 예시에서는 도메인 example.com의 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. 그러지 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"
            }
        ]
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetDomain 섹션을 참조하세요.

get-domains

다음 코드 예시에서는 get-domains의 사용 방법을 보여줍니다.

AWS CLI

모든 도메인 정보 가져오기

다음 get-domains 예시에서는 구성된 AWS 리전 내 모든 도메인의 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. 그러지 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"
                }
            ]
        }
    ]
}
```


- API 세부 정보는 AWS CLI 명령 참조의 GetDomains 섹션을 참조하세요.

`get-export-snapshot-record`

다음 코드 예시에서는 `get-export-snapshot-record`의 사용 방법을 보여줍니다.

AWS CLI

Amazon EC2로 내보낸 스냅샷 레코드를 가져오는 방법

다음 `get-export-snapshot-record` 예시에서는 Amazon EC2로 내보낸 Amazon Lightsail 인스턴스 또는 디스크 스냅샷의 세부 정보를 표시합니다.

```
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"
            }
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetExportSnapshotRecord 섹션을 참조하세요.

get-instance-access-details

다음 코드 예시에서는 get-instance-access-details의 사용 방법을 보여줍니다.

AWS CLI

인스턴스의 호스트 키 정보를 가져오는 방법

다음 get-instance-access-details 예시에서는 인스턴스 WordPress_Multisite-1의 호스트 키 정보를 표시합니다.

```
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"
            }
        ]
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [GetInstanceAccessDetails](#) 섹션을 참조하세요.

`get-instance-metric-data`

다음 코드 예시에서는 `get-instance-metric-data`의 사용 방법을 보여줍니다.

AWS CLI

인스턴스의 지표 데이터를 가져오는 방법

다음 `get-instance-metric-data` 예시에서는 인스턴스 `MEAN-1`과 같이 `1571342400` 및 `1571428800` 사이의 `7200`초(2시간)당 `CPUUtilization`의 평균 백분율을 반환합니다.

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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetInstanceMetricData 섹션을 참조하세요.

get-instance-port-states

다음 코드 예시에서는 get-instance-port-states의 사용 방법을 보여줍니다.

AWS CLI

인스턴스에 대한 방화벽 정보를 가져오는 방법

다음 get-instance-port-states 예시에서는 인스턴스 MEAN-1에 대해 구성된 방화벽 포트를 반환합니다.

```
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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [GetInstancePortStates](https://docs.aws.amazon.com/goto/aws-cli/lightsail-2016-11-28/GetInstancePortStates) 섹션을 참조하세요.

`get-instance-snapshot`

다음 코드 예시에서는 `get-instance-snapshot`의 사용 방법을 보여줍니다.

AWS CLI

특정 인스턴스 스냅샷의 정보 가져오기

다음 `get-instance-snapshot` 예시에서는 지정된 인스턴스 스냅샷의 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
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetInstanceSnapshot 섹션을 참조하세요.

get-instance-snapshots

다음 코드 예시에서는 get-instance-snapshots의 사용 방법을 보여줍니다.

AWS CLI

모든 인스턴스 스냅샷에 대한 정보를 가져오는 방법

다음 get-instance-snapshots 예시에서는 구성된 AWS 리전 내 모든 인스턴스 스냅샷의 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
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetInstanceSnapshots 섹션을 참조하세요.

`get-instance-state`

다음 코드 예시에서는 `get-instance-state`의 사용 방법을 보여줍니다.

AWS CLI

인스턴스 상태에 대한 정보를 가져오는 방법

다음 `get-instance-state` 예시에서는 지정된 인스턴스의 상태를 반환합니다.

```
aws lightsail get-instance-state \
    --instance-name MEAN-1
```

출력:

```
{
    "state": {
        "code": 16,
        "name": "running"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetInstanceState 섹션을 참조하세요.

`get-instance`

다음 코드 예시에서는 `get-instance`의 사용 방법을 보여줍니다.

AWS CLI

인스턴스 정보 가져오기

다음 `get-instance` 예시에서는 인스턴스 `MEAN-1`의 세부 정보를 표시합니다.

```
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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetInstance 섹션을 참조하세요.

get-instances

다음 코드 예시에서는 `get-instances`의 사용 방법을 보여줍니다.

AWS CLI

모든 인스턴스 정보 가져오기

다음 `get-instances` 예시에서는 구성된 AWS 리전 내 모든 인스턴스의 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetInstances 섹션을 참조하세요.

get-key-pair

다음 코드 예시에서는 get-key-pair의 사용 방법을 보여줍니다.

AWS CLI

키 페어 정보 가져오기

다음 get-key-pair 예시에서는 지정된 키 페어의 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetKeyPair 섹션을 참조하세요.

get-key-pairs

다음 코드 예시에서는 get-key-pairs의 사용 방법을 보여줍니다.

AWS CLI

모든 키 페어의 정보 가져오기

다음 get-key-pairs 예시에서는 구성된 AWS 리전 내 모든 키 페어의 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetKeyPairs 섹션을 참조하세요.

get-load-balancer-tls-certificates

다음 코드 예시에서는 `get-load-balancer-tls-certificates`의 사용 방법을 보여줍니다.

AWS CLI

로드 밸런서의 TLS 인증서 정보 가져오기

다음 `get-load-balancer-tls-certificates` 예시에서는 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"
            ]
        }
    ]
}
```


- API 세부 정보는 AWS CLI 명령 참조의 [GetLoadBalancerTlsCertificates](https://docs.aws.amazon.com/goto/aws-cli/lightsail-2016-11-28/GetLoadBalancerTlsCertificates) 섹션을 참조하세요.

get-load-balancer

다음 코드 예시에서는 get-load-balancer의 사용 방법을 보여줍니다.

AWS CLI

로드 밸런서 정보 가져오기

다음 get-load-balancer 예시에서는 지정된 로드 밸런서의 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"
        }
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetLoadBalancer 섹션을 참조하세요.

get-load-balancers

다음 코드 예시에서는 get-load-balancers의 사용 방법을 보여줍니다.

AWS CLI

모든 로드 밸런서에 대한 정보를 가져오는 방법

다음 get-load-balancers 예시에서는 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"
            }
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetLoadBalancers 섹션을 참조하세요.

get-operation

다음 코드 예시에서는 get-operation의 사용 방법을 보여줍니다.

AWS CLI

단일 작업에 대한 정보를 가져오는 방법

다음 get-operation 예시에서는 지정된 작업의 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
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetOperation 섹션을 참조하세요.

get-operations-for-resource

다음 코드 예시에서는 get-operations-for-resource의 사용 방법을 보여줍니다.

AWS CLI

리소스에 대한 모든 작업을 가져오는 방법

다음 get-operations-for-resource 예시에서는 지정된 리소스에 대한 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
            },
            ...
            }
        ]
}
```


- API 세부 정보는 AWS CLI 명령 참조의 GetOperationsForResource 섹션을 참조하세요.

`get-operations`

다음 코드 예시에서는 `get-operations`의 사용 방법을 보여줍니다.

AWS CLI

모든 작업의 정보 가져오기

다음 `get-operations` 예시에서는 구성된 AWS 리전 내 모든 작업의 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
        },
```

```
        ...
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetOperations 섹션을 참조하세요.

get-regions

다음 코드 예시는 get-regions의 사용 방법을 보여 줍니다.

AWS CLI

에 대한 모든 AWS 리전을 가져오는 방법 Amazon Lightsail

다음 get-regions 예시에서는 Amazon Lightsail의 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": []
        },
        ...
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [GetRegions](#) 섹션을 참조하세요.

get-relational-database-blueprints

다음 코드 예시에서는 `get-relational-database-blueprints`의 사용 방법을 보여줍니다.

AWS CLI

새 관계형 데이터베이스에 대한 청사진을 가져오는 방법

다음 `get-relational-database-blueprints` 예시에서는 Amazon Lightsail에서 새 관계형 데이터베이스를 생성하는 데 사용할 수 있는 모든 사용 가능한 관계형 데이터베이스 블루프린트의 세부 정보를 표시합니다.

```
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
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetRelationalDatabaseBlueprints 섹션을 참조하세요.

`get-relational-database-bundles`

다음 코드 예시에서는 `get-relational-database-bundles`의 사용 방법을 보여줍니다.

AWS CLI

새 관계형 데이터베이스의 번들을 가져오는 방법

다음 `get-relational-database-bundles` 예시에서는 Amazon Lightsail에서 새 관계형 데이터베이스를 생성하는 데 사용할 수 있는 모든 사용 가능한 관계형 데이터베이스 번들의 세부 정보를 표시합니다. `--include-inactive` 플래그가 명령에 지정되지 않았으므로 응답에 비활성 번들이 포함되지 않는다는 사실에 유의하세요. 비활성 번들은 새 관계형 데이터베이스를 생성하는 데 사용할 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 Creating a database in Amazon Lightsail 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetRelationalDatabaseBundles 섹션을 참조하세요.

`get-relational-database-events`

다음 코드 예시에서는 `get-relational-database-events`의 사용 방법을 보여줍니다.

AWS CLI

관계형 데이터베이스의 이벤트를 가져오는 방법

다음 `get-relational-database-events` 예시에서는 지정된 관계형 데이터베이스에서 지난 17시간(1020분) 동안 발생한 이벤트의 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"
            ]
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetRelationalDatabaseEvents 섹션을 참조하세요.

`get-relational-database-log-events`

다음 코드 예시에서는 `get-relational-database-log-events`의 사용 방법을 보여줍니다.

AWS CLI

관계형 데이터베이스의 로그 이벤트를 가져오는 방법

다음 `get-relational-database-log-events` 예시에서는 관계형 데이터베이스 `Database1`에서 `1570733176` 및 `1571597176` 사이에 기록된 지정된 로그에 대한 세부 정보를 표시합니다. 반환된 정보는 `head`에서 시작하도록 구성됩니다.

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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [GetRelationalDatabaseLogEvents](#) 섹션을 참조하세요.

`get-relational-database-log-streams`

다음 코드 예시에서는 `get-relational-database-log-streams`의 사용 방법을 보여줍니다.

AWS CLI

관계형 데이터베이스의 로그 스트림을 가져오는 방법

다음 `get-relational-database-log-streams` 예시에서는 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"
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [GetRelationalDatabaseLogStreams](#) 섹션을 참조하세요.

get-relational-database-master-user-password

다음 코드 예시에서는 get-relational-database-master-user-password의 사용 방법을 보여줍니다.

AWS CLI

관계형 데이터베이스의 마스터 사용자 암호 가져오기

다음 get-relational-database-master-user-password 예시에서는 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetRelationalDatabaseMasterUserPassword 섹션을 참조하세요.

get-relational-database-metric-data

다음 코드 예시에서는 get-relational-database-metric-data의 사용 방법을 보여줍니다.

AWS CLI

관계형 데이터베이스의 지표 데이터를 가져오는 방법

다음 get-relational-database-metric-data 예시에서는 관계형 데이터베이스 Database1에서 1570733176 및 1571597176 사이에 24시간(86400초) 동안 기록된 지표 DatabaseConnections의 합계를 반환합니다.

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"
            }
        ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetRelationalDatabaseMetricData 섹션을 참조하세요.

`get-relational-database-parameters`

다음 코드 예시에서는 `get-relational-database-parameters`의 사용 방법을 보여줍니다.

AWS CLI

관계형 데이터베이스의 파라미터를 가져오는 방법

다음 `get-relational-database-parameters` 예시에서는 지정된 관계형 데이터베이스에 사용 가능한 모든 파라미터의 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 개발 안내서의 Updating database parameters in Amazon Lightsail 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetRelationalDatabaseParameters 섹션을 참조하세요.

get-relational-database-snapshot

다음 코드 예시에서는 get-relational-database-snapshot의 사용 방법을 보여줍니다.

AWS CLI

관계형 데이터베이스 스냅샷의 정보 가져오기

다음 get-relational-database-snapshot 예시에서는 지정된 관계형 데이터베이스 스냅샷의 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [GetRelationalDatabaseSnapshot](https://docs.aws.amazon.com/goto/aws-cli/lightsail-2016-11-28/GetRelationalDatabaseSnapshot) 섹션을 참조하세요.

get-relational-database-snapshots

다음 코드 예시에서는 get-relational-database-snapshots의 사용 방법을 보여줍니다.

AWS CLI

모든 관계형 데이터베이스 스냅샷의 정보 가져오기

다음 get-relational-database-snapshots 예시에서는 구성된 AWS 리전 내 모든 관계형 데이터베이스 스냅샷의 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetRelationalDatabaseSnapshots 섹션을 참조하세요.

get-relational-database

다음 코드 예시에서는 get-relational-database의 사용 방법을 보여줍니다.

AWS CLI

관계형 데이터베이스의 정보 가져오기

다음 `get-relational-database` 예시에서는 지정된 관계형 데이터베이스의 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": []
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetRelationalDatabase 섹션을 참조하세요.

get-relational-databases

다음 코드 예시에서는 `get-relational-databases`의 사용 방법을 보여줍니다.

AWS CLI

모든 관계형 데이터베이스의 정보 가져오기

다음 `get-relational-databases` 예시에서는 구성된 AWS 리전 내 모든 관계형 데이터베이스의 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": []
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetRelationalDatabases 섹션을 참조하세요.

get-static-ip

다음 코드 예시에서는 get-static-ip의 사용 방법을 보여줍니다.

AWS CLI

고정 IP 정보 가져오기

다음 get-static-ip 예시에서는 지정된 고정 IP의 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
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetStaticIp 섹션을 참조하세요.

get-static-ips

다음 코드 예시에서는 get-static-ips의 사용 방법을 보여줍니다.

AWS CLI

모든 고정 IP의 정보 가져오기

다음 get-static-ips 예시에서는 구성된 AWS 리전 내 모든 정적 IP의 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
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetStaticIps 섹션을 참조하세요.

`is-vpc-peered`

다음 코드 예시는 `is-vpc-peered`의 사용 방법을 보여 줍니다.

AWS CLI

Amazon Lightsail 가상 프라이빗 클라우드가 피어링되었는지 확인하는 방법

다음 `is-vpc-peered` 예시에서는 지정된 AWS 리전의 Amazon Lightsail 가상 프라이빗 클라우드 (VPC)의 피어링 상태를 반환합니다.

```
aws lightsail is-vpc-peered \
    --region us-west-2
```

출력:

```
{
```

```
    "isPeered": true
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [IsVpcPeered](#) 섹션을 참조하세요.

open-instance-public-ports

다음 코드 예시에서는 open-instance-public-ports의 사용 방법을 보여줍니다.

AWS CLI

인스턴스의 방화벽 포트를 열려면

다음 open-instance-public-ports 예시에서는 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
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [OpenInstancePublicPorts](#) 섹션을 참조하세요.

peer-vpc

다음 코드 예시는 peer-vpc의 사용 방법을 보여 줍니다.

AWS CLI

Amazon Lightsail 가상 프라이빗 클라우드 피어링

다음 peer-vpc 예시에서는 지정된 AWS 리전의 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
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 PeerVpc 섹션을 참조하세요.

reboot-instance

다음 코드 예시에서는 reboot-instance의 사용 방법을 보여줍니다.

AWS CLI

인스턴스 재부팅

다음 `reboot-instance` 예시에서는 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
        }
    ]
}
```


- API 세부 정보는 AWS CLI 명령 참조의 RebootInstance 섹션을 참조하세요.

`reboot-relational-database`

다음 코드 예시에서는 `reboot-relational-database`의 사용 방법을 보여줍니다.

AWS CLI

관계형 데이터베이스 재부팅

다음 `reboot-relational-database` 예시에서는 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
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 RebootRelationalDatabase 섹션을 참조하세요.

`release-static-ip`

다음 코드 예시에서는 `release-static-ip`의 사용 방법을 보여줍니다.

AWS CLI

고정 IP 삭제

다음 `release-static-ip` 예시에서는 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
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ReleaseStaticIp 섹션을 참조하세요.

`start-instance`

다음 코드 예시에서는 `start-instance`의 사용 방법을 보여줍니다.

AWS CLI

인스턴스 시작

다음 `start-instance` 예시에서는 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
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [StartInstance](https://docs.aws.amazon.com/goto/aws-cli/lightsail-2016-11-28/StartInstance) 섹션을 참조하세요.

start-relational-database

다음 코드 예시에서는 `start-relational-database`의 사용 방법을 보여줍니다.

AWS CLI

관계형 데이터베이스 시작

다음 `start-relational-database` 예시에서는 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
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 StartRelationalDatabase 섹션을 참조하세요.

stop-instance

다음 코드 예시에서는 stop-instance의 사용 방법을 보여줍니다.

AWS CLI

인스턴스 중지

다음 stop-instance 예시에서는 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
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 StopInstance 섹션을 참조하세요.

stop-relational-database

다음 코드 예시에서는 `stop-relational-database`의 사용 방법을 보여줍니다.

AWS CLI

관계형 데이터베이스 중지

다음 `stop-relational-database` 예시에서는 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
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 StopRelationalDatabase 섹션을 참조하세요.

unpeer-vpc

다음 코드 예시는 `unpeer-vpc`의 사용 방법을 보여 줍니다.

AWS CLI

Amazon Lightsail 가상 프라이빗 클라우드를 피어링 취소하는 방법

다음 unpeer-vpc 예시에서는 지정된 AWS 리전의 Amazon Lightsail 가상 프라이빗 클라우드 (VPC)를 피어링 해제합니다.

```
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
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [UnpeerVpc](https://docs.aws.amazon.com/goto/aws-cli/lightsail-2016-11-28/UnpeerVpc) 섹션을 참조하세요.

AWS CLI를 사용한 Macie 예시

다음 코드 예시는 Macie와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`describe-buckets`

다음 코드 예시에서는 `describe-buckets`의 사용 방법을 보여줍니다.

AWS CLI

Amazon Macie가 계정에서 모니터링 및 분석하는 하나 이상의 S3 버킷에 대한 데이터 쿼리

다음 `describe-buckets` 예제에서는 이름이 amzn-s3-demo-bucket으로 시작되고 현재 AWS 리전에 있는 모든 S3 버킷에 대한 메타데이터를 쿼리합니다.

```
aws macie2 describe-buckets \
    --criteria '{"bucketName":{"prefix":"amzn-s3-demo-bucket"}}'
```

출력:

```
{
    "buckets": [
        {
            "accountId": "123456789012",
            "allowsUnencryptedObjectUploads": "FALSE",
            "automatedDiscoveryMonitoringStatus": "MONITORED",
            "bucketArn": "arn:aws:s3:::amzn-s3-demo-bucket1",
            "bucketCreatedAt": "2020-05-18T19:54:00+00:00",
            "bucketName": "amzn-s3-demo-bucket1",
            "classifiableObjectCount": 13,
            "classifiableSizeInBytes": 1592088,
            "jobDetails": {
                "isDefinedInJob": "TRUE",
                "isMonitoredByJob": "TRUE",
                "lastJobId": "08c81dc4a2f3377fae45c9ddaEXAMPLE",
                "lastJobRunTime": "2024-08-19T14:55:30.270000+00:00"
            },
            "lastAutomatedDiscoveryTime": "2024-10-22T19:11:25.364000+00:00",
            "lastUpdated": "2024-10-25T07:33:06.337000+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",
            "automatedDiscoveryMonitoringStatus": "MONITORED",
            "bucketArn": "arn:aws:s3:::amzn-s3-demo-bucket2",
            "bucketCreatedAt": "2020-11-25T18:24:38+00:00",
            "bucketName": "amzn-s3-demo-bucket2",
            "classifiableObjectCount": 8,
            "classifiableSizeInBytes": 133810,
            "jobDetails": {
                "isDefinedInJob": "TRUE",
                "isMonitoredByJob": "FALSE",
                "lastJobId": "188d4f6044d621771ef7d65f2EXAMPLE",
                "lastJobRunTime": "2024-07-09T19:37:11.511000+00:00"
            },
            "lastAutomatedDiscoveryTime": "2024-10-24T19:11:25.364000+00:00",
            "lastUpdated": "2024-10-25T07:33:06.337000+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 사용자 안내서의 S3 버킷 인벤토리 필터링을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeBuckets를 참조하세요.

AWS CLI를 사용한 Amazon Managed Grafana 예제

다음 코드 예제에서는 Amazon Managed Grafana에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여 줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

list-workspaces

다음 코드 예시는 list-workspaces의 사용 방법을 보여 줍니다.

AWS CLI

사용자 자격 증명에 지정된 리전에서 계정의 워크스페이스를 나열하는 방법

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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListWorkspaces를 참조하세요.

AWS CLI를 사용한 MediaConnect 예시

다음 코드 예시는 MediaConnect와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`add-flow-outputs`

다음 코드 예시에서는 `add-flow-outputs`의 사용 방법을 보여줍니다.

AWS CLI

흐름에 출력 추가

다음 add-flow-outputs 예시에서는 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 Elemental MediaConnect 사용자 안내서의 흐름에 출력 추가를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AddFlowOutputs를 참조하세요.

create-flow

다음 코드 예시에서는 `create-flow`의 사용 방법을 보여줍니다.

AWS CLI

흐름 생성

다음 `create-flow` 예시에서는 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 AWS Elemental MediaConnect 사용자 안내서의 흐름 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateFlow를 참조하세요.

delete-flow

다음 코드 예시에서는 delete-flow의 사용 방법을 보여줍니다.

AWS CLI

흐름 삭제

다음 delete-flow 예시에서는 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 Elemental MediaConnect 사용자 안내서의 흐름 삭제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteFlow를 참조하세요.

describe-flow

다음 코드 예시에서는 describe-flow의 사용 방법을 보여줍니다.

AWS CLI

흐름의 세부 정보 보기

다음 describe-flow 예시에서는 ARN, 가용 영역, 상태, 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 Elemental MediaConnect 사용자 안내서의 흐름 세부 정보 보기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeFlow를 참조하세요.

grant-flow-entitlements

다음 코드 예시에서는 `grant-flow-entitlements`의 사용 방법을 보여줍니다.

AWS CLI

흐름에 권한 부여

다음 `grant-flow-entitlements` 예시에서는 다른 AWS 계정과 콘텐츠를 공유할 수 있는 권한을 지정된 기존 흐름에 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 Elemental MediaConnect 사용자 안내서의 흐름에 대한 권한 부여를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GrantFlowEntitlements를 참조하세요.

list-entitlements

다음 코드 예시에서는 list-entitlements의 사용 방법을 보여줍니다.

AWS CLI

권한 목록 보기

다음 list-entitlements 예시에서는 계정에 부여된 모든 권한의 목록을 표시합니다.

```
aws mediaconnect list-entitlements
```

출력:

```
{
    "Entitlements": [
        {
            "EntitlementArn": "arn:aws:mediaconnect:us-
west-2:111122223333:entitlement:1-11aa22bb11aa22bb-3333cccc4444:MyEntitlement",
            "EntitlementName": "MyEntitlement"
        }
    ]
}
```

자세한 내용은 AWS Elemental MediaConnect API 참조의 ListEntitlements를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListEntitlements를 참조하세요.

`list-flows`

다음 코드 예시에서는 `list-flows`의 사용 방법을 보여줍니다.

AWS CLI

흐름 목록 보기

다음 `list-flows` 예시에서는 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 Elemental MediaConnect 사용자 안내서의 흐름 목록 보기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListFlows를 참조하세요.

list-tags-for-resource

다음 코드 예시에서는 list-tags-for-resource의 사용 방법을 보여줍니다.

AWS CLI

MediaConnect 리소스의 태그 나열

다음 list-tags-for-resource 예시에서는 지정된 MediaConnect 리소스에 연결된 태그 키와 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 Elemental MediaConnect API 참조의 ListTagsForResource, TagResource, UntagResource를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource를 참조하세요.

remove-flow-output

다음 코드 예시에서는 remove-flow-output의 사용 방법을 보여줍니다.

AWS CLI

흐름에서 출력 제거

다음 remove-flow-output 예시에서는 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 Elemental MediaConnect 사용자 안내서의 흐름에서 출력 제거를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RemoveFlowOutput을 참조하세요.

revoke-flow-entitlement

다음 코드 예시에서는 revoke-flow-entitlement의 사용 방법을 보여줍니다.

AWS CLI

권한 취소

다음 revoke-flow-entitlement 예시에서는 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 AWS Elemental MediaConnect 사용자 안내서의 권한 취소를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RevokeFlowEntitlement를 참조하세요.

start-flow

다음 코드 예시에서는 start-flow의 사용 방법을 보여줍니다.

AWS CLI

흐름 시작

다음 start-flow 예시에서는 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 AWS Elemental MediaConnect 사용자 안내서의 흐름 시작을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartFlow를 참조하세요.

stop-flow

다음 코드 예시에서는 stop-flow의 사용 방법을 보여줍니다.

AWS CLI

흐름 중지

다음 stop-flow 예시에서는 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 AWS Elemental MediaConnect 사용자 안내서의 흐름 중지를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StopFlow를 참조하세요.

tag-resource

다음 코드 예시에서는 `tag-resource`의 사용 방법을 보여줍니다.

AWS CLI

MediaConnect 리소스에 태그 추가

다음 `tag-resource` 예시에서는 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 Elemental MediaConnect API 참조의 ListTagsForResource, TagResource, UntagResource를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

untag-resource

다음 코드 예시에서는 `untag-resource`의 사용 방법을 보여줍니다.

AWS CLI

MediaConnect 리소스에서 태그 제거

다음 untag-resource 예시에서는 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 Elemental MediaConnect API 참조의 ListTagsForResource, TagResource, UntagResource를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

update-flow-entitlement

다음 코드 예시에서는 update-flow-entitlement의 사용 방법을 보여줍니다.

AWS CLI

권한 업데이트

다음 update-flow-entitlement 예시에서는 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 Elemental MediaConnect 사용자 안내서의 권한 업데이트를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateFlowEntitlement를 참조하세요.

update-flow-output

다음 코드 예시에서는 update-flow-output의 사용 방법을 보여줍니다.

AWS CLI

흐름의 출력 업데이트

다음 update-flow-output 예시에서는 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 Elemental MediaConnect 사용자 안내서의 흐름에 대한 출력 업데이트를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateFlowOutput을 참조하세요.

update-flow-source

다음 코드 예시에서는 update-flow-source의 사용 방법을 보여줍니다.

AWS CLI

기존 흐름의 소스 업데이트

다음 update-flow-source 예시에서는 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 Elemental MediaConnect 사용자 안내서의 흐름 소스 업데이트를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateFlowSource를 참조하세요.

AWS CLI를 사용한 MediaConvert 예시

다음 코드 예시는 MediaConvert와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

cancel-job

다음 코드 예시에서는 `cancel-job` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

대기열에 있는 작업을 취소하는 방법

다음 cancel-job 예시에서는 ID 1234567891234-abc123를 사용하여 작업을 취소합니다. 서비스 처리가 시작된 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 Elemental MediaConvert 사용 설명서의 Working with AWS Elemental MediaConvert Jobs를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CancelJob 섹션을 참조하세요.

create-job-template

다음 코드 예시에서는 create-job-template 코드를 사용하는 방법을 보여줍니다.

AWS CLI

작업 템플릿 생성

다음 create-job-template 예시에서는 시스템에 있는 job-template.json 파일에 지정된 트랜스코딩 설정으로 작업 템플릿을 만듭니다.

```
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 명령을 사용하여 작업 템플릿 JSON 파일을 만든 다음 파일을 수정하는 경우 객체를 제거하되 그 안에 설정 하위 객체는 그대로 유지합니다. LastUpdated, Arn, Type, CreatedAt의 키-값 쌍을 제거해야 합니다. JSON 파일이나 명령줄에서 카테고리, 설명, 이름, 대기열을 지정할 수 있습니다.

계정별 엔드포인트를 가져오려면 describe-endpoints를 사용하거나 엔드포인트 없이 명령을 전송하세요. 서비스가 오류와 엔드포인트를 반환합니다.

요청이 성공하면 서비스에서 사용자가 만든 작업 템플릿에 대한 JSON 사양을 반환합니다.

자세한 내용은 AWS Elemental MediaConvert 사용 설명서의 Working with AWS Elemental MediaConvert Job Templates를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateJobTemplate 섹션을 참조하세요.

create-job

다음 코드 예시에서는 create-job 코드를 사용하는 방법을 보여줍니다.

AWS CLI

작업 생성

다음 create-job 예시에서는 명령을 보내는 소스 시스템에 있는 파일 job.json에 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 작업 설정을 선택한 다음, 작업 섹션 하단에서 작업 JSON 표시를 선택하여 JSON 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 Elemental MediaConvert 사용자 안내서의 AWS Elemental MediaConvert 작업 처리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateJob을 참조하세요.

create-preset

다음 코드 예시에서는 create-preset 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용자 지정 출력 사전 설정 생성

다음 create-preset 예시에서는 preset.json 파일에 지정된 출력 설정을 기반으로 사용자 지정 출력 프리셋을 생성합니다. 카테고리, 설명, 이름은 JSON 파일이나 명령줄에서 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-preset를 사용하여 사전 설정 JSON 파일을 만든 다음 출력 파일을 수정하는 경우 LastUpdated, Arn, Type, CreatedAt 키-값 쌍을 제거해야 합니다.

계정별 엔드포인트를 가져오려면 describe-endpoints를 사용하거나 엔드포인트 없이 명령을 전송하세요. 서비스가 오류와 엔드포인트를 반환합니다.

자세한 내용은 AWS Elemental MediaConvert 사용 설명서의 Working with AWS Elemental MediaConvert Output Presets를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreatePreset 섹션을 참조하세요.

create-queue

다음 코드 예시에서는 create-queue 코드를 사용하는 방법을 보여줍니다.

AWS CLI

대기열 생성

다음 create-queue 예시에서는 사용자 지정 트랜스코딩 대기열을 만듭니다.

```
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 Elemental MediaConvert 사용 설명서의 Working with AWS Elemental MediaConvert Queues를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateQueue를 참조하세요.

delete-job-template

다음 코드 예시에서는 `delete-job-template` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

작업 템플릿 삭제

다음 `delete-job-template` 예시에서는 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. `aws mediaconvert list-job-templates`를 실행하여 템플릿이 삭제되었는지 확인합니다.

자세한 내용은 AWS Elemental MediaConvert 사용 설명서의 Working with AWS Elemental MediaConvert Job Templates를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteJobTemplate 섹션을 참조하세요.

delete-preset

다음 코드 예시에서는 `delete-preset` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

온디맨드 대기열 삭제

다음 `delete-preset` 예시에서는 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. `aws mediaconvert list-presets`를 실행하여 사전 설정이 삭제되었는지 확인합니다.

자세한 내용은 AWS Elemental MediaConvert 사용 설명서의 Working with AWS Elemental MediaConvert Output Presets를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeletePreset 섹션을 참조하세요.

`delete-queue`

다음 코드 예시에서는 `delete-queue` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

온디맨드 대기열 삭제

다음 `delete-queue` 예시에서는 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. `aws mediaconvert list-queues`를 실행하여 대기열이 삭제되었는지 확인합니다.

자세한 내용은 AWS Elemental MediaConvert 사용 설명서의 Working with AWS Elemental MediaConvert Queues를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteQueue를 참조하세요.

describe-endpoints

다음 코드 예시에서는 describe-endpoints 코드를 사용하는 방법을 보여줍니다.

AWS CLI

계정별 엔드포인트 가져오기

다음 describe-endpoints 예시에서는 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 AWS MediaConvert API 참조의 Getting Started with MediaConvert Using the API를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeEndpoints 섹션을 참조하세요.

get-job-template

다음 코드 예시에서는 get-job-template 코드를 사용하는 방법을 보여줍니다.

AWS CLI

작업 템플릿에 대한 세부 정보 가져오기

다음 get-job-template 예시에서는 지정된 사용자 지정 작업 템플릿의 JSON 정의를 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 Elemental MediaConvert 사용 설명서의 Working with AWS Elemental MediaConvert Job Templates를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetJobTemplate 섹션을 참조하세요.

get-job

다음 코드 예시에서는 get-job 코드를 사용하는 방법을 보여줍니다.

AWS CLI

특정 작업의 세부 정보 가져오기

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 비롯한 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 Elemental MediaConvert 사용자 안내서의 AWS Elemental MediaConvert 작업 처리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetJob을 참조하세요.

get-preset

다음 코드 예시에서는 get-preset 코드를 사용하는 방법을 보여줍니다.

AWS CLI

특정 사전 설정에 대한 세부 정보 가져오기

다음 `get-preset` 예시에서는 지정된 사용자 지정 사전 설정의 JSON 정의를 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 Elemental MediaConvert 사용 설명서의 Working with AWS Elemental MediaConvert Output Presets를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetPreset 섹션을 참조하세요.

get-queue

다음 코드 예시에서는 get-queue 코드를 사용하는 방법을 보여줍니다.

AWS CLI

대기열에 대한 세부 정보를 가져오는 방법

다음 get-queue 예시에서는 지정된 데이터 스트림의 세부 정보를 반환합니다.

```
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 Elemental MediaConvert 사용 설명서의 Working with AWS Elemental MediaConvert Queues를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetQueue 섹션을 참조하세요.

list-job-templates

다음 코드 예시에서는 `list-job-templates` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

예시 1: 사용자 지정 작업 템플릿을 나열하는 방법

다음 `list-job-templates` 예시에서는 현재 리전의 모든 사용자 지정 작업 템플릿을 나열합니다. 시스템 작업 템플릿을 나열하려면 다음 예시를 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
        }
    ]
}
```

예시 2: MediaConvert 시스템 작업 템플릿을 나열하는 방법

다음 `list-job-templates` 예시에서는 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 Elemental MediaConvert 사용 설명서의 Working with AWS Elemental MediaConvert Job Templates를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListJobTemplates 섹션을 참조하세요.

list-jobs

다음 코드 예시에서는 `list-jobs` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

리전 내 모든 작업의 세부 정보 가져오기

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 Elemental MediaConvert 사용자 안내서의 AWS Elemental MediaConvert 작업 처리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListJobs를 참조하세요.

list-presets

다음 코드 예시에서는 `list-presets` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

예시 1: 사용자 지정 출력 사전 설정을 나열하는 방법

다음 `list-presets` 예시에서는 사용자 지정 출력 사전 설정을 나열합니다. 시스템 사전 설정을 나열하려면 다음 예시를 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."
        }
    ]
}
```

예시 2: 시스템 출력 사전 설정을 나열하는 방법

다음 `list-presets` 예시에서는 사용 가능한 MediaConvert 시스템 사전 설정을 나열합니다. 사용자 지정 사전 설정을 나열하려면 이전 예시를 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 Elemental MediaConvert 사용 설명서의 Working with AWS Elemental MediaConvert Output Presets를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListPresets 섹션을 참조하세요.

list-queues

다음 코드 예시에서는 list-queues 코드를 사용하는 방법을 보여줍니다.

AWS CLI

대기열 나열

다음 list-queues 예시에서는 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 Elemental MediaConvert 사용 설명서의 Working with AWS Elemental MediaConvert Queues를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListQueues를 참조하세요.

`list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

MediaConvert 대기열, 작업 템플릿 또는 출력 사전 설정의 태그를 나열하는 방법

다음 `list-tags-for-resource` 예시에서는 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 Elemental MediaConvert 사용 설명서의 Tagging AWS Elemental MediaConvert Queues, Job Templates, and Output Presets를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource 섹션을 참조하세요.

`update-job-template`

다음 코드 예시에서는 `update-job-template` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

변경 템플릿 생성

다음 `update-job-template` 예시에서는 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.json`의 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
}
```

요청에 오류가 발생하더라도 시스템에서는 요청과 함께 전송한 JSON 페이로드를 반환합니다. 따라서 반환되는 JSON이 반드시 작업 템플릿의 새 정의일 필요는 없습니다.

JSON 페이로드가 길 수 있으므로 오류 메시지를 보려면 위로 스크롤해야 할 수 있습니다.

자세한 내용은 AWS Elemental MediaConvert 사용 설명서의 Working with AWS Elemental MediaConvert Job Templates를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateJobTemplate 섹션을 참조하세요.

update-preset

다음 코드 예시에서는 update-preset 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사전 설정을 변경하는 방법

다음 update-preset 예시는 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 Elemental MediaConvert 사용 설명서의 Working with AWS Elemental MediaConvert Output Presets를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdatePreset 섹션을 참조하세요.

update-queue

다음 코드 예시에서는 update-queue 코드를 사용하는 방법을 보여줍니다.

AWS CLI

대기열을 변경

다음 update-queue 예시에서는 상태를 PAUSED로 변경하여 지정된 대기열을 일시 중지합니다,

```
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 Elemental MediaConvert 사용 설명서의 Working with AWS Elemental MediaConvert Queues를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateQueue 섹션을 참조하세요.

AWS CLI를 사용한 MediaLive 예시

다음 코드 예시는 MediaLive와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

create-channel

다음 코드 예시에서는 `create-channel`의 사용 방법을 보여줍니다.

AWS CLI

채널 생성

다음 `create-channel` 예시에서는 지정하려는 파라미터가 포함된 JSON 파일을 전달하여 채널을 생성합니다.

이 예시의 채널은 비디오, 오디오 및 임베디드 캡션이 포함된 소스에 연결하는 HLS PULL 입력을 수집합니다. 채널은 Akamai 서버를 대상으로 하는 하나의 HLS 출력 그룹을 생성합니다. 출력 그룹에는 두 개의 출력이 포함되어 있습니다. 하나는 H.265 비디오 및 AAC 오디오용이고 다른 하나는 Web-VTT 캡션용이며 영어로만 제공됩니다.

이 예시 채널의 JSON에는 HLS PULL 입력을 사용하고 Akamai를 대상으로 하는 HLS 출력 그룹을 생성하는 채널에 필요한 최소 파라미터가 포함됩니다. JSON에는 다음과 같은 주요 섹션이 포함되어 있습니다.

`InputAttachments`: 오디오에 대한 소스 하나와 캡션에 대한 소스 하나를 지정합니다. 비디오 선택기는 지정하지 않습니다. 즉, MediaLive가 소스에서 찾은 첫 번째 비디오를 추출합니다. `Destinations`: 이 채널의 단일 출력 그룹에 대한 두 개의 IP 주소(URL)를 포함합니다. 이러한 주소에는 암호가 필요합니다. `EncoderSettings`: 하위 섹션을 포함합니다. `AudioDescriptions`: InputAttachments의 소스를 사용하고 AAC 형식의 오디오를 생성하는 하나의 오디오 출력 자산이 채널에 포함되어 있음을 명시합니다. `CaptionDescriptions`: InputAttachments의 소스를 사용하고 Web-VTT 형식의 캡션을 생성하는 하나의 캡션 출력 자산이 포함되어 있음을 명시합니다. `VideoDescriptions`: 지정된 해상도를 가진 하나의 비디오 출력 자산이 채널에 포함되어 있음을 명시합니다. `OutputGroups`: 출력 그룹을 지정합니다. 이 예시에는 `Akamai`라는 리소스 그룹이 하

나 있습니다. 연결은 HLS PUT를 사용하여 구축됩니다. 출력 그룹에는 출력 두 개가 포함되어 있습니다. 한 가지 출력은 비디오 자산(Video_high)과 오디오 자산(Audio_EN)입니다. 하나의 출력은 캡션 자산(WebVTT_EN)입니다.

이 예시에서는 일부 파라미터에 값이 없거나 중첩된 빈 파라미터가 포함되어 있습니다. 예를 들어 Video_and_audio 출력의 OutputSettings에는 빈 파라미터 M3u8Settings로 끝나는 여러 중첩 파라미터가 포함되어 있습니다. 이 파라미터를 포함해야 하지만 하위 파라미터는 하나, 여러 개 또는 모두 생략할 수 있습니다. 즉, 하위 파라미터가 기본값을 사용하거나 null이 됩니다.

이 예시 채널에 적용되지만 이 파일에 지정되지 않은 모든 파라미터는 기본값을 사용하거나 null로 설정되거나 MediaLive에서 생성한 고유한 값을 사용합니다.

```
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.

채널용 ARN입니다. ARN의 마지막 부분은 고유한 채널 ID입니다. `EgressEndpoints`는 PUSH 입력에만 사용되므로 이 예시 채널에서는 비어 있습니다. 적용하면 콘텐츠가 푸시되는 MediaLive의 주소가 표시됩니다. `OutputGroups`, `Outputs`. 포함되지 않았지만 이 채널에 관련된 파라미터를 포함하여 출력 그룹 및 출력에 대한 모든 파라미터가 표시됩니다. 파라미터가 비어 있을 수 있으며(이 채널 구성에서 파라미터 또는 특성이 비활성화되었음을 의미할 수 있음) 또는 적용될 기본값을 표시할 수 있습니다. `LogLevel`은 기본값(DISABLED)으로 설정되어 있습니다. `Tags`는 기본값(null)으로 설정되어 있습니다. `PipelinesRunningCount` 및 `State`는 채널의 현재 상태를 보여줍니다.

자세한 내용은 AWS Elemental MediaLive 사용자 안내서의 채널을 처음부터 새로 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateChannel을 참조하세요.

`create-input`

다음 코드 예시에서는 `create-input`의 사용 방법을 보여줍니다.

AWS CLI

입력 생성

다음 `create-input` 예시에서는 이 유형의 입력에 적용되는 파라미터가 포함된 JSON 파일을 전달하여 HLS PULL 입력을 생성합니다. 이 예시 입력의 JSON은 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: 입력 ARN입니다. ARN의 마지막 부분은 고유 입력 ID입니다. Attached Channels: 새로 생성된 입력에 대해 항상 비어 있습니다. Destinations: PUSH 입력에만 사용되므로 이 예시에서는 비어 있습니다. Id: 입력 ID입니다. ARN의 ID와 동일합니다. MediaConnectFlows: MediaConnect 유형의 입력에만 사용되므로 이 예시에서는 비어 있습니다. SecurityGroups: PUSH 입력에만 사용되므로 비어 있습니다. State: 이 입력의 상태입니다. Tags: 비어 있습니다(이 파라미터의 기본값).

자세한 내용은 AWS Elemental MediaLive 사용자 안내서의 [입력 생성](#)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [CreateInput](#)을 참조하세요.

AWS CLI를 사용한 MediaPackage 예시

다음 코드 예시는 MediaPackage와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`create-channel`

다음 코드 예시에서는 `create-channel`의 사용 방법을 보여줍니다.

AWS CLI

채널 생성

다음 `create-channel` 명령은 현재 계정에서 `sportschannel`이라는 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 AWS Elemental MediaPackage 사용자 안내서의 채널 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateChannel을 참조하세요.

create-origin-endpoint

다음 코드 예시에서는 create-origin-endpoint의 사용 방법을 보여줍니다.

AWS CLI

오리진 엔드포인트 생성

다음 create-origin-endpoint 명령은 JSON 파일에 제공된 패키지 설정과 지정된 엔드포인트 설정으로 cmafsports라는 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 Elemental MediaPackage 사용자 안내서의 엔드포인트 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateOriginEndpoint를 참조하세요.

delete-channel

다음 코드 예시는 delete-channel의 사용 방법을 보여 줍니다.

AWS CLI

채널 삭제

다음 delete-channel 명령은 test라는 채널을 삭제합니다.

```
aws mediapackage delete-channel \
    --id test
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Elemental MediaPackage 사용자 안내서의 채널 삭제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteChannel을 참조하세요.

delete-origin-endpoint

다음 코드 예시에서는 delete-origin-endpoint의 사용 방법을 보여줍니다.

AWS CLI

오리진 엔드포인트 삭제

다음 delete-origin-endpoint 명령은 tester2라는 오리진 엔드포인트를 삭제합니다.

```
aws mediapackage delete-origin-endpoint \
    --id tester2
```

자세한 내용은 AWS Elemental MediaPackage 사용자 안내서의 엔드포인트 삭제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteOriginEndpoint를 참조하세요.

describe-channel

다음 코드 예시에서는 describe-channel의 사용 방법을 보여줍니다.

AWS CLI

채널 설명

다음 `describe-channel` 명령은 `test`라는 채널의 모든 세부 정보를 표시합니다.

```
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 사용자 안내서의 채널 세부 정보 보기<https://docs.aws.amazon.com/mediapackage/latest/ug/channels-view.html>를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeChannel을 참조하세요.

`describe-origin-endpoint`

다음 코드 예시에서는 `describe-origin-endpoint`의 사용 방법을 보여줍니다.

AWS CLI

오리진 엔드포인트 설명

다음 describe-origin-endpoint 명령은 cmaf_sports라는 오리진 엔드포인트의 모든 세부 정보를 표시합니다.

```
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 AWS Elemental MediaPackage 사용자 안내서의 [단일 엔드포인트 보기]를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeOriginEndpoint]를 참조하세요.

list-channels

다음 코드 예시에서는 `list-channels`의 사용 방법을 보여줍니다.

AWS CLI

모든 채널 나열

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 AWS Elemental MediaPackage 사용자 안내서의 [채널 세부 정보 보기](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ListChannels](#)를 참조하세요.

list-origin-endpoints

다음 코드 예시에서는 `list-origin-endpoints`의 사용 방법을 보여줍니다.

AWS CLI

채널의 모든 오리진 엔드포인트 나열

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 Elemental MediaPackage 사용자 안내서의 채널에 연결된 모든 엔드포인트 보기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListOriginEndpoints를 참조하세요.

`list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`의 사용 방법을 보여줍니다.

AWS CLI

리소스에 할당된 태그 나열

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 AWS Elemental MediaPackage 사용자 안내서의 AWS Elemental MediaPackage의 리소스에 태그 지정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource를 참조하세요.

`rotate-ingest-endpoint-credentials`

다음 코드 예시에서는 `rotate-ingest-endpoint-credentials`의 사용 방법을 보여줍니다.

AWS CLI

수집 자격 증명 교체

다음 `rotate-ingest-endpoint-credentials` 명령은 지정된 수집 엔드포인트의 WebDAV 사용자 이름과 암호를 교체합니다.

```
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 AWS Elemental MediaPackage 사용자 안내서의 입력 URL에서 자격 증명 교체를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RotateIngestEndpointCredentials를 참조하세요.

tag-resource

다음 코드 예시에서는 tag-resource의 사용 방법을 보여줍니다.

AWS CLI

리소스에 태그 추가

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 AWS Elemental MediaPackage 사용자 안내서의 AWS Elemental MediaPackage의 리소스에 태그 지정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

untag-resource

다음 코드 예시에서는 untag-resource의 사용 방법을 보여줍니다.

AWS CLI

리소스에서 태그 제거

다음 untag-resource 명령은 지정된 채널에서 키 region이 있는 태그를 제거합니다.

```
aws mediapackage untag-resource \
    --resource-arn arn:aws:mediapackage:us-
west-2:111222333:channels/6d345804ec3f46c9b454a91d4a80d0e0 \
    --tag-keys region
```

자세한 내용은 AWS Elemental MediaPackage 사용자 안내서의 AWS Elemental MediaPackage의 리소스에 태그 지정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

update-channel

다음 코드 예시에서는 update-channel의 사용 방법을 보여줍니다.

AWS CLI

채널 업데이트

다음 update-channel 명령은 sportschannel이라는 채널을 업데이트하여 설명 24x7 sports를 포함합니다.

```
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 Elemental MediaPackage 사용자 안내서의 채널 편집을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateChannel을 참조하세요.

update-origin-endpoint

다음 코드 예시에서는 update-origin-endpoint의 사용 방법을 보여줍니다.

AWS CLI

오리진 엔드포인트 업데이트

다음 update-origin-endpoint 명령은 cmaf_sports라는 오리진 엔드포인트를 업데이트합니다. 시간 지연을 0초로 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 Elemental MediaPackage 사용자 안내서의 엔드포인트 편집을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateOriginEndpoint를 참조하세요.

AWS CLI를 사용한 MediaPackage VOD 예시

다음 코드 예시는 MediaPackage VOD와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`create-asset`

다음 코드 예시에서는 `create-asset`의 사용 방법을 보여줍니다.

AWS CLI

애셋 생성

다음 `create-asset` 예시에서는 현재 AWS 계정에서 `Chicken_Asset`이라는 애셋을 생성합니다. 애셋은 MediaPackage로 파일 `30sec_chicken.smil`을 수집합니다.

```
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 AWS Elemental MediaPackage 사용자 안내서의 애셋 수집을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateAsset을 참조하세요.

`create-packaging-configuration`

다음 코드 예시에서는 `create-packaging-configuration`의 사용 방법을 보여줍니다.

AWS CLI

패키징 구성 생성

다음 `create-packaging-configuration` 예시에서는 `hls_chicken`이라는 패키징 그룹에 `new_hls`라는 패키징 구성을 생성합니다. 이 예시에서는 `hls_pc.json`이라는 디스크의 파일을 사용하여 세부 정보를 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 AWS Elemental MediaPackage 사용자 안내서의 패키징 구성 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreatePackagingConfiguration을 참조하세요.

create-packaging-group

다음 코드 예시에서는 create-packaging-group의 사용 방법을 보여줍니다.

AWS CLI

패키징 그룹 생성

다음 create-packaging-group 예시에서는 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 AWS Elemental MediaPackage 사용자 안내서의 패키징 그룹 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreatePackagingGroup을 참조하세요.

delete-asset

다음 코드 예시에서는 delete-asset의 사용 방법을 보여줍니다.

AWS CLI

애셋 삭제

다음 delete-asset 예시에서는 30sec_chicken이라는 애셋을 삭제합니다.

```
aws mediapackage-vod delete-asset \
    --id 30sec_chicken
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Elemental MediaPackage 사용자 안내서의 애셋 삭제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteAsset을 참조하세요.

delete-packaging-configuration

다음 코드 예시에서는 delete-packaging-configuration의 사용 방법을 보여줍니다.

AWS CLI

패키징 구성 삭제

다음 delete-packaging-configuration 예시에서는 CMAF이라는 패키징 구성을 삭제합니다.

```
aws mediapackage-vod delete-packaging-configuration \
    --id CMAF
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Elemental MediaPackage 사용자 안내서의 패키징 구성 삭제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeletePackagingConfiguration을 참조하세요.

delete-packaging-group

다음 코드 예시에서는 delete-packaging-group의 사용 방법을 보여줍니다.

AWS CLI

패키징 그룹 삭제

다음 delete-packaging-group 예시에서는 Dash_widevine이라는 패키징 그룹을 삭제합니다.

```
aws mediapackage-vod delete-packaging-group \
    --id Dash_widevine
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Elemental MediaPackage 사용자 안내서의 [패키징 그룹 삭제](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DeletePackagingGroup](#)을 참조하세요.

describe-asset

다음 코드 예시에서는 describe-asset의 사용 방법을 보여줍니다.

AWS CLI

애셋 설명

다음 describe-asset 예시에서는 30sec_chicken이라는 애셋의 모든 세부 정보를 표시합니다.

```
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 AWS Elemental MediaPackage 사용자 안내서의 애셋 세부 정보 보기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeAsset을 참조하세요.

`describe-packaging-configuration`

다음 코드 예시에서는 `describe-packaging-configuration`의 사용 방법을 보여줍니다.

AWS CLI

패키징 구성 설명

다음 `describe-packaging-configuration` 예시에서는 DASH라는 패키징 구성의 모든 세부 정보를 표시합니다.

```
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 AWS Elemental MediaPackage 사용자 안내서의 패키징 구성 세부 정보 보기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribePackagingConfiguration을 참조하세요.

describe-packaging-group

다음 코드 예시에서는 describe-packaging-group의 사용 방법을 보여줍니다.

AWS CLI

패키징 그룹 설명

다음 describe-packaging-group 예시에서는 Packaging_group_1이라는 패키징 그룹의 모든 세부 정보를 표시합니다.

```
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 AWS Elemental MediaPackage 사용자 안내서의 패키징 그룹 세부 정보 보기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribePackagingGroup을 참조하세요.

list-assets

다음 코드 예시에서는 `list-assets`의 사용 방법을 보여줍니다.

AWS CLI

모든 애셋 나열

다음 `list-assets` 예시에서는 현재 AWS 계정에 구성된 모든 애셋을 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 Elemental MediaPackage 사용자 안내서의 애셋 세부 정보 보기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListAssets를 참조하세요.

`list-packaging-configurations`

다음 코드 예시에서는 `list-packaging-configurations`의 사용 방법을 보여줍니다.

AWS CLI

모든 패키징 구성 나열

다음 `list-packaging-configurations` 예시에서는 `Packaging_group_1`이라는 패키징 그룹에 구성된 모든 패키징 구성을 나열합니다.

```
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 AWS Elemental MediaPackage 사용자 안내서의 패키징 구성 세부 정보 보기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListPackagingConfigurations를 참조하세요.

list-packaging-groups

다음 코드 예시에서는 `list-packaging-groups`의 사용 방법을 보여줍니다.

AWS CLI

모든 패키징 그룹 나열

다음 `list-packaging-groups` 예시에서는 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 AWS Elemental MediaPackage 사용자 안내서의 패키징 그룹 세부 정보 보기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListPackagingGroups를 참조하세요.

AWS CLI를 사용한 MediaStore 데이터 플레인 예시

다음 코드 예시에서는 MediaStore 데이터 플레인과 함께 AWS Command Line Interface를 사용하여 일반적인 시나리오를 구현하고 개별 서비스 작업을 수행하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- [작업]

작업

delete-object

다음 코드 예시에서는 delete-object의 사용 방법을 보여줍니다.

AWS CLI

객체 삭제

다음 delete-object 예시에서는 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 Elemental MediaStore 사용자 안내서의 [객체 삭제]를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteObject]를 참조하세요.

describe-object

다음 코드 예시에서는 describe-object의 사용 방법을 보여줍니다.

AWS CLI

객체의 헤더 보기

다음 describe-object 예시에서는 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 Elemental MediaStore 사용자 안내서의 [객체 세부 정보 보기]를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeObject]를 참조하세요.

get-object

다음 코드 예시에서는 `get-object`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 전체 객체 다운로드

다음 `get-object` 예시에서는 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"
}
```

예시 2: 객체의 일부 다운로드

다음 `get-object` 예시에서는 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 Elemental MediaStore 사용자 안내서의 객체 다운로드를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetObject를 참조하세요.

list-items

다음 코드 예시에서는 `list-items`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 컨테이너에 저장된 항목(객체 및 폴더) 목록 보기

다음 `list-items` 예시에서는 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"
            }
        ]
}
```

예시 2: 폴더에 저장된 항목(객체 및 폴더) 목록 보기

다음 `list-items` 예시에서는 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 Elemental MediaStore 사용자 안내서의 객체 목록 보기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListItems를 참조하세요.

`put-object`

다음 코드 예시에서는 `put-object`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 컨테이너에 객체 업로드

다음 `put-object` 예시에서는 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"
}
```

예시 2: 컨테이너 내 폴더에 객체 업로드

다음 `put-object` 예시에서는 컨테이너 내 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 Elemental MediaStore 사용자 안내서의 객체 업로드를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutObject를 참조하세요.

AWS CLI를 사용한 MediaTailor 예시

다음 코드 예시에서는 MediaTailor와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

delete-playback-configuration

다음 코드 예시에서는 delete-playback-configuration의 사용 방법을 보여줍니다.

AWS CLI

구성 삭제

다음 delete-playback-configuration 예시에서는 campaign_short라는 구성을 삭제합니다.

```
aws mediatailor delete-playback-configuration \
    --name campaign_short
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Elemental MediaTailor 사용자 안내서의 구성 삭제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeletePlaybackConfiguration을 참조하세요.

get-playback-configuration

다음 코드 예시에서는 get-playback-configuration의 사용 방법을 보여줍니다.

AWS CLI

구성 설명

다음 get-playback-configuration은 west_campaign이라는 구성의 모든 세부 정보를 표시합니다.

```
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 Elemental MediaTailor 사용자 안내서의 구성 보기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetPlaybackConfiguration을 참조하세요.

list-playback-configurations

다음 코드 예시에서는 list-playback-configurations의 사용 방법을 보여줍니다.

AWS CLI

모든 구성 나열

다음 list-playback-configurations는 현재 AWS 계정의 구성에 대한 모든 세부 정보를 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 AWS Elemental MediaTailor 사용자 안내서의 구성 보기<https://docs.aws.amazon.com/mediatailor/latest/ug/configurations-view.html>를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListPlaybackConfigurations를 참조하세요.

put-playback-configuration

다음 코드 예시에서는 put-playback-configuration의 사용 방법을 보여줍니다.

AWS CLI

구성 생성

다음 put-playback-configuration은 campaign_short라는 구성을 생성합니다.

```
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 AWS Elemental MediaTailor 사용자 안내서의 구성 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutPlaybackConfiguration을 참조하세요.

AWS CLI를 사용한 MemoryDB 예시

다음 코드 예시에서는 MemoryDB와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

copy-snapshot

다음 코드 예시에서는 copy-snapshot의 사용 방법을 보여줍니다.

AWS CLI

스냅샷 복사

다음 copy-snapshot 예시에서는 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 사용자 안내서의 스냅샷 복사를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CopySnapshot을 참조하세요.

create-acl

다음 코드 예시에서는 `create-acl`의 사용 방법을 보여줍니다.

AWS CLI

ACL 생성

다음 `create-acl` 예시에서는 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 사용자 안내서의 액세스 제어 목록을 사용하여 사용자 인증을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateAcl을 참조하세요.

create-cluster

다음 코드 예시에서는 `create-cluster`의 사용 방법을 보여줍니다.

AWS CLI

클러스터 생성

다음 `create-cluster` 예시에서는 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 사용자 안내서의 클러스터 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateCluster를 참조하세요.

`create-parameter-group`

다음 코드 예제에서는 `create-parameter-group`의 사용 방법을 보여줍니다.

AWS CLI

파라미터 그룹 생성

다음 `create-parameter-group` 예시에서는 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 사용자 안내서의 파라미터 그룹 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateParameterGroup을 참조하세요.

`create-snapshot`

다음 코드 예시에서는 `create-snapshot`의 사용 방법을 보여줍니다.

AWS CLI

스냅샷 생성

다음 `create-snapshot` 예시에서는 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 사용자 안내서의 수동 스냅샷 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateSnapshot을 참조하세요.

`create-subnet-group`

다음 코드 예시에서는 `create-subnet-group`의 사용 방법을 보여줍니다.

AWS CLI

서브넷 그룹 생성

다음 `create-subnet-group` 예시에서는 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 사용자 안내서의 [서브넷 그룹 생성]을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [CreateSubnetGroup]을 참조하세요.

`create-user`

다음 코드 예시에서는 `create-user`의 사용 방법을 보여줍니다.

AWS CLI

사용자 생성

다음 `create-user` 예시에서는 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 사용자 안내서의 액세스 제어 목록을 사용하여 사용자 인증을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateUser를 참조하세요.

delete-acl

다음 코드 예시에서는 `delete-acl`의 사용 방법을 보여줍니다.

AWS CLI

ACL 삭제

다음 `delete-acl` 예시에서는 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 사용자 안내서의 [액세스 제어 목록을 사용하여 사용자 인증](#)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteAcl](#)을 참조하세요.

`delete-cluster`

다음 코드 예시에서는 `delete-cluster`의 사용 방법을 보여줍니다.

AWS CLI

클러스터 삭제

다음 `delete-cluster` 예시에서는 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 사용자 안내서의 클러스터 삭제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteCluster를 참조하세요.

delete-parameter-group

다음 코드 예제에서는 delete-parameter-group의 사용 방법을 보여줍니다.

AWS CLI

파라미터 그룹 삭제

다음 delete-parameter-group 예시에서는 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 사용자 안내서의 파라미터 그룹 삭제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteParameterGroup을 참조하세요.

delete-snapshot

다음 코드 예시에서는 delete-snapshot의 사용 방법을 보여줍니다.

AWS CLI

스냅샷 삭제

다음 delete-snapshot 예시에서는 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 사용자 안내서의 스냅샷 삭제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteSnapshot을 참조하세요.

`delete-subnet-group`

다음 코드 예시에서는 `delete-subnet-group`의 사용 방법을 보여줍니다.

AWS CLI

서브넷 그룹 삭제

다음 `delete-subnet-group` 예시에서는 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 사용자 안내서의 서브넷 그룹 삭제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteSubnetGroup을 참조하세요.

delete-user

다음 코드 예시에서는 delete-user의 사용 방법을 보여줍니다.

AWS CLI

사용자 삭제

다음 delete-user 예시에서는 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 사용자 안내서의 액세스 제어 목록을 사용하여 사용자 인증을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteUser를 참조하세요.

describe-acls

다음 코드 예시에서는 describe-acls의 사용 방법을 보여줍니다.

AWS CLI

ACL 목록 반환

다음 describe-acls는 ACL 목록을 반환합니다.

```
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 사용자 안내서의 [액세스 제어 목록을 사용하여 사용자 인증]을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeAcls]를 참조하세요.

`describe-clusters`

다음 코드 예시에서는 `describe-clusters`의 사용 방법을 보여줍니다.

AWS CLI

클러스터 목록 반환

다음 describe-clusters는 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 사용자 안내서의 클러스터 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeClusters를 참조하세요.

describe-engine-versions

다음 코드 예시에서는 describe-engine-versions의 사용 방법을 보여줍니다.

AWS CLI

엔진 버전 목록 반환

다음 describe-engine-versions는 엔진 버전 목록을 반환합니다.

```
aws memorydb describe-engine-versions
```

출력:

```
{
    "EngineVersions": [
        {
            "EngineVersion": "6.2",
            "EnginePatchVersion": "6.2.6",
            "ParameterGroupFamily": "memorydb_redis6"
        }
    ]
}
```

자세한 내용은 MemoryDB 사용자 안내서의 엔진 버전 및 업그레이드를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeEngineVersions를 참조하세요.

describe-events

다음 코드 예시에서는 describe-events의 사용 방법을 보여줍니다.

AWS CLI

이벤트 목록 반환

다음 describe-events는 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 사용자 안내서의 이벤트 모니터링을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeEvents를 참조하세요.

describe-parameter-groups

다음 코드 예시에서는 describe-parameter-groups의 사용 방법을 보여줍니다.

AWS CLI

파라미터 그룹 목록 반환

다음 describe-parameter-groups는 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 사용자 안내서의 파라미터 그룹을 사용하여 엔진 파라미터 구성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeParameterGroups를 참조하세요.

describe-parameters

다음 코드 예시에서는 describe-parameters의 사용 방법을 보여줍니다.

AWS CLI

파라미터 목록 반환

다음 describe-parameters는 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 사용자 안내서의 파라미터 그룹을 사용하여 엔진 파라미터 구성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeParameters를 참조하세요.

describe-snapshots

다음 코드 예시에서는 describe-snapshots의 사용 방법을 보여줍니다.

AWS CLI

스냅샷 목록 반환

다음 describe-snapshots는 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 사용자 안내서의 스냅샷 및 복원을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeSnapshots를 참조하세요.

describe-subnet-groups

다음 코드 예시에서는 describe-subnet-groups의 사용 방법을 보여줍니다.

AWS CLI

서브넷 그룹 목록 반환

다음 describe-subnet-groups는 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 사용자 안내서의 서브넷 및 서브넷 그룹을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeSubnetGroups를 참조하세요.

describe-users

다음 코드 예시에서는 describe-users의 사용 방법을 보여줍니다.

AWS CLI

사용자 목록 반환

다음 describe-users는 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 사용자 안내서의 액세스 제어 목록을 사용하여 사용자 인증을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeUsers를 참조하세요.

failover-shard

다음 코드 예시에서는 `failover-shard`의 사용 방법을 보여줍니다.

AWS CLI

샤드 장애 조치

다음 failover-shard는 샤드를 장애 조치합니다.

```
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 사용자 안내서의 MultiAZ로 가동 중지 시간 최소화를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 FailoverShard를 참조하세요.

list-allowed-node-type-updates

다음 코드 예시에서는 `list-allowed-node-type-updates`의 사용 방법을 보여줍니다.

AWS CLI

허용된 노드 유형 업데이트의 목록 반환

다음 list-allowed-node-type-updates는 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 사용자 안내서의 스케일링을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListAllowedNodeTypeUpdates를 참조하세요.

`list-tags`

다음 코드 예시에서는 `list-tags`의 사용 방법을 보여줍니다.

AWS CLI

태그 목록 반환

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 사용자 안내서의 리소스 태그 지정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTags를 참조하세요.

reset-parameter-group

다음 코드 예시에서는 `reset-parameter-group`의 사용 방법을 보여줍니다.

AWS CLI

파라미터 그룹 재설정

다음 reset-parameter-group은 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 사용자 안내서의 [파라미터 그룹을 사용하여 엔진 파라미터 구성]을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ResetParameterGroup]을 참조하세요.

`tag-resource`

다음 코드 예시에서는 `tag-resource`의 사용 방법을 보여줍니다.

AWS CLI

리소스에 태그 지정

다음 tag-resource는 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 사용자 안내서의 리소스 태그 지정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

untag-resource

다음 코드 예시에서는 untag-resource의 사용 방법을 보여줍니다.

AWS CLI

ACL 업데이트

다음 update-acl은 사용자를 추가하여 ACL을 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 사용자 안내서의 리소스 태그 지정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

update-cluster

다음 코드 예시에서는 update-cluster의 사용 방법을 보여줍니다.

AWS CLI

클러스터 업데이트

다음 update-cluster는 클러스터의 파라미터 그룹을 my-parameter-group으로 업데이트합니다.

```
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 사용자 안내서의 클러스터 수정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateCluster를 참조하세요.

update-parameter-group

다음 코드 예시에서는 update-parameter-group의 사용 방법을 보여줍니다.

AWS CLI

파라미터 그룹 업데이트

다음 update-parameter-group은 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 사용자 안내서의 파라미터 그룹 수정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateParameterGroup을 참조하세요.

update-subnet-group

다음 코드 예시에서는 update-subnet-group의 사용 방법을 보여줍니다.

AWS CLI

서브넷 그룹 업데이트

다음 update-subnet-group은 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 사용자 안내서의 서브넷 및 서브넷 그룹을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateSubnetGroup을 참조하세요.

update-user

다음 코드 예시에서는 update-user의 사용 방법을 보여줍니다.

AWS CLI

사용자 업데이트

다음 update-user는 사용자의 액세스 문자열을 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 사용자 안내서의 액세스 제어 목록을 사용하여 사용자 인증을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateUser를 참조하세요.

AWS CLI를 사용한 Amazon MSK 예시

다음 코드 예시는 Amazon MSK와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

create-cluster

다음 코드 예시에서는 `create-cluster`의 사용 방법을 보여줍니다.

AWS CLI

Amazon MSK 클러스터 생성

다음 create-cluster 예시에서는 3개의 브로커 노드로 MessagingCluster라는 MSK 클러스터를 생성합니다. brokernodegroupinfo.json이라는 JSON 파일은 Amazon MSK가 브로커 노드를 배포할 세 개의 서브넷을 지정합니다. 이 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateCluster를 참조하세요.

create-configuration

다음 코드 예시에서는 create-configuration의 사용 방법을 보여줍니다.

AWS CLI

사용자 지정 Amazon MSK 구성 생성

다음 `create-configuration` 예시에서는 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.txt`의 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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateConfiguration을 참조하세요.

describe-cluster

다음 코드 예시에서는 describe-cluster의 사용 방법을 보여줍니다.

AWS CLI

클러스터 설명

다음 describe-cluster 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeCluster를 참조하세요.

get-bootstrap-brokers

다음 코드 예시에서는 `get-bootstrap-brokers`의 사용 방법을 보여줍니다.

AWS CLI

부트스트랩 브로커 가져오기

다음 `get-bootstrap-brokers` 예시에서는 Amazon MSK 클러스터의 부트스트랩 브로커 정보를 가져옵니다.

```
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 가져오기](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [GetBootstrapBrokers](#)를 참조하세요.

`list-clusters`

다음 코드 예시에서는 `list-clusters`의 사용 방법을 보여줍니다.

AWS CLI

사용 가능한 클러스터 나열

다음 `list-clusters` 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListClusters를 참조하세요.

update-broker-storage

다음 코드 예시에서는 `update-broker-storage`의 사용 방법을 보여줍니다.

AWS CLI

브로커의 EBS 스토리지 업데이트

다음 `update-broker-storage` 예시에서는 클러스터 내 모든 브로커의 EBS 스토리지 양을 업데이트합니다. Amazon MSK는 각 브로커의 목표 스토리지 양을 예시에 지정된 양으로 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 대한 ARN을 반환합니다. 이 작업이 완료되었는지 확인하려면 이 ARN과 함께 `describe-cluster-operation` 명령을 입력으로 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 브로커의 EBS 스토리지 업데이트를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateBrokerStorage를 참조하세요.

update-cluster-configuration

다음 코드 예시에서는 update-cluster-configuration의 사용 방법을 보여줍니다.

AWS CLI

Amazon MSK 클러스터 구성 업데이트

다음 update-cluster-configuration 예시에서는 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 대한 ARN을 반환합니다. 이 작업이 완료되었는지 확인하려면 이 ARN과 함께 describe-cluster-operation 명령을 입력으로 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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateClusterConfiguration을 참조하세요.

AWS CLI를 사용한 Network Flow Monitor 예제

다음 코드 예제에서는 Network Flow Monitor와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

create-monitor

다음 코드 예시는 `create-monitor`의 사용 방법을 보여 줍니다.

AWS CLI

모니터를 생성하려면

다음 `create-monitor` 예제에서는 지정된 계정에 demo라는 모니터를 생성합니다.

```
aws networkflowmonitor create-monitor \
    --monitor-name demo \
    --local-resources type="AWS::EC2::VPC",identifier="arn:aws:ec2:us-east-1:123456789012:vpc/vpc-03ea55eeda25adbb0"  \
    --scope-arn arn:aws:networkflowmonitor:us-east-1:123456789012:scope/e21cda79-30a0-4c12-9299-d8629d76d8cf
```

출력:

```
{
    "monitorArn": "arn:aws:networkflowmonitor:us-east-1:123456789012:monitor/demo",
    "monitorName": "demo",
    "monitorStatus": "ACTIVE",
    "tags": {}
}
```

자세한 내용은 Amazon CloudWatch 사용 설명서에서 Network Flow Monitor에서 모니터 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateMonitor를 참조하세요.

create-scope

다음 코드 예시는 `create-scope`의 사용 방법을 보여 줍니다.

AWS CLI

범위를 생성하려면

다음 `create-scope` 예제에서는 Network Flow Monitor가 네트워크 트래픽 지표를 생성할 리소스 세트를 포함하는 범위를 생성합니다.

```
aws networkflowmonitor create-scope \
    --targets '[{"targetIdentifier":{"targetId":
{"accountId":"123456789012"},"targetType":"ACCOUNT"},"region":"us-east-1"}]'
```

출력:

```
{
    "scopeId": "97626f8d-8a21-4b5d-813a-1a0962dd4615",
    "status": "IN_PROGRESS",
    "tags": {}
}
```

자세한 내용은 Amazon CloudWatch 사용 설명서에서 Network Flow Monitor의 구성 요소 및 기능을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateScope 섹션을 참조하세요.

delete-monitor

다음 코드 예시는 delete-monitor의 사용 방법을 보여 줍니다.

AWS CLI

모니터 삭제

다음 delete-monitor 예제에서는 지정된 계정에서 demo라는 모니터를 삭제합니다.

```
aws networkflowmonitor delete-monitor \
    --monitor-name demo
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon CloudWatch 사용 설명서의 Network Flow Monitor에서 모니터 삭제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteMonitor를 참조하세요.

delete-scope

다음 코드 예시는 delete-scope의 사용 방법을 보여 줍니다.

AWS CLI

범위를 삭제하려면

다음 delete-scope 예제에서는 지정된 범위를 삭제합니다.

```
aws networkflowmonitor delete-scope \
    --scope-id fdc20616-6bb4-4242-a24e-a748e65ca7ac
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon CloudWatch 사용 설명서에서 Network Flow Monitor의 구성 요소 및 기능을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteScope를 참조하세요.

get-monitor

다음 코드 예시는 get-monitor의 사용 방법을 보여 줍니다.

AWS CLI

모니터 정보를 검색하려면

다음 get-monitor 예제는 지정된 계정에서 demo라는 모니터에 대한 정보를 표시합니다.

```
aws networkflowmonitor get-monitor \
    --monitor-name Demo
```

출력:

```
{
    "monitorArn": "arn:aws:networkflowmonitor:us-east-1:123456789012:monitor/Demo",
    "monitorName": "Demo",
    "monitorStatus": "ACTIVE",
    "localResources": [
        {
            "type": "AWS::EC2::VPC",
            "identifier": "arn:aws:ec2:us-east-1:123456789012:vpc/
vpc-03ea55eeda25adbb0"
        }
    ],
    "remoteResources": [],
    "createdAt": "2024-12-09T12:21:51.616000-06:00",
    "modifiedAt": "2024-12-09T12:21:55.412000-06:00",
    "tags": {}
}
```

자세한 내용은 Amazon CloudWatch 사용 설명서에서 Network Flow Monitor의 구성 요소 및 기능을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetMonitor를 참조하세요.

get-query-results-workload-insights-top-contributors-data

다음 코드 예시는 get-query-results-workload-insights-top-contributors-data의 사용 방법을 보여 줍니다.

AWS CLI

워크로드 인사이트에 대한 상위 기여자 데이터를 검색하려면

다음 get-query-results-workload-insights-top-contributors-data 예제는 지정된 쿼리에 대한 데이터를 반환합니다.

```
aws networkflowmonitor get-query-results-workload-insights-top-contributors-data \
    --scope-id e21cda79-30a0-4c12-9299-d8629d76d8cf \
    --query-id cc4f4ab3-3103-33b8-80ff-d6597a0c6cea
```

출력:

```
{
    "datapoints": [
        {
            "timestamps": [
                "2024-12-09T19:00:00+00:00",
                "2024-12-09T19:05:00+00:00",
                "2024-12-09T19:10:00+00:00"
            ],
            "values": [
                259943.0,
                194856.0,
                216432.0
            ],
            "label": "use1-az6"
        }
    ],
    "unit": "Bytes"
}
```

자세한 내용은 Amazon CloudWatch 사용 설명서의 워크로드 인사이트로 네트워크 흐름 평가를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetQueryResultsWorkloadInsightsTopContributorsData 섹션을 참조하세요.

get-query-results-workload-insights-top-contributors

다음 코드 예시는 get-query-results-workload-insights-top-contributors의 사용 방법을 보여 줍니다.

AWS CLI

워크로드 인사이트에서 상위 기여자를 검색하려면

다음 `get-query-results-workload-insights-top-contributors` 예제는 지정된 쿼리에 대한 데이터를 반환합니다.

```
aws networkflowmonitor get-query-results-workload-insights-top-contributors \
    --scope-id e21cda79-30a0-4c12-9299-d8629d76d8cf \
    --query-id 1fc423d3-b144-37a6-80e6-e2c7d26eea0c
```

출력:


```
{
    "topContributors": [
        {
            "accountId": "123456789012",
            "localSubnetId": "subnet-0a5b30fb95dca2c14",
            "localAz": "use1-az6",
            "localVpcId": "vpc-03ea55eeda25adbb0",
            "localRegion": "us-east-1",
            "remoteIdentifier": "",
            "value": 908443,
            "localSubnetArn": "arn:aws:ec2:us-east-1:123456789012:subnet/
subnet-0a5b30fb95dca2c14",
            "localVpcArn": "arn:aws:ec2:us-east-1:123456789012:vpc/
vpc-03ea55eeda25adbb0"
        }
    ]
}
```


자세한 내용은 Amazon CloudWatch 사용 설명서의 [워크로드 인사이트로 네트워크 흐름 평가]를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [GetQueryResultsWorkloadInsightsTopContributors] 섹션을 참조하세요.

`get-query-status-monitor-top-contributors`

다음 코드 예시는 `get-query-status-monitor-top-contributors`의 사용 방법을 보여 줍니다.

AWS CLI

쿼리 상태를 검색하려면

다음 get-query-status-monitor-top-contributors 예제에서는 지정된 계정에서 쿼리의 현재 상태를 표시합니다.

```
aws networkflowmonitor get-query-status-monitor-top-contributors \
    --monitor-name Demo \
    --query-id 5398eabd-bc40-3f5f-aba3-bcb639d3c7ca
```

출력:

```
{
    "status": "SUCCEEDED"
}
```

자세한 내용은 Amazon CloudWatch 사용 설명서의 [워크로드 인사이트로 네트워크 흐름 평가](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [GetQueryStatusMonitorTopContributors](#) 섹션을 참조하세요.

get-query-status-workload-insights-top-contributors-data

다음 코드 예시는 get-query-status-workload-insights-top-contributors-data의 사용 방법을 보여 줍니다.

AWS CLI

쿼리 상태를 검색하려면

다음 get-query-status-workload-insights-top-contributors-data 예제에서는 지정된 계정에서 쿼리의 현재 상태를 표시합니다.

```
aws networkflowmonitor get-query-status-workload-insights-top-contributors-data \
    --scope-id e21cda79-30a0-4c12-9299-d8629d76d8cf \
    --query-id 4333754d-8ae1-3f29-b6b7-c36db2e7f8ac
```

출력:

```
{
    "status": "SUCCEEDED"
}
```

자세한 내용은 Amazon CloudWatch 사용 설명서의 [워크로드 인사이트로 네트워크 흐름 평가]를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [GetQueryStatusWorkloadInsightsTopContributorsData] 섹션을 참조하세요.

get-query-status-workload-insights-top-contributors

다음 코드 예시는 `get-query-status-workload-insights-top-contributors`의 사용 방법을 보여 줍니다.

AWS CLI

쿼리 상태를 검색하려면

다음 `get-query-status-workload-insights-top-contributors` 예제에서는 지정된 계정에서 쿼리의 현재 상태를 표시합니다.

```
aws networkflowmonitor get-query-status-workload-insights-top-contributors \
    --scope-id e21cda79-30a0-4c12-9299-d8629d76d8cf \
    --query-id f2a87c70-3e5a-362e-8beb-4747d13d8419
```

출력:

```
{
    "status": "SUCCEEDED"
}
```

자세한 내용은 Amazon CloudWatch 사용 설명서의 [워크로드 인사이트로 네트워크 흐름 평가]를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [GetQueryStatusWorkloadInsightsTopContributors] 섹션을 참조하세요.

get-scope

다음 코드 예시는 `get-scope`의 사용 방법을 보여 줍니다.

AWS CLI

범위 정보를 검색하려면

다음 get-scope 예제에서는 상태, 태그, 이름 및 대상 세부 정보와 같은 범위에 대한 정보를 표시합니다.

```
aws networkflowmonitor get-scope \
    --scope-id e21cda79-30a0-4c12-9299-d8629d76d8cf
```

출력:

```
{
    "scopeId": "e21cda79-30a0-4c12-9299-d8629d76d8cf",
    "status": "SUCCEEDED",
    "scopeArn": "arn:aws:networkflowmonitor:us-east-1:123456789012:scope/
e21cda79-30a0-4c12-9299-d8629d76d8cf",
    "targets": [
        {
            "targetIdentifier": {
                "targetId": {
                    "accountId": "123456789012"
                },
                "targetType": "ACCOUNT"
            },
            "region": "us-east-1"
        }
    ],
    "tags": {}
}
```

자세한 내용은 Amazon CloudWatch 사용 설명서에서 Network Flow Monitor의 구성 요소 및 기능을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetScope 섹션을 참조하세요.

list-monitors

다음 코드 예시는 list-monitors의 사용 방법을 보여 줍니다.

AWS CLI

모니터 목록을 검색하려면

다음 list-monitors 예제에서는 지정된 계정의 모든 모니터를 반환합니다.

```
aws networkflowmonitor list-monitors
```

출력:

```
{
    "monitors": [
        {
            "monitorArn": "arn:aws:networkflowmonitor:us-
east-1:123456789012:monitor/Demo",
            "monitorName": "Demo",
            "monitorStatus": "ACTIVE"
        }
    ]
}
```

자세한 내용은 Amazon CloudWatch 사용 설명서에서 Network Flow Monitor의 구성 요소 및 기능을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListMonitors를 참조하세요.

list-scopes

다음 코드 예시는 `list-scopes`의 사용 방법을 보여 줍니다.

AWS CLI

범위 목록을 검색하려면

다음 `list-scopes` 예제에서는 지정된 계정의 모든 범위를 나열합니다.

```
aws networkflowmonitor list-scopes
```

출력:

```
{
    "scopes": [
        {
            "scopeId": "fdc20616-6bb4-4242-a24e-a748e65ca7ac",
            "status": "SUCCEEDED",
            "scopeArn": "arn:aws:networkflowmonitor:us-east-1:123456789012:scope/
fdc20616-6bb4-4242-a24e-a748e65ca7ac"
        }
```

```
    ]
}
```

자세한 내용은 Amazon CloudWatch 사용 설명서에서 Network Flow Monitor의 구성 요소 및 기능을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListScopes를 참조하세요.

list-tags-for-resource

다음 코드 예시는 `list-tags-for-resource`의 사용 방법을 보여 줍니다.

AWS CLI

태그를 나열하려면

다음 `list-tags-for-resource` 예제에서는 지정된 리소스에 할당된 모든 태그를 반환합니다.

```
aws networkflowmonitor list-tags-for-resource \
    --resource-arn arn:aws:networkflowmonitor:us-east-1:123456789012:monitor/Demo
```

출력:

```
{
    "tags": {
        "Value": "Production",
        "Key": "stack"
    }
}
```

자세한 내용은 Amazon CloudWatch 사용 설명서에서 Amazon CloudWatch 리소스 태깅을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource를 참조하세요.

start-query-monitor-top-contributors

다음 코드 예시는 `start-query-monitor-top-contributors`의 사용 방법을 보여 줍니다.

AWS CLI

쿼리를 시작하려면

다음 `start-query-monitor-top-contributors` 예제에서는 쿼리 ID를 반환하여 상위 기여자를 검색하는 쿼리를 시작합니다.

```
aws networkflowmonitor start-query-monitor-top-contributors \
    --monitor-name Demo \
    --start-time 2024-12-09T19:00:00Z \
    --end-time 2024-12-09T19:15:00Z \
    --metric-name DATA_TRANSFERRED \
    --destination-category UNCLASSIFIED
```

출력:

```
{
    "queryId": "aecd3a88-0283-35b0-a17d-6e944dc8531d"
}
```

자세한 내용은 Amazon CloudWatch 사용 설명서의 워크로드 인사이트로 네트워크 흐름 평가를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartQueryMonitorTopContributors 섹션을 참조하세요.

`start-query-workload-insights-top-contributors-data`

다음 코드 예시는 `start-query-workload-insights-top-contributors-data`의 사용 방법을 보여 줍니다.

AWS CLI

쿼리를 시작하려면

다음 `start-query-workload-insights-top-contributors-data` 예제에서는 쿼리 ID를 반환하여 상위 기여자를 검색하는 쿼리를 시작합니다.

```
aws networkflowmonitor start-query-workload-insights-top-contributors-data \
    --scope-id e21cda79-30a0-4c12-9299-d8629d76d8cf \
    --start-time 2024-12-09T19:00:00Z \
    --end-time 2024-12-09T19:15:00Z \
    --metric-name DATA_TRANSFERRED \
    --destination-category UNCLASSIFIED
```

출력:

```
{
    "queryId": "cc4f4ab3-3103-33b8-80ff-d6597a0c6cea"
}
```

자세한 내용은 Amazon CloudWatch 사용 설명서의 [워크로드 인사이트로 네트워크 흐름 평가](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [StartQueryWorkloadInsightsTopContributorsData](#) 섹션을 참조하세요.

`start-query-workload-insights-top-contributors`

다음 코드 예시는 `start-query-workload-insights-top-contributors`의 사용 방법을 보여 줍니다.

AWS CLI

쿼리를 시작하려면

다음 `start-query-workload-insights-top-contributors` 예제에서는 쿼리 ID를 반환하여 상위 기여자를 검색하는 쿼리를 시작합니다.

```
aws networkflowmonitor start-query-workload-insights-top-contributors \
    --scope-id e21cda79-30a0-4c12-9299-d8629d76d8cf \
    --start-time 2024-12-09T19:00:00Z \
    --end-time 2024-12-09T19:15:00Z \
    --metric-name DATA_TRANSFERRED \
    --destination-category UNCLASSIFIED
```

출력:

```
{
    "queryId": "1fc423d3-b144-37a6-80e6-e2c7d26eea0c"
}
```

자세한 내용은 Amazon CloudWatch 사용 설명서의 [워크로드 인사이트로 네트워크 흐름 평가](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [StartQueryWorkloadInsightsTopContributors](#) 섹션을 참조하세요.

`stop-query-monitor-top-contributors`

다음 코드 예시는 `stop-query-monitor-top-contributors`의 사용 방법을 보여 줍니다.

AWS CLI

쿼리를 중지하려면

다음 `stop-query-monitor-top-contributors` 예제에서는 지정된 계정의 쿼리를 중지합니다.

```
aws networkflowmonitor stop-query-monitor-top-contributors \
    --monitor-name Demo \
    --query-id aecd3a88-0283-35b0-a17d-6e944dc8531d
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon CloudWatch 사용 설명서의 워크로드 인사이트로 네트워크 흐름 평가를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StopQueryMonitorTopContributors 섹션을 참조하세요.

`stop-query-workload-insights-top-contributors-data`

다음 코드 예시는 `stop-query-workload-insights-top-contributors-data`의 사용 방법을 보여 줍니다.

AWS CLI

쿼리를 중지하려면

다음 `stop-query-workload-insights-top-contributors-data` 예제에서는 지정된 계정의 쿼리를 중지합니다.

```
aws networkflowmonitor stop-query-workload-insights-top-contributors-data \
    --scope-id e21cda79-30a0-4c12-9299-d8629d76d8cf \
    --query-id cc4f4ab3-3103-33b8-80ff-d6597a0c6cea
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon CloudWatch 사용 설명서의 워크로드 인사이트로 네트워크 흐름 평가를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StopQueryWorkloadInsightsTopContributorsData 섹션을 참조하세요.

stop-query-workload-insights-top-contributors

다음 코드 예시는 stop-query-workload-insights-top-contributors의 사용 방법을 보여 줍니다.

AWS CLI

쿼리를 중지하려면

다음 stop-query-workload-insights-top-contributors 예제에서는 지정된 계정의 쿼리를 중지합니다.

```
aws networkflowmonitor stop-query-workload-insights-top-contributors \
    --scope-id e21cda79-30a0-4c12-9299-d8629d76d8cf \
    --query-id 1fc423d3-b144-37a6-80e6-e2c7d26eea0c
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon CloudWatch 사용 설명서의 워크로드 인사이트로 네트워크 흐름 평가를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StopQueryWorkloadInsightsTopContributors 섹션을 참조하세요.

tag-resource

다음 코드 예시는 tag-resource의 사용 방법을 보여 줍니다.

AWS CLI

지정된 리소스에 태그를 추가하는 방법

다음 tag-resource 예제에서는 지정된 계정에서 모니터에 태그를 추가합니다.

```
aws networkflowmonitor tag-resource \
    --resource-arn arn:aws:networkflowmonitor:us-east-1:123456789012:monitor/Demo \
    --tags Key=stack,Value=Production
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon CloudWatch 사용 설명서에서 Amazon CloudWatch 리소스 태깅을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

untag-resource

다음 코드 예시는 untag-resource의 사용 방법을 보여 줍니다.

AWS CLI

지정된 리소스에서 태그를 제거하려면

다음 untag-resource 예제에서는 지정된 계정에서 모니터의 태그를 제거합니다.

```
aws networkflowmonitor untag-resource \
    --resource-arn arn:aws:networkflowmonitor:us-east-1:123456789012:monitor/Demo \
    --tag-keys stack
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon CloudWatch 사용 설명서에서 Amazon CloudWatch 리소스 태깅을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

update-monitor

다음 코드 예시는 update-monitor의 사용 방법을 보여 줍니다.

AWS CLI

기존 모니터를 업데이트하려면

다음 update-monitor 예제에서는 지정된 계정에서 Demo라는 모니터를 업데이트합니다.

```
aws networkflowmonitor update-monitor \
    --monitor-name Demo \
    --local-resources-to-add type="AWS::EC2::VPC",identifier="arn:aws:ec2:us-
east-1:123456789012:vpc/vpc-048d08dfbec623f94"
```

출력:

```
{
    "monitorArn": "arn:aws:networkflowmonitor:us-east-1:123456789012:monitor/Demo",
    "monitorName": "Demo",
    "monitorStatus": "ACTIVE",
    "tags": {
        "Value": "Production",
        "Key": "stack"
    }
}
```

자세한 내용은 Amazon CloudWatch 사용 설명서에서 Network Flow Monitor의 구성 요소 및 기능을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateMonitor를 참조하세요.

AWS CLI를 사용한 Network Manager 예시

다음 코드 예시에서는 Network Manager에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

associate-customer-gateway

다음 코드 예시에서는 `associate-customer-gateway` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

고객 게이트웨이를 연결하는 방법

다음 `associate-customer-gateway` 예시는 지정된 글로벌 네트워크의 고객 게이트웨이 `cgw-11223344556677889`를 디바이스 `device-07f6fd08867abc123`과 연결합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 AssociateCustomerGateway 섹션을 참조하세요.

associate-link

다음 코드 예시에서는 `associate-link` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

링크를 연결하는 방법

다음 `associate-link` 예시에서는 `link-11112222aaaabbbb1` 링크를 디바이스 `device-07f6fd08867abc123`과 연결합니다. 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 안내서의 Device and Link Associations를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AssociateLink 섹션을 참조하세요.

create-core-network

다음 코드 예시에서는 `create-core-network` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

코어 네트워크 생성

다음 `create-core-network` 예시에서는 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 Cloud WAN 사용 설명서의 글로벌 및 코어 네트워크를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateCoreNetwork 섹션을 참조하세요.

`create-device`

다음 코드 예시에서는 `create-device` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

디바이스 생성

다음 `create-device` 예시에서는 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 안내서의 Working with Devices를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateDevice 섹션을 참조하세요.

create-global-network

다음 코드 예시에서는 `create-global-network` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

글로벌 네트워크 생성

다음 `create-global-network` 예시에서는 새로운 글로벌 네트워크를 생성합니다. 생성 시 초기 상태는 `PENDING`입니다.

```
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"
    }
```

```
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateGlobalNetwork 섹션을 참조하세요.

create-link

다음 코드 예시에서는 create-link 코드를 사용하는 방법을 보여줍니다.

AWS CLI

링크 생성

다음 create-link 예시에서는 지정된 글로벌 네트워크에 링크를 생성합니다. 링크에는 링크 유형, 대역폭 및 제공업체에 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 안내서의 Working with Links를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateLink 섹션을 참조하세요.

create-site

다음 코드 예시에서는 create-site 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사이트 생성

다음 create-site 예시에서는 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 안내서의 Working with Sites를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateSite 섹션을 참조하세요.

create-vpc-attachment

다음 코드 예시에서는 `create-vpc-attachment` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

VPC 연결 생성

다음 `create-vpc-attachment` 예시에서는 코어 네트워크에서 IPv6를 지원하는 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 Cloud WAN 사용 설명서의 Create an attachment를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateVpcAttachment 섹션을 참조하세요.

delete-attachment

다음 코드 예시에서는 delete-attachment 코드를 사용하는 방법을 보여줍니다.

AWS CLI

첨부 파일을 삭제하는 방법

다음 delete-attachment 예시에서는 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 Cloud WAN 사용 설명서의 Delete attachments를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteAttachment 섹션을 참조하세요.

`delete-bucket-analytics-configuration`

다음 코드 예시에서는 `delete-bucket-analytics-configuration` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

버킷의 분석 구성 삭제

다음 `delete-bucket-analytics-configuration` 예시에서는 지정된 버킷 및 ID에 대한 분석 구성을 제거합니다.

```
aws s3api delete-bucket-analytics-configuration \
    --bucket amzn-s3-demo-bucket \
    --id 1
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteBucketAnalyticsConfiguration을 참조하세요.

`delete-bucket-metrics-configuration`

다음 코드 예시에서는 `delete-bucket-metrics-configuration` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

버킷의 지표 구성 삭제

다음 `delete-bucket-metrics-configuration` 예시에서는 지정된 버킷 및 ID에 대한 지표 구성을 제거합니다.

```
aws s3api delete-bucket-metrics-configuration \
    --bucket amzn-s3-demo-bucket \
    --id 123
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteBucketMetricsConfiguration을 참조하세요.

delete-core-network

다음 코드 예시에서는 delete-core-network 코드를 사용하는 방법을 보여줍니다.

AWS CLI

코어 네트워크를 삭제하는 방법

다음 delete-core-network 예시에서는 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 Cloud WAN 사용 설명서의 Core networks를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteCoreNetwork 섹션을 참조하세요.

delete-device

다음 코드 예시에서는 `delete-device` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

디바이스 삭제

다음 `delete-device` 예시에서는 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 안내서의 Working with Devices를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteDevice 섹션을 참조하세요.

delete-global-network

다음 코드 예시에서는 delete-global-network 코드를 사용하는 방법을 보여줍니다.

AWS CLI

글로벌 네트워크 삭제

다음 delete-global-network 예시에서는 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteGlobalNetwork 섹션을 참조하세요.

delete-link

다음 코드 예시에서는 delete-link 코드를 사용하는 방법을 보여줍니다.

AWS CLI

링크 삭제

다음 delete-link 예시에서는 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 안내서의 Working with Links를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteLink 섹션을 참조하세요.

delete-public-access-block

다음 코드 예시에서는 `delete-public-access-block` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

버킷의 퍼블릭 액세스 차단 구성 삭제

다음 `delete-public-access-block` 예시에서는 지정된 버킷에서 퍼블릭 액세스 차단 구성을 제거합니다.

```
aws s3api delete-public-access-block \
    --bucket amzn-s3-demo-bucket
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 DeletePublicAccessBlock을 참조하세요.

delete-site

다음 코드 예시에서는 delete-site 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사이트 삭제

다음 delete-site 예시에서는 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 안내서의 Working with Sites를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteSite 섹션을 참조하세요.

deregister-transit-gateway

다음 코드 예시에서는 deregister-transit-gateway 코드를 사용하는 방법을 보여줍니다.

AWS CLI

글로벌 네트워크에서 전송 게이트웨이의 등록 취소

다음 `deregister-transit-gateway` 예시에서는 지정된 글로벌 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 Registrations를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeregisterTransitGateway 섹션을 참조하세요.

`describe-global-networks`

다음 코드 예시에서는 `describe-global-networks` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

글로벌 네트워크를 설명하는 방법

다음 `describe-global-networks` 예시에서는 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"
        }
    ]
}
```


- API 세부 정보는 AWS CLI 명령 참조의 [DescribeGlobalNetworks](https://docs.aws.amazon.com/goto/aws-cli/networkmanager-2019-07-05/DescribeGlobalNetworks) 섹션을 참조하세요.

disassociate-customer-gateway

다음 코드 예시에서는 disassociate-customer-gateway 코드를 사용하는 방법을 보여줍니다.

AWS CLI

고객 게이트웨이 연결을 해제하는 방법

다음 disassociate-customer-gateway 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 DisassociateCustomerGateway 섹션을 참조하세요.

disassociate-link

다음 코드 예시에서는 disassociate-link 코드를 사용하는 방법을 보여줍니다.

AWS CLI

링크의 연결 해제

다음 disassociate-link 예시에서는 지정된 글로벌 네트워크의 디바이스 device-07f6fd08867abc123에서 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 안내서의 Device and Link Associations를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisassociateLink 섹션을 참조하세요.

get-bucket-analytics-configuration

다음 코드 예시에서는 get-bucket-analytics-configuration 코드를 사용하는 방법을 보여줍니다.

AWS CLI

특정 ID를 가진 버킷의 분석 구성 가져오기

다음 get-bucket-analytics-configuration 예시에서는 지정된 버킷 및 ID에 대한 분석 구성을 표시합니다.

```
aws s3api get-bucket-analytics-configuration \
    --bucket amzn-s3-demo-bucket \
    --id 1
```

출력:

```
{
    "AnalyticsConfiguration": {
        "StorageClassAnalysis": {},
        "Id": "1"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetBucketAnalyticsConfiguration을 참조하세요.

get-bucket-metrics-configuration

다음 코드 예시에서는 get-bucket-metrics-configuration 코드를 사용하는 방법을 보여줍니다.

AWS CLI

특정 ID를 가진 버킷의 지표 구성 가져오기

다음 get-bucket-metrics-configuration 예시에서는 지정된 버킷 및 ID의 지표 구성을 표시합니다.

```
aws s3api get-bucket-metrics-configuration \
```

```
    --bucket amzn-s3-demo-bucket \
    --id 123
```

출력:

```
{
    "MetricsConfiguration": {
        "Filter": {
            "Prefix": "logs"
        },
        "Id": "123"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetBucketMetricsConfiguration을 참조하세요.

get-customer-gateway-associations

다음 코드 예시에서는 get-customer-gateway-associations 코드를 사용하는 방법을 보여줍니다.

AWS CLI

고객 게이트웨이 연결을 가져오는 방법

다음 get-customer-gateway-associations 예시에서는 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"
            }
        ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetCustomerGatewayAssociations 섹션을 참조하세요.

get-devices

다음 코드 예시에서는 get-devices 코드를 사용하는 방법을 보여줍니다.

AWS CLI

디바이스를 가져오는 방법

다음 get-devices 예시에서는 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetDevices 섹션을 참조하세요.

get-link-associations

다음 코드 예시에서는 get-link-associations 코드를 사용하는 방법을 보여줍니다.

AWS CLI

링크 연결을 가져오는 방법

다음 get-link-associations 예시에서는 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetLinkAssociations 섹션을 참조하세요.

get-links

다음 코드 예시에서는 get-links 코드를 사용하는 방법을 보여줍니다.

AWS CLI

링크를 가져오는 방법

다음 get-links 예시에서는 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetLinks 섹션을 참조하세요.

get-object-retention

다음 코드 예시에서는 get-object-retention 코드를 사용하는 방법을 보여줍니다.

AWS CLI

객체에 대한 객체 보존 구성 가져오기

다음 get-object-retention 예시에서는 지정된 객체에 대한 객체 보존 구성을 가져옵니다.

```
aws s3api get-object-retention \
    --bucket amzn-s3-demo-bucket-with-object-lock \
    --key doc1.rtf
```

출력:

```
{
```

```
    "Retention": {
        "Mode": "GOVERNANCE",
        "RetainUntilDate": "2025-01-01T00:00:00.000Z"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetObjectRetention을 참조하세요.

get-public-access-block

다음 코드 예시에서는 get-public-access-block 코드를 사용하는 방법을 보여줍니다.

AWS CLI

버킷의 퍼블릭 액세스 차단 구성을 설정하거나 수정

다음 get-public-access-block 예시에서는 지정된 버킷의 퍼블릭 액세스 차단 구성을 표시합니다.

```
aws s3api get-public-access-block --bucket amzn-s3-demo-bucket
```

출력:

```
{
    "PublicAccessBlockConfiguration": {
        "IgnorePublicAcls": true,
        "BlockPublicPolicy": true,
        "BlockPublicAcls": true,
        "RestrictPublicBuckets": true
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetPublicAccessBlock을 참조하세요.

get-sites

다음 코드 예시에서는 get-sites 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사이트를 가져오는 방법

다음 `get-sites` 예시에서는 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetSites 섹션을 참조하세요.

`get-transit-gateway-registrations`

다음 코드 예시에서는 `get-transit-gateway-registrations` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

전송 게이트웨이 등록 가져오기

다음 `get-transit-gateway-registrations` 예시에서는 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"
            }
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetTransitGatewayRegistrations 섹션을 참조하세요.

get-vpc-attachment

다음 코드 예시에서는 get-vpc-attachment 코드를 사용하는 방법을 보여줍니다.

AWS CLI

VPC 연결 가져오기

다음 get-vpc-attachment 예시에서는 VPC 연결에 관한 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 Cloud WAN 사용 설명서의 Attachments를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetVpcAttachment 섹션을 참조하세요.

list-bucket-analytics-configurations

다음 코드 예시에서는 `list-bucket-analytics-configurations` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

버킷의 분석 구성 목록 가져오기

다음 `list-bucket-analytics-configurations`는 지정된 버킷의 분석 구성 목록을 가져옵니다.

```
aws s3api list-bucket-analytics-configurations \
    --bucket amzn-s3-demo-bucket
```

출력:

```
{
    "AnalyticsConfigurationList": [
        {
            "StorageClassAnalysis": {},
            "Id": "1"
        }
    ],
    "IsTruncated": false
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListBucketAnalyticsConfigurations를 참조하세요.

`list-bucket-metrics-configurations`

다음 코드 예시에서는 `list-bucket-metrics-configurations` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

버킷의 지표 구성 목록 검색

다음 `list-bucket-metrics-configurations` 예시에서는 지정된 버킷에 대한 지표 구성 목록을 검색합니다.

```
aws s3api list-bucket-metrics-configurations \
    --bucket amzn-s3-demo-bucket
```

출력:

```
{
    "IsTruncated": false,
    "MetricsConfigurationList": [
        {
            "Filter": {
                "Prefix": "logs"
            },
            "Id": "123"
        },
        {
            "Filter": {
                "Prefix": "tmp"
```

```
            },
            "Id": "234"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListBucketInventoryConfigurations 섹션을 참조하세요.

list-tags-for-resource

다음 코드 예시에서는 `list-tags-for-resource` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

리소스에 대한 태그 나열

다음 `list-tags-for-resource` 예시에서는 지정된 디바이스 리소스 (`device-07f6fd08867abc123`)의 태그를 나열합니다.

```
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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource 섹션을 참조하세요.

put-bucket-metrics-configuration

다음 코드 예시에서는 `put-bucket-metrics-configuration` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

버킷의 지표 구성 설정

다음 `put-bucket-metrics-configuration` 예시에서는 지정된 버킷의 ID 123에 대한 지표 구성을 설정합니다.

```
aws s3api put-bucket-metrics-configuration \
    --bucket amzn-s3-demo-bucket \
    --id 123 \
    --metrics-configuration '{"Id": "123", "Filter": {"Prefix": "logs"}}'
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 PutBucketLifecycleConfiguration 섹션을 참조하세요.

`put-object-retention`

다음 코드 예시에서는 `put-object-retention` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

객체에 대한 객체 보존 구성 설정

다음 `put-object-retention` 예시에서는 지정된 객체에 대한 객체 보존 구성을 2025년 1월 1일 전까지로 설정합니다.

```
aws s3api put-object-retention \
    --bucket amzn-s3-demo-bucket-with-object-lock \
    --key doc1.rtf \
    --retention '{ "Mode": "GOVERNANCE", "RetainUntilDate": "2025-01-01T00:00:00" }'
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 PutObjectRetention을 참조하세요.

`put-public-access-block`

다음 코드 예시에서는 `put-public-access-block` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

버킷의 퍼블릭 액세스 차단 구성 설정

다음 put-public-access-block 예시는 지정된 버킷에 대한 퍼블릭 액세스 차단 구성을 설정합니다.

```
aws s3api put-public-access-block \
    --bucket amzn-s3-demo-bucket \
    --public-access-block-
configuration "BlockPublicAcls=true,IgnorePublicAcls=true,BlockPublicPolicy=true,RestrictPub
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 PutPublicAccessBlock 섹션을 참조하세요.

register-transit-gateway

다음 코드 예시에서는 register-transit-gateway 코드를 사용하는 방법을 보여줍니다.

AWS CLI

글로벌 네트워크에 전송 게이트웨이 등록

다음 register-transit-gateway 예시에서는 지정된 글로벌 네트워크에 전송 게이트웨이를 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 안내서의 Transit Gateway Registrations를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RegisterTransitGateway 섹션을 참조하세요.

reject-attachment

다음 코드 예시에서는 `reject-attachment` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

첨부 파일을 거부하는 방법

다음 `reject-attachment` 예시에서는 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 Cloud WAN 사용 설명서의 Attachment acceptance를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RejectAttachment 섹션을 참조하세요.

start-route-analysis

다음 코드 예시에서는 `start-route-analysis` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

경로 분석을 시작하는 방법

다음 `start-route-analysis` 예시에서는 선택적 `include-return-path`를 포함하여 소스와 대상 간의 분석을 시작합니다.

```
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 AWS Global Networks for Transit Gateways 사용 설명서의 Route Analyzer를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartRouteAnalysis 섹션을 참조하세요.

tag-resource

다음 코드 예시에서는 tag-resource 코드를 사용하는 방법을 보여줍니다.

AWS CLI

리소스에 태그 적용

다음 tag-resource 예시에서는 Network=Northeast 태그를 디바이스 device-07f6fd08867abc123에 적용합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

untag-resource

다음 코드 예시에서는 untag-resource 코드를 사용하는 방법을 보여줍니다.

AWS CLI

리소스에서 태그 제거

다음 untag-resource 예시에서는 디바이스 device-07f6fd08867abc123에서 Network 키가 있는 태그를 제거합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

update-device

다음 코드 예시에서는 update-device 코드를 사용하는 방법을 보여줍니다.

AWS CLI

디바이스 업데이트

다음 update-device 예시에서는 디바이스 device-07f6fd08867abc123의 사이트 ID를 지정하여 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 안내서의 Working with Devices를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateDevice 섹션을 참조하세요.

update-global-network

다음 코드 예시에서는 update-global-network 코드를 사용하는 방법을 보여줍니다.

AWS CLI

글로벌 네트워크 업데이트

다음 update-global-network 예시에서는 글로벌 네트워크 global-network-01231231231231231에 대한 설명을 업데이트합니다.

```
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 안내서의 Global Networks를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateGlobalNetwork 섹션을 참조하세요.

update-link

다음 코드 예시에서는 update-link 코드를 사용하는 방법을 보여줍니다.

AWS CLI

링크 업데이트

다음 update-link 예시에서는 링크 link-11112222aaaabbbb1에 대한 대역폭 정보를 업데이트합니다.

```
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 안내서의 Working with Links를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateLink 섹션을 참조하세요.

update-site

다음 코드 예시에서는 update-site 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사이트 업데이트

다음 update-site 예시에서는 지정된 글로벌 네트워크의 사이트 site-444555aaabbb11223에 대한 설명을 업데이트합니다.

```
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 안내서의 Working with Sites를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateSite 섹션을 참조하세요.

AWS CLI를 사용한 OpenSearch 서비스 예제

다음 코드 예제에서는 OpenSearch Service에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여 줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`create-elasticsearch-domain`

다음 코드 예시는 `create-elasticsearch-domain`의 사용 방법을 보여 줍니다.

AWS CLI

데이터를 Amazon Elasticsearch Service 도메인에 작성하는 방법

다음 `create-elasticsearch-domain` 명령은 VPC 내에 새 Amazon Elasticsearch Service 도메인을 생성하고 단일 사용자로 액세스를 제한합니다. Amazon ES는 지정된 서브넷 및 보안 그룹 ID에서 VPC ID를 추론합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateElasticsearchDomain을 참조하세요.

describe-elasticsearch-domain-config

다음 코드 예시는 describe-elasticsearch-domain-config의 사용 방법을 보여 줍니다.

AWS CLI

도메인 구성 세부 정보를 얻는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeElasticsearchDomainConfig를 참조하세요.

describe-elasticsearch-domain

다음 코드 예시는 describe-elasticsearch-domain의 사용 방법을 보여 줍니다.

AWS CLI

단일 도메인에 대한 세부 정보를 얻는 방법

다음 describe-elasticsearch-domain 예제에서는 주어진 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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeElasticsearchDomain을 참조하세요.

describe-elasticsearch-domains

다음 코드 예시는 describe-elasticsearch-domains의 사용 방법을 보여 줍니다.

AWS CLI

하나 이상의 도메인에 대한 세부 정보를 얻는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeElasticsearchDomains를 참조하세요.

describe-reserved-elasticsearch-instances

다음 코드 예시는 describe-reserved-elasticsearch-instances의 사용 방법을 보여 줍니다.

AWS CLI

예약된 모든 인스턴스를 보는 방법

다음 describe-elasticsearch-domains 예제에서는 리전에서 예약한 모든 인스턴스의 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 예약형 인스턴스를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeReservedElasticsearchInstances를 참조하세요.

list-domain-names

다음 코드 예시는 `list-domain-names`의 사용 방법을 보여 줍니다.

AWS CLI

도메인 목록을 표시하는 방법

다음 `list-domain-names` 예제에서는 리전 내 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListDomainNames를 참조하세요.

AWS CLI를 사용한 OpsWorks 예시

다음 코드 예시에서는 OpsWorks에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`assign-instance`

다음 코드 예시에서는 `assign-instance`의 사용 방법을 보여줍니다.

AWS CLI

계층에 등록된 인스턴스 할당

다음 예시에서는 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 Assigning a Registered Instance to a Layer 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AssignInstance 섹션을 참조하세요.

assign-volume

다음 코드 예시에서는 assign-volume의 사용 방법을 보여줍니다.

AWS CLI

인스턴스에 등록된 볼륨 할당

다음 예시에서는 등록된 Amazon Elastic Block Store(Amazon EBS) 볼륨을 인스턴스에 할당합니다. 볼륨은 Amazon Elastic Compute Cloud(Amazon EC2) 볼륨 ID가 아닌 스택에 볼륨을 등록할 때 AWS OpsWorks가 할당하는 GUID인 볼륨 ID로 식별됩니다. assign-volume을 실행하기 전에 먼저 update-volume을 실행하여 볼륨에 마운트 포인트를 할당해야 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 Assigning Amazon EBS Volumes to an Instance 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AssignVolume를 참조하세요.

associate-elastic-ip

다음 코드 예시에서는 associate-elastic-ip의 사용 방법을 보여줍니다.

AWS CLI

인스턴스와 탄력적 IP 주소 연결

다음 예시에서는 로 표현되는 인스턴스와 탄력적 IP 주소를 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 Resource Management 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AssociateElasticIp 섹션을 참조하세요.

`attach-elastic-load-balancer`

다음 코드 예시에서는 `attach-elastic-load-balancer`의 사용 방법을 보여줍니다.

AWS CLI

계층에 로드 밸런서 연결

다음 예시에서는 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 Balancing 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AttachElasticLoadBalancer 섹션을 참조하세요.

`create-app`

다음 코드 예시에서는 `create-app`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 앱 생성

다음 예시에서는 GitHub 리포지토리에 저장된 코드에서 SimplePHPApp이라는 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"
}
```

예시 2: 데이터베이스가 연결된 앱 생성

다음 예시에서는 퍼블릭 S3 버킷의 .zip 아카이브에 저장된 코드에서 JSP 앱을 생성합니다. RDS DB 인스턴스를 연결하여 앱의 데이터 저장소 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 `appsource.json`에 있으며 다음을 포함합니다.

```
{
    "Type": "archive",
    "Url": "https://s3.amazonaws.com/opsworks-demo-assets/simplejsp.zip"
}
```

데이터베이스 소스 정보는 `datasource.json`에 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 `Type`을 `OpsworksMysqlInstance`로 설정합니다. 이러한 인스턴스는 AWS OpsWorks에서 생성되므로 등록할 필요가 없습니다.

출력:

```
{
    "AppId": "26a61ead-d201-47e3-b55c-2a7c666942f8"
}
```

자세한 내용은 AWS OpsWorks 사용 설명서의 Adding Apps 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [CreateApp](https://docs.aws.amazon.com/goto/aws-cli/opsworks-2013-02-18/CreateApp)을 참조하세요.

`create-deployment`

다음 코드 예시에서는 `create-deployment`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 앱을 배포하고 스택 명령 실행

다음 예시에서는 `create-deployment` 명령을 사용하여 앱을 배포하고 스택 명령을 실행하는 방법을 보여줍니다. 명령을 지정하는 JSON 객체의 따옴표(") 문자 앞에는 모두 이스케이프 문자(\)가 추가됩니다. 이스케이프 문자가 없으면 명령이 잘못된 JSON 오류를 반환할 수 있습니다.

다음 `create-deployment` 예시에서는 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"
}
```

예시 2: Rails 앱을 배포하고 데이터베이스를 마이그레이션

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.

예시 3: 레시피 실행

다음 `create-deployment` 명령은 지정된 스택의 인스턴스에서 사용자 지정 레시피인 `phpapp::appsetup`을 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 Run Stack Commands 섹션을 참조하세요.

예시 4: 종속성 설치

다음 `create-deployment` 명령은 지정된 스택의 인스턴스에 패키지 또는 Ruby gem과 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 Run Stack Commands 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateDeployment를 참조하세요.

`create-instance`

다음 코드 예시에서는 `create-instance`의 사용 방법을 보여줍니다.

AWS CLI

인스턴스 생성

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 get-hostname-suggestion을 호출합니다. 이 이름은 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 Adding an Instance to a Layer 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateInstance 섹션을 참조하세요.

create-layer

다음 코드 예시에서는 create-layer의 사용 방법을 보여줍니다.

AWS CLI

계층 생성

다음 create-layer 명령은 지정된 스택에 MyPHPLayer라는 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 How to Create a Layer 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateLayer 섹션을 참조하세요.

create-server

다음 코드 예시에서는 create-server의 사용 방법을 보여줍니다.

AWS CLI

서버 생성

다음 `create-server` 예시에서는 기본 리전`automate-06`에 이름이 지정된 새 Chef Automate 서버를 생성합니다. 기본값은 유지할 백업 수, 유지 관리 및 백업 시작 시간과 같은 대부분의 다른 설정에 사용됩니다. `create-server` 명령을 실행하기 전에 AWS OpsWorks for Chef Automate 사용자 안내서의 [AWS OpsWorks for Chef Automate 시작하기](https://docs.aws.amazon.com/opsworks/latest/userguide/opscm-getting-started.html)에서 사전 조건을 완료합니다.

```
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 AWS Chef Automate용 OpsWorks API 참조의 CreateServer 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateServer 섹션을 참조하세요.

create-stack

다음 코드 예시에서는 create-stack의 사용 방법을 보여줍니다.

AWS CLI

스택 생성

다음 create-stack 명령은 CLI 스택이라는 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 AWS OpsWorks에서 생성하는 파라미터를 사용합니다. 계정의 Amazon 리소스 이름(ARNs)을 가져오려면 IAM 콘솔로 이동하여 탐색 패널에서 Roles를 선택하고 역할 또는 프로필을 선택한 다음 Summary 탭을 선택합니다.

출력:

```
{
  "StackId": "f6673d70-32e6-4425-8999-265dd002fec7"
}
```

추가 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 Create a New Stack 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateStack 섹션을 참조하세요.

`create-user-profile`

다음 코드 예시에서는 `create-user-profile`의 사용 방법을 보여줍니다.

AWS CLI

사용자 프로필 생성

create-user-profile을 직접적으로 호출해 사용자 프로파일을 생성하여 AWS Identity and Access Manager(IAM) 사용자를 AWS OpsWorks로 가져옵니다. 다음 예시에서는 Amazon 리소스 이름(ARN)으로 식별되는 cli-user-test IAM 사용자의 사용자 프로파일을 생성합니다. 이 예시에서는 사용자에게 `myusername`의 SSH 사용자 이름을 할당하고 자체 관리를 활성화하여 사용자가 SSH 퍼블릭 키를 지정할 수 있도록 합니다.

```
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 IAM 사용자를 AWS OpsWorks로 가져오지만 연결된 정책에서 부여한 권한만 갖도록 합니다. `set-permissions` 명령을 사용하여 스택별 AWS OpsWorks 권한을 부여할 수 있습니다.

추가 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 Importing Users into AWS OpsWorks 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateUserProfile 섹션을 참조하세요.

`delete-app`

다음 코드 예시에서는 `delete-app`의 사용 방법을 보여줍니다.

AWS CLI

앱 삭제

다음 예시에서는 앱 ID로 식별되는 지정된 앱을 삭제합니다. AWS OpsWorks 콘솔에서 앱의 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 Apps 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteApp을 참조하세요.

`delete-instance`

다음 코드 예시에서는 `delete-instance`의 사용 방법을 보여줍니다.

AWS CLI

인스턴스 삭제

다음 `delete-instance` 예시에서는 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 `stop-instance`를 직접적으로 호출하여 인스턴스를 중지한 다음 인스턴스가 중지될 때까지 기다려야 합니다. `describe-instances`를 실행하여 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 Deleting AWS OpsWorks Instances 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteInstance 섹션을 참조하세요.

delete-layer

다음 코드 예시에서는 delete-layer의 사용 방법을 보여줍니다.

AWS CLI

계층 삭제

다음 예시에서는 지정된 계층을 삭제하며 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-instance를 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 Deleting AWS OpsWorks Instances 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteLayer 섹션을 참조하세요.

delete-stack

다음 코드 예시에서는 delete-stack의 사용 방법을 보여줍니다.

AWS CLI

스택 삭제

다음 예시에서는 스택 ID로 식별되는 지정된 스택을 삭제합니다. AWS OpsWorks 콘솔에서 스택 설정을 클릭하거나 describe-stacks 명령을 실행하여 스택 ID를 얻을 수 있습니다.

참고: 계층을 삭제하기 전에 delete-app, delete-instance 및 delete-layer를 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 Shut Down a Stack 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteStack을 참조하세요.

delete-user-profile

다음 코드 예시에서는 delete-user-profile의 사용 방법을 보여줍니다.

AWS CLI

사용자 프로필을 삭제하고 AWS OpsWorks에서 IAM 사용자를 제거하는 방법

다음 예시에서는 지정된 AWS ID 및 Amazon 리소스 이름(ARN)으로 식별되는 액세스 관리(IAM) 사용자의 사용자 프로파일을 삭제합니다. 작업은 AWS OpsWorks에서 사용자를 제거하지만 IAM 사용자를 삭제하지는 않습니다. 해당 태스크에 IAM 콘솔, CLI 또는 API를 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 AWS OpsWorks 사용 설명서의 Importing Users into AWS OpsWorks 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteUserProfile 섹션을 참조하세요.

deregister-elastic-ip

다음 코드 예시에서는 deregister-elastic-ip의 사용 방법을 보여줍니다.

AWS CLI

스택에서 탄력적 IP 주소 등록 취소

다음 예시에서는 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 Deregistering Elastic IP Addresses 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeregisterElasticIp 섹션을 참조하세요.

deregister-instance

다음 코드 예시에서는 deregister-instance의 사용 방법을 보여줍니다.

AWS CLI

스택에서 등록된 인스턴스를 등록 취소하는 방법

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 Deregistering a Registered Instance 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeregisterInstance 섹션을 참조하세요.

deregister-rds-db-instance

다음 코드 예시에서는 deregister-rds-db-instance의 사용 방법을 보여줍니다.

AWS CLI

스택에서 Amazon RDS DB 인스턴스 등록을 취소하는 방법

다음 예시에서는 ARN으로 식별되는 RDS DB 인스턴스를 스택에서 등록 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 Deregistering Amazon RDS Instances 섹션을 참조하세요.

instance ID: clitestdb Master usernams: cliuser Master PWD: some23!pwd DB Name: mydb aws opsworks deregister-rds-db-instance --region us-east-1 --rds-db-instance-arn arn:aws:rds:us-west-2:645732743964:db:clitestdb

- API 세부 정보는 AWS CLI 명령 참조의 DeregisterRdsDbInstance 섹션을 참조하세요.

`deregister-volume`

다음 코드 예시에서는 `deregister-volume`의 사용 방법을 보여줍니다.

AWS CLI

Amazon EBS 볼륨 등록 취소

다음 예시에서는 스택에서 EBS 볼륨의 등록을 취소합니다. 볼륨은 EC2 볼륨 ID가 아닌 스택에 볼륨을 등록할 때 AWS OpsWorks가 할당한 GUID인 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 Deregistering Amazon EBS Volumes 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeregisterVolume 섹션을 참조하세요.

describe-apps

다음 코드 예시에서는 describe-apps의 사용 방법을 보여줍니다.

AWS CLI

앱을 설명하는 방법

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 Apps 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeApps](#) 섹션을 참조하세요.

describe-commands

다음 코드 예시에서는 describe-commands의 사용 방법을 보여줍니다.

AWS CLI

명령을 설명하는 방법

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 Lifecycle Events 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeCommands 섹션을 참조하세요.

describe-deployments

다음 코드 예시에서는 describe-deployments의 사용 방법을 보여줍니다.

AWS CLI

배포를 설명하는 방법

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 Deploying Apps 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDeployments 섹션을 참조하세요.

describe-elastic-ips

다음 코드 예시에서는 describe-elastic-ips의 사용 방법을 보여줍니다.

AWS CLI

탄력적 IP 인스턴스를 설명하는 방법

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 Instances 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeElasticIps 섹션을 참조하세요.

describe-elastic-load-balancers

다음 코드 예시에서는 describe-elastic-load-balancers의 사용 방법을 보여줍니다.

AWS CLI

스택의 탄력적 로드 밸런서를 설명하는 방법

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 Apps 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeElasticLoadBalancers 섹션을 참조하세요.

describe-instances

다음 코드 예시에서는 describe-instances의 사용 방법을 보여줍니다.

AWS CLI

인스턴스 설명

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 예시는 인스턴스가 두 개 있는 스택에 대한 것입니다. 첫 번째 인스턴스는 등록된 EC2 인스턴스이고 두 번째 인스턴스는 AWS OpsWorks에서 생성되었습니다.

```
{
  "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 Instances 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeInstances를 참조하세요.

describe-layers

다음 코드 예시에서는 describe-layers의 사용 방법을 보여줍니다.

AWS CLI

스택의 계층을 설명하는 방법

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 Layers 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeLayers 섹션을 참조하세요.

describe-load-based-auto-scaling

다음 코드 예시에서는 `describe-load-based-auto-scaling`의 사용 방법을 보여줍니다.

AWS CLI

계층의 로드 기반 스케일링 구성 설명

다음 예시에서는 지정된 계층의 로드 기반 조정 구성을 설명합니다. 계층은 계층의 세부 정보 페이지 또는 `describe-layers` 실행을 통해 찾을 수 있는 계층 ID로 식별됩니다.

```
aws opsworks describe-load-based-auto-scaling --region us-east-1 --layer-ids 6bec29c9-c866-41a0-aba5-fa3e374ce2a1
```

출력 : 예시 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 How Automatic Load-based Scaling Works 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeLoadBasedAutoScaling 섹션을 참조하세요.

describe-my-user-profile

다음 코드 예시에서는 `describe-my-user-profile`의 사용 방법을 보여줍니다.

AWS CLI

사용자 프로필을 가져오는 방법

다음 예시에서는 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 대부분이 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 AWS OpsWorks 사용 설명서의 Importing Users into AWS OpsWorks 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeMyUserProfile 섹션을 참조하세요.

describe-permissions

다음 코드 예시에서는 `describe-permissions`의 사용 방법을 보여줍니다.

AWS CLI

사용자의 스택별 AWS OpsWorks 권한 수준을 얻으려면

다음 예시에서는 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 Granting Per-Stack Permissions Levels 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribePermissions 섹션을 참조하세요.

describe-raid-arrays

다음 코드 예시에서는 describe-raid-arrays의 사용 방법을 보여줍니다.

AWS CLI

RAID 배열을 설명하는 방법

다음 예시에서는 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 Volumes 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeRaidArrays 섹션을 참조하세요.

describe-rds-db-instances

다음 코드 예시에서는 `describe-rds-db-instances`의 사용 방법을 보여줍니다.

AWS CLI

스택의 등록된 Amazon RDS 인스턴스를 설명하는 방법

다음 예시에서는 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 등록된 하나의 RDS 인스턴스가 있는 스택의 출력입니다.

```
{
  "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 Resource Management 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeRdsDbInstances 섹션을 참조하세요.

describe-stack-provisioning-parameters

다음 코드 예시에서는 `describe-stack-provisioning-parameters`의 사용 방법을 보여줍니다.

AWS CLI

스택에 대한 프로비저닝 파라미터를 반환하는 방법

다음 `describe-stack-provisioning-parameters` 예시에서는 지정된 스택에 대한 프로비저닝 파라미터를 반환합니다. 프로비저닝 파라미터에는 OpsWorks가 스택의 인스턴스에서 에이전트를 관리하는 데 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 Run Stack Commands 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeStackProvisioningParameters 섹션을 참조하세요.

`describe-stack-summary`

다음 코드 예시에서는 `describe-stack-summary`의 사용 방법을 보여줍니다.

AWS CLI

스택의 구성을 설명하는 방법

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 Stacks 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeStackSummary 섹션을 참조하세요.

describe-stacks

다음 코드 예시에서는 describe-stacks의 사용 방법을 보여줍니다.

AWS CLI

스택 설명

다음 describe-stacks 명령은 계정의 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 Stacks 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeStacks 섹션을 참조하세요.

describe-timebased-auto-scaling

다음 코드 예시에서는 describe-timebased-auto-scaling의 사용 방법을 보여줍니다.

AWS CLI

인스턴스의 시간 기반 스케일링 구성 설명

다음 예시에서는 지정된 인스턴스의 시간 기반 조정 구성을 설명합니다. 인스턴스는 인스턴스의 세부 정보 페이지 또는 describe-instances 실행을 통해 찾을 수 있는 인스턴스 ID로 식별됩니다.

```
aws opsworks describe-time-based-auto-scaling --region us-east-1 --instance-ids 701f2ffe-5d8e-4187-b140-77b75f55de8d
```

출력 : 이 예시에는 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 How Automatic Time-based Scaling Works 섹션을 참조하세요.

- sdAPI 세부 정보는 AWS CLI 명령 참조의 DescribeTimebasedAutoScaling 섹션을 참조하세요.

describe-user-profiles

다음 코드 예시에서는 describe-user-profiles의 사용 방법을 보여줍니다.

AWS CLI

사용자 프로필을 설명하는 방법

다음 describe-user-profiles 명령은 계정의 사용자 프로파일을 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 Managing AWS OpsWorks Users 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeUserProfiles 섹션을 참조하세요.

describe-volumes

다음 코드 예시에서는 describe-volumes의 사용 방법을 보여줍니다.

AWS CLI

스택의 볼륨을 설명하는 방법

다음 예시에서는 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 Resource Management 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeVolumes 섹션을 참조하세요.

detach-elastic-load-balancer

다음 코드 예시에서는 detach-elastic-load-balancer의 사용 방법을 보여줍니다.

AWS CLI

계층에서 로드 밸런서 분리

다음 예시에서는 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 Balancing 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DetachElasticLoadBalancer 섹션을 참조하세요.

disassociate-elastic-ip

다음 코드 예시에서는 disassociate-elastic-ip의 사용 방법을 보여줍니다.

AWS CLI

인스턴스에서 탄력적 IP 주소 연결 해제

이 예시에서는 VPC의 인스턴스에서 탄력적 IP 주소를 연결 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 Resource Management 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisassociateElasticIp 섹션을 참조하세요.

get-hostname-suggestion

다음 코드 예시에서는 get-hostname-suggestion의 사용 방법을 보여줍니다.

AWS CLI

계층의 다음 호스트 이름을 가져오는 방법

다음 예시에서는 지정된 계층에 대해 다음 번 생성된 호스트 이름을 가져옵니다. 이 예시에 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 Create a New Stack 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetHostnameSuggestion 섹션을 참조하세요.

reboot-instance

다음 코드 예시에서는 reboot-instance의 사용 방법을 보여줍니다.

AWS CLI

인스턴스 재부팅

다음 예시에서는 인스턴스를 다시 시작합니다.

```
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 Rebooting an Instance 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RebootInstance 섹션을 참조하세요.

register-elastic-ip

다음 코드 예시에서는 register-elastic-ip의 사용 방법을 보여줍니다.

AWS CLI

스택에 탄력적 IP 주소 등록

다음 예시에서는 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 Registering Elastic IP Addresses with a Stack 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RegisterElasticIp 섹션을 참조하세요.

`register-rds-db-instance`

다음 코드 예시에서는 `register-rds-db-instance`의 사용 방법을 보여줍니다.

AWS CLI

스택에 Amazon RDS 인스턴스 등록

다음 예시에서는 Amazon 리소스 이름(ARN)으로 식별되는 Amazon RDS DB 인스턴스를 지정된 스택에 등록합니다. 인스턴스의 마스터 사용자 이름과 암호도 지정합니다. 단, AWS OpsWorks는 이러한 값을 검증하지 않습니다. 둘 중 하나가 올바르지 않으면 애플리케이션이 데이터베이스에 연결되지 않습니다.

```
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 Registering Amazon RDS Instances with a Stack 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RegisterRdsDbInstance 섹션을 참조하세요.

register-volume

다음 코드 예시에서는 `register-volume`의 사용 방법을 보여줍니다.

AWS CLI

스택에 Amazon EBS 볼륨 등록

다음 예시에서는 볼륨 ID로 식별되는 Amazon EBS 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 Registering Amazon EBS Volumes with a Stack 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RegisterVolume 섹션을 참조하세요.

register

다음 코드 예시에서는 `register`의 사용 방법을 보여줍니다.

AWS CLI

스택에 인스턴스 등록

다음 예시에서는 AWS Opsworks 외부에서 생성된 스택에 인스턴스를 등록하는 다양한 방법을 보여줍니다. 등록할 인스턴스 또는 별도의 워크스테이션에서 `register`를 실행할 수 있습니다. 자세한 내용은 AWS OpsWorks 사용 설명서의 Registering Amazon EC2 and On-premises Instances 섹션을 참조하세요.

참고 : 간결성을 위해 예시에서는 `region` 인수를 생략합니다.

Amazon RDS 인스턴스 등록

EC2 인스턴스를 등록하고 있음을 표시하려면 --infrastructure-class 인수를 ec2로 설정합니다.

다음 예시에서는 별도의 워크스테이션에서 지정된 스택으로 EC2 인스턴스를 등록합니다. 인스턴스는 EC2 ID i-12345678로 식별됩니다. 이 예시에서는 워크스테이션의 기본 SSH 사용자 이름을 사용하고 기본 프라이빗 SSH 키와 같이 암호가 필요하지 않은 인증 기술을 사용하여 인스턴스에 로그인하려고 시도합니다. 실패하면 register에서 암호를 쿼리합니다.

```
aws opsworks register --infrastructure-class=ec2 --stack-id 935450cc-61e0-4b03-
a3e0-160ac817d2bb i-12345678
```

다음 예시에서는 별도의 워크스테이션에서 지정된 스택을 등록합니다. --ssh-username 및 --ssh-private-key 인수를 사용하여 명령이 인스턴스에 로그인하는 데 사용하는 SSH 사용자 이름과 프라이빗 키 파일을 명시적으로 지정합니다. ec2-user는 Amazon Linux 인스턴스의 표준 사용자 이름입니다. Ubuntu 인스턴스에 ubuntu를 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 예시에서는 register 명령을 실행하는 EC2 인스턴스를 등록합니다. SSH를 사용하여 인스턴스에 로그인하고 인스턴스 ID 또는 호스트 이름 대신 --local 인수로 register를 실행합니다.

```
aws opsworks register --infrastructure-class ec2 --stack-id 935450cc-61e0-4b03-
a3e0-160ac817d2bb --local
```

온프레미스 인스턴스 등록

온프레미스 인스턴스를 등록하고 있음을 표시하려면 --infrastructure-class 인수를 on-premises로 설정합니다.

다음 예시에서는 별도의 워크스테이션의 지정된 스택으로 기존 온프레미스 인스턴스를 등록합니다. 인스턴스는 IP 주소 192.0.2.3으로 식별됩니다. 이 예시에서는 워크스테이션의 기본 SSH 사용자 이름을 사용하고 기본 프라이빗 SSH 키와 같이 암호가 필요하지 않은 인증 기술을 사용하여 인스턴스에 로그인하려고 시도합니다. 실패하면 register에서 암호를 쿼리합니다.

```
aws opsworks register --infrastructure-class on-premises --stack-
id 935450cc-61e0-4b03-a3e0-160ac817d2bb 192.0.2.3
```

다음 예시에서는 별도의 워크스테이션의 지정된 스택으로 온프레미스 인스턴스를 등록합니다. 인스턴스는 호스트 이름 host1로 식별됩니다. --override-... 인수는 AWS OpsWorks에 호스트 이름으로 webserver1, 인스턴스의 퍼블릭 및 프라이빗 IP 주소로 192.0.2.3 및 10.0.0.2를 각각 표시하도록 지시합니다.

```
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 예시에서는 별도의 워크스테이션의 지정된 스택으로 온프레미스 인스턴스를 등록합니다. 인스턴스는 IP 주소로 식별됩니다. register는 지정된 SSH 사용자 이름과 프라이빗 키 파일을 사용하여 인스턴스에 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 예시에서는 별도의 워크스테이션의 지정된 스택으로 기존 온프레미스 인스턴스를 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 예시에서는 register 명령을 실행하는 온프레미스 인스턴스를 등록합니다. SSH를 사용하여 인스턴스에 로그인하고 인스턴스 ID 또는 호스트 이름 대신 --local 인수로 register를 실행합니다.

```
aws opsworks register --infrastructure-class on-premises --stack-
id 935450cc-61e0-4b03-a3e0-160ac817d2bb --local
```

출력: 다음은 EC2 인스턴스를 등록하기 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 Registering an Instance with an AWS OpsWorks Stack 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [Register](#) 섹션을 참조하세요.

set-load-based-auto-scaling

다음 코드 예시에서는 `set-load-based-auto-scaling`의 사용 방법을 보여줍니다.

AWS CLI

계층에 대한 로드 기반 스케일링 설정

다음 예시에서는 지정된 계층에 대한 로드 기반 스케일링을 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 예시에서는 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 예시에서는 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 Using Automatic Load-based Scaling 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 SetLoadBasedAutoScaling 섹션을 참조하세요.

`set-permission`

다음 코드 예시에서는 `set-permission`의 사용 방법을 보여줍니다.

AWS CLI

스택당 AWS OpsWorks 권한 수준을 부여하는 방법

`create-user-profile`을 직접적으로 호출하여 AWS Identity and Access Management(IAM) 사용자를 AWS OpsWorks로 가져올 때 사용자는 연결된 IAM 정책에서 부여한 권한만 가집니다. 사용자의 정책을 수정하여 AWS OpsWorks 권한을 부여할 수 있습니다. 그러나 사용자를 가져온 다음 `set-permission` 명령을 사용하여 사용자에게 액세스가 필요한 각 스택의 표준 권한 수준 중 하나를 부여하는 것이 더 쉬운 경우가 많습니다.

다음 예시에서는 Amazon 리소스 이름(ARN)으로 식별되는 사용자에 대해 지정된 스택에 대한 권한을 부여합니다. 이 예시에서는 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 Granting AWS OpsWorks Users Per-Stack Permissions 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 SetPermission 섹션을 참조하세요.

`set-time-based-auto-scaling`

다음 코드 예시에서는 `set-time-based-auto-scaling`의 사용 방법을 보여줍니다.

AWS CLI

계층에 대한 시간 기반 크기 조정 구성을 설정하는 방법

다음 예시에서는 지정된 인스턴스의 시간 기반 조정 구성을 설명합니다. 먼저 `create-instance`를 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 예시에서는 라는 작업 디렉터리의 별도의 파일에 일정을 넣습니다`schedule.json`. 이 예시에서는 인스턴스가 월요일과 화요일의 정오 UTC(협정 세계시) 전후로 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 Using Automatic Time-based Scaling 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 SetTimeBasedAutoScaling 섹션을 참조하세요.

`start-instance`

다음 코드 예시에서는 `start-instance`의 사용 방법을 보여줍니다.

AWS CLI

인스턴스 시작

다음 `start-instance` 명령은 지정된 24/7 인스턴스를 시작합니다.

```
aws opsworks start-instance --instance-id f705ee48-9000-4890-8bd3-20eb05825aaf
```

출력 : 없음. describe-instances를 사용하여 인스턴스의 상태를 확인합니다.

팁 start-stack을 직접적으로 호출하여 하나의 명령으로 스택의 모든 오프라인 인스턴스를 시작할 수 있습니다.

추가 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 Manually Starting, Stopping, and Rebooting 24/7 Instances 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartInstance 섹션을 참조하세요.

start-stack

다음 코드 예시에서는 start-stack의 사용 방법을 보여줍니다.

AWS CLI

스택의 인스턴스 시작

다음 예시에서는 스택의 24/7 인스턴스를 모두 시작합니다. 특정 인스턴스를 시작하는 방법 start-instance를 사용합니다.

```
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 Starting an Instance 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartStack 섹션을 참조하세요.

stop-instance

다음 코드 예시에서는 stop-instance의 사용 방법을 보여줍니다.

AWS CLI

인스턴스 중지

다음 예시에서는 지정된 인스턴스를 중지합니다. 이 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
```

start-instance를 직접적으로 호출하여 중지된 인스턴스를 다시 시작하거나 delete-instance를 직접적으로 호출하여 인스턴스를 삭제할 수 있습니다.

출력 : 없음.

추가 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 Stopping an Instance 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StopInstance 섹션을 참조하세요.

stop-stack

다음 코드 예시에서는 stop-stack의 사용 방법을 보여줍니다.

AWS CLI

스택의 인스턴스를 중지하는 방법

다음 예시에서는 스택의 24/7 인스턴스를 모두 중지합니다. 특정 인스턴스를 중지하는 방법 stop-instance를 사용합니다.

```
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 Stopping an Instance 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StopStack 섹션을 참조하세요.

unassign-instance

다음 코드 예시에서는 unassign-instance의 사용 방법을 보여줍니다.

AWS CLI

계층에서 등록된 인스턴스 할당 해제

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.

추가 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 Unassigning a Registered Instance 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UnassignInstance 섹션을 참조하세요.

unassign-volume

다음 코드 예시에서는 unassign-volume의 사용 방법을 보여줍니다.

AWS CLI

인스턴스에서 볼륨 할당을 취소하는 방법

다음 예시에서는 인스턴스에서 등록된 Amazon Elastic Block Store(Amazon EBS) 볼륨을 할당 취소합니다. 볼륨은 Amazon Elastic Compute Cloud(Amazon EC2) 볼륨 ID가 아닌 스택에 볼륨을 등록할 때 AWS OpsWorks가 할당하는 GUID인 볼륨 ID로 식별됩니다.

```
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 Unassigning Amazon EBS Volumes 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UnassignVolume 섹션을 참조하세요.

update-app

다음 코드 예시에서는 update-app의 사용 방법을 보여줍니다.

AWS CLI

앱을 업데이트하는 방법

다음 예시에서는 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 Editing Apps 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateApp 섹션을 참조하세요.

`update-elastic-ip`

다음 코드 예시에서는 `update-elastic-ip`의 사용 방법을 보여줍니다.

AWS CLI

탄력적 IP 주소 이름을 업데이트하는 방법

다음 예시에서는 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 Resource Management 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateElasticIp 섹션을 참조하세요.

`update-instance`

다음 코드 예시에서는 `update-instance`의 사용 방법을 보여줍니다.

AWS CLI

인스턴스를 업데이트하는 방법

다음 예시에서는 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.

추가 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 Editing the Instance Configuration 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateInstance 섹션을 참조하세요.

update-layer

다음 코드 예시에서는 update-layer의 사용 방법을 보여줍니다.

AWS CLI

계층 업데이트

다음 예시에서는 Amazon EBS 최적화 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 Editing an OpsWorks Layer's Configuration 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateLayer 섹션을 참조하세요.

update-my-user-profile

다음 코드 예시에서는 update-my-user-profile의 사용 방법을 보여줍니다.

AWS CLI

사용자의 프로파일 업데이트

다음 예시에서는 지정된 SSH 퍼블릭 키를 사용하도록 development 사용자 프로파일을 업데이트합니다. 사용자의 AWS 자격 증명은 credentials 파일(~\.aws\credentials)의 development 프로파일로 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 설명서의 Editing AWS OpsWorks User Settings 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateMyUserProfile 섹션을 참조하세요.

update-rds-db-instance

다음 코드 예시에서는 update-rds-db-instance의 사용 방법을 보여줍니다.

AWS CLI

등록된 Amazon RDS DB 인스턴스를 업데이트하는 방법

다음 예시에서는 Amazon RDS 인스턴스의 마스터 암호 값을 업데이트합니다. 단, 이 명령은 RDS 인스턴스의 마스터 암호를 변경하지 않고 AWS OpsWorks에 제공하는 암호만 변경합니다. 이 암호가 RDS 인스턴스의 암호와 일치하지 않으면 애플리케이션이 데이터베이스에 연결되지 않습니다.

```
aws opsworks --region us-east-1 update-rds-db-instance --db-password 123456789
```

출력 : 없음.

추가 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 Registering Amazon RDS Instances with a Stack 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateRdsDbInstance 섹션을 참조하세요.

update-volume

다음 코드 예시에서는 update-volume의 사용 방법을 보여줍니다.

AWS CLI

등록된 볼륨을 업데이트하는 방법

다음 예시에서는 인스턴스에서 등록된 Amazon Elastic Block Store(Amazon EBS) 볼륨을 할당 취소합니다. 볼륨은 Amazon Elastic Compute Cloud(Amazon EC2) 볼륨 ID가 아닌 스택에 등록할 때 AWS OpsWorks가 볼륨에 할당하는 GUID인 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 Assigning Amazon EBS Volumes to an Instance 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateVolume 섹션을 참조하세요.

AWS CLI를 사용한 AWS OpsWorks CM 예시

다음 코드 예시에서는 AWS OpsWorks CM에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

associate-node

다음 코드 예시에서는 associate-node의 사용 방법을 보여줍니다.

AWS CLI

노드 연결

다음 associate-node 명령은 automate-06이라는 Chef Automate 서버에 i-44de882p라는 노드를 연결합니다. 즉, associate-node 명령에 의해 노드에 설치된 chef-client 에이전트 소프트웨어를 통해 automate-06 서버가 노드를 관리하고 노드에 레시피 명령을 전달합니다. 유효한 노드 이름은 EC2 인스턴스 ID입니다.

```
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 사용자 안내서의 AWS OpsWorks for Chef Automate에서 노드 자동 추가를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AssociateNode를 참조하세요.

create-backup

다음 코드 예시에서는 create-backup의 사용 방법을 보여줍니다.

AWS CLI

백업 생성

다음 create-backup 명령은 us-east-1 리전에 automate-06이라는 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 사용자 안내서의 AWS OpsWorks for Chef Automate 백업 및 복원을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateBackup을 참조하세요.

`create-server`

다음 코드 예시에서는 `create-server`의 사용 방법을 보여줍니다.

AWS CLI

서버 생성

다음 `create-server` 예시에서는 기본 리전`automate-06`에 이름이 지정된 새 Chef Automate 서버를 생성합니다. 기본값은 유지할 백업 수, 유지 관리 및 백업 시작 시간과 같은 대부분의 다른 설정에 사용됩니다. `create-server` 명령을 실행하기 전에 AWS OpsWorks for Chef Automate 사용자 안내서의 AWS OpsWorks for Chef Automate 시작하기에서 사전 조건을 완료합니다.

```
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"
```

출력에는 새 서버에 대한 다음과 유사한 정보가 표시됩니다.

```
{
    "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 AWS OpsWorks for Chef Automate API 참조의 UpdateServer를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateServer를 참조하세요.

delete-backup

다음 코드 예시에서는 delete-backup의 사용 방법을 보여줍니다.

AWS CLI

백업 삭제

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"
```

출력에는 백업 삭제 성공 여부가 표시됩니다.

추가 정보

자세한 내용은 AWS OpsWorks 사용자 안내서의 AWS OpsWorks for Chef Automate 백업 및 복원을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteBackup을 참조하세요.

`delete-server`

다음 코드 예시에서는 `delete-server`의 사용 방법을 보여줍니다.

AWS CLI

서버 삭제

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"
```

출력에는 서버 삭제 성공 여부가 표시됩니다.

추가 정보

자세한 내용은 AWS OpsWorks 사용자 안내서의 AWS OpsWorks for Chef Automate 삭제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteServer를 참조하세요.

`describe-account-attributes`

다음 코드 예시에서는 `describe-account-attributes`의 사용 방법을 보여줍니다.

AWS CLI

계정 속성 설명

다음 `describe-account-attributes` 명령은 계정에서 AWS OpsWorks for Chef Automate 리소스를 사용한 정보를 반환합니다.

```
aws opsworks-cm describe-account-attributes
```

명령에서 반환되는 각 계정 속성 항목의 출력은 다음과 유사합니다. 출력:

```
{
 "Attributes": [
    {
       "Maximum": 5,
```

```
        "Name": "ServerLimit",
        "Used": 2
    }
 ]
}
```

추가 정보

자세한 내용은 AWS OpsWorks for Chef Automate API 참조의 DescribeAccountAttributes를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeAccountAttributes를 참조하세요.

describe-backups

다음 코드 예시에서는 describe-backups의 사용 방법을 보여줍니다.

AWS CLI

백업 설명

다음 describe-backups 명령은 기본 리전의 계정에 연결된 모든 백업의 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 사용자 안내서의 AWS OpsWorks for Chef Automate 백업 및 복원을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeBackups를 참조하세요.

describe-events

다음 코드 예시에서는 describe-events의 사용 방법을 보여줍니다.

AWS CLI

이벤트 설명

다음 describe-events 예시에서는 지정된 Chef Automate 서버에 연결된 모든 이벤트의 정보를 반환합니다.

```
aws opsworks-cm describe-events \
    --server-name 'automate-06'
```

명령에서 반환되는 각 이벤트 항목의 출력은 다음 예시와 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 사용자 안내서의 일반 문제 해결 팁을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeEvents를 참조하세요.

describe-node-association-status

다음 코드 예시에서는 describe-node-association-status의 사용 방법을 보여줍니다.

AWS CLI

노드 연결 상태 설명

다음 describe-node-association-status 명령은 노드를 automate-06이라는 Chef Automate 서버에 연결하라는 요청의 상태를 반환합니다.

```
aws opsworks-cm describe-node-association-status --server-
name "automate-06" --node-association-status-token "AflJKl+/
GoKLZJBdDQEx0O65CDi57blQe9nKM8joSok0pQ9xr8DqApBN9/1O6sLdSvlfDEKkEx+eoCHvjoWHaOs="
```

명령에서 반환되는 각 계정 속성 항목의 출력은 다음과 유사합니다. 출력:

```
{
 "NodeAssociationStatus": "IN_PROGRESS"
}
```

추가 정보

자세한 내용은 AWS OpsWorks for Chef Automate API 참조의 DescribeNodeAssociationStatus를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeNodeAssociationStatus를 참조하세요.

`describe-servers`

다음 코드 예시에서는 `describe-servers`의 사용 방법을 보여줍니다.

AWS CLI

서버 설명

다음 `describe-servers` 명령은 계정에 연결되고 기본 리전에 있는 모든 서버의 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 AWS OpsWorks for Chef Automate API 안내서의 DescribeServers를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeServers를 참조하세요.

disassociate-node

다음 코드 예시에서는 disassociate-node의 사용 방법을 보여줍니다.

AWS CLI

노드 연결 해제

다음 disassociate-node 명령은 automate-06라는 Chef Automate 서버의 관리에서 i-44de882p라는 노드를 제거하여 해당 노드의 연결을 해제합니다. 유효한 노드 이름은 EC2 인스턴스 ID입니다.

```
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 사용자 안내서의 AWS OpsWorks for Chef Automate 삭제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DisassociateNode](#)를 참조하세요.

restore-server

다음 코드 예시에서는 `restore-server`의 사용 방법을 보여줍니다.

AWS CLI

서버 복원

다음 `restore-server` 명령은 ID가 `automate-06-2016-11-22T16:13:27.998Z`인 백업에서 기본 리전에 `automate-06`이라는 Chef Automate 서버의 인플레이스 복원을 수행합니다. 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 ID뿐입니다. 출력:

```
(None)
```

추가 정보

자세한 내용은 AWS OpsWorks 사용자 안내서의 실패한 AWS OpsWorks for Chef Automate 복원을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [RestoreServer](#)를 참조하세요.

start-maintenance

다음 코드 예시에서는 `start-maintenance`의 사용 방법을 보여줍니다.

AWS CLI

유지 관리 시작

다음 `start-maintenance` 예시에서는 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 사용자 안내서의 시스템 유지 관리(Puppet Enterprise 서버) 또는 시스템 유지 관리(Chef Automate 서버)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartMaintenance를 참조하세요.

update-server-engine-attributes

다음 코드 예시에서는 update-server-engine-attributes의 사용 방법을 보여줍니다.

AWS CLI

서버 엔진 속성 업데이트

다음 update-server-engine-attributes 명령은 automate-06이라는 Chef Automate 서버의 CHEF_PIVOTAL_KEY 엔진 속성 값을 업데이트합니다. 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"
```

출력에는 업데이트된 서버에 대한 다음과 유사한 정보가 표시됩니다.

```
{
    "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 AWS OpsWorks for Chef Automate API 참조의 UpdateServerEngineAttributes를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateServerEngineAttributes를 참조하세요.

update-server

다음 코드 예시에서는 `update-server`의 사용 방법을 보여줍니다.

AWS CLI

서버 업데이트

다음 `update-server` 명령은 기본 리전에서 지정된 Chef Automate 서버의 유지 관리 시작 시간을 업데이트합니다. `--preferred-maintenance-window` 파라미터가 추가되어 서버 유지 관리의 시작 날짜와 시간을 UTC 기준 월요일 오전 9시 15분으로 변경합니다.

```
aws opsworks-cm update-server \
    --server-name "automate-06" \
    --preferred-maintenance-window "Mon:09:15"
```

출력에는 업데이트된 서버에 대한 다음과 유사한 정보가 표시됩니다.

```
{
    "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 AWS OpsWorks for Chef Automate API 참조의 UpdateServer를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateServer를 참조하세요.

AWS CLI를 사용한 Organizations 예시

다음 코드 예시는 Organizations와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

accept-handshake

다음 코드 예시에서는 accept-handshake 코드를 사용하는 방법을 보여줍니다.

AWS CLI

다른 계정에서 핸드셰이크를 수락하는 방법

조직의 소유자인 Bill은 이전에 Juan의 계정을 초대하여 조직에 가입했습니다. 다음 예시에서는 악수를 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"
        }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 AcceptHandshake 섹션을 참조하세요.

attach-policy

다음 코드 예시에서는 attach-policy 코드를 사용하는 방법을 보여줍니다.

AWS CLI

정책을 루트, OU 또는 계정에 연결

예시 1

다음 예시에서는 서비스 제어 정책(SCP)을 OU에 연결하는 방법을 보여줍니다.

```
aws organizations attach-policy
                --policy-id p-examplepolicyid111
                --target-id ou-examplerootid111-exampleouid111
```

예시 2

다음 예시에서는 계정에 서비스 제어 정책을 직접 연결하는 방법을 보여줍니다.

```
aws organizations attach-policy
```

```
                    --policy-id p-examplepolicyid111
                    --target-id 333333333333
```

- API 세부 정보는 AWS CLI 명령 참조의 AttachPolicy를 참조하세요.

cancel-handshake

다음 코드 예시에서는 cancel-handshake 코드를 사용하는 방법을 보여줍니다.

AWS CLI

다른 계정에서 전송된 핸드셰이크를 취소하는 방법

Bill은 이전에 자신의 조직에 가입하도록 Susan의 계정에 초대장을 보냈습니다. 마음이 바뀌어 Susan이 초대를 수락하기 전에 초대를 취소하기로 결정합니다. 다음 예시에서는 Bill의 취소를 보여줍니다.

```
aws organizations cancel-handshake --handshake-id h-examplehandshakeid111
```

출력에는 현재 CANCELED 상태임을 보여주는 핸드셰이크 객체가 포함됩니다.

```
{
        "Handshake": {
                "Id": "h-examplehandshakeid111",
                "State":"CANCELED",
                "Action": "INVITE",
                "Arn": "arn:aws:organizations::111111111111:handshake/o-
exampleorgid/invite/h-examplehandshakeid111",
                "Parties": [
                        {
                                "Id": "o-exampleorgid",
                                "Type": "ORGANIZATION"
                        },
                        {
                                "Id": "susan@example.com",
                                "Type": "EMAIL"
                        }
                ],
                "Resources": [
                        {
                                "Type": "ORGANIZATION",
                                "Value": "o-exampleorgid",
```

```
                                "Resources": [
                                        {
                                                "Type": "MASTER_EMAIL",
                                                "Value": "bill@example.com"
                                        },
                                        {
                                                "Type": "MASTER_NAME",
                                                "Value": "Master Account"
                                        },
                                        {
                                                "Type": "ORGANIZATION_FEATURE_SET",
                                                "Value": "CONSOLIDATED_BILLING"
                                        }
                                ]
                        },
                        {
                                "Type": "EMAIL",
                                "Value": "anika@example.com"
                        },
                        {
                                "Type": "NOTES",
                                "Value": "This is a request for Susan's account to
 join Bob's organization."
                        }
                ],
                "RequestedTimestamp": 1.47008383521E9,
                "ExpirationTimestamp": 1.47137983521E9
        }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [CancelHandshake](https://docs.aws.amazon.com/goto/aws-cli/organizations-2016-11-28/CancelHandshake) 섹션을 참조하세요.

`create-account`

다음 코드 예시에서는 `create-account` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

자동으로 조직에 가입되는 멤버 계정 생성

다음 예시에서는 조직 내에 멤버 계정을 생성하는 방법을 보여줍니다. 멤버 계정은 Production Account라는 이름과 susan@example.com이라는 이메일 주소로 구성됩니다. roleName 파라미터가 지정되지 않았으므로 Organizations는 OrganizationAccountAccessRole의 기본 이름을 사용하

여 IAM 역할을 자동으로 생성합니다. 또한 IamUserAccessToBilling 파라미터가 지정되지 않았으므로 IAM 사용자 또는 역할에 계정 결제 데이터에 액세스할 수 있는 충분한 권한을 허용하는 설정은 기본값인 ALLOW로 설정됩니다. Organizations는 Susan에게 'Welcome to 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 describe-create-account-status 명령에 대한 Id 응답 값을 create-account-request-id 파라미터의 값으로 제공하여 요청의 현재 상태를 쿼리할 수 있습니다.

자세한 내용은 AWS Organizations 사용자 안내서의 조직 내 AWS 계정 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateAccount를 참조하세요.

`create-organization`

다음 코드 예시에서는 `create-organization` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

예시 1: 새 조직 생성

Bill은 111111111111 계정의 자격 증명을 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 조직의 세부 정보가 있는 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 조직 생성

다음 예시에서는 통합 결제 기능만 지원하는 조직을 만듭니다.

```
aws organizations create-organization --feature-set CONSOLIDATED_BILLING
```

출력에는 새 조직의 세부 정보가 있는 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 사용자 안내서의 조직 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateOrganization을 참조하세요.

`create-organizational-unit`

다음 코드 예시에서는 `create-organizational-unit` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

루트 또는 상위 OU에 OU 생성

다음 예시에서는 AccountingOU라는 OU를 생성하는 방법을 보여줍니다.

```
aws organizations create-organizational-unit --parent-id r-examplerootid111 --
name AccountingOU
```

출력에는 새 OU의 세부 정보가 있는 organizationalUnit 객체가 포함됩니다.

```
{
        "OrganizationalUnit": {
                "Id": "ou-examplerootid111-exampleouid111",
                "Arn": "arn:aws:organizations::111111111111:ou/o-exampleorgid/ou-
examplerootid111-exampleouid111",
                "Name": "AccountingOU"
        }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateOrganizationalUnit을 참조하세요.

`create-policy`

다음 코드 예시에서는 `create-policy` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

예시 1: JSON 정책의 텍스트 소스 파일을 사용하여 정책 생성

다음 예시에서는 이름이 `AllowAllS3Actions`인 서비스 제어 정책(SCP)을 생성하는 방법을 보여줍니다. 정책 콘텐츠는 `policy.json`이라는 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 정책의 세부 정보가 있는 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"
                }
        }
}
```

예시 2: JSON 정책을 파라미터로 사용하여 정책 생성

다음 예시에서는 동일한 SCP를 생성하는 방법을 보여줍니다. 이번에는 정책 콘텐츠를 파라미터에 JSON 문자열로 임베딩합니다. 파라미터에서 문자열을 큰따옴표로 묶은 리터럴로 취급하려면 큰따옴표 앞에 백슬래시를 추가하여 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 사용자 안내서의 조직 정책 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreatePolicy를 참조하세요.

decline-handshake

다음 코드 예시에서는 decline-handshake 코드를 사용하는 방법을 보여줍니다.

AWS CLI

다른 계정에서 전송된 핸드셰이크를 거부하는 방법

다음 예시에서는 계정 222222222222의 소유자인 관리자인 Susan이 Bill의 조직에 가입하라는 초대를 거부하는 것을 보여줍니다. DeclineHandshake 작업은 핸드셰이크 객체를 반환하여 상태가 이제 DECLINED임을 표시합니다.

```
aws organizations decline-handshake --handshake-id h-examplehandshakeid111
```

출력에는 DECLINED의 새 상태를 보여주는 핸드셰이크 객체가 포함됩니다.


```
{
        "Handshake": {
                "Id": "h-examplehandshakeid111",
                "State": "DECLINED",
                "Resources": [
                        {
                                "Type": "ORGANIZATION",
                                "Value": "o-exampleorgid",
                                "Resources": [
                                        {
                                                "Type": "MASTER_EMAIL",
                                                "Value": "bill@example.com"
                                        },
                                        {
                                                "Type": "MASTER_NAME",
                                                "Value": "Master Account"
                                        }
                                ]
                        },
                        {
                                "Type": "EMAIL",
                                "Value": "susan@example.com"
                        },
                        {
                                "Type": "NOTES",
                                "Value": "This is an invitation to Susan's account
 to join the Bill's organization."
                        }
                ],
                "Parties": [
                        {
                                "Type": "EMAIL",
                                "Id": "susan@example.com"
                        },
                        {
                                "Type": "ORGANIZATION",
                                "Id": "o-exampleorgid"
                        }
                ],
```

```
                "Action": "INVITE",
                "RequestedTimestamp": 1470684478.687,
                "ExpirationTimestamp": 1471980478.687,
                "Arn": "arn:aws:organizations::111111111111:handshake/o-
exampleorgid/invite/h-examplehandshakeid111"
        }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeclineHandshake 섹션을 참조하세요.

delete-organization

다음 코드 예시에서는 delete-organization 코드를 사용하는 방법을 보여줍니다.

AWS CLI

조직 삭제

다음 예시에서는 조직을 삭제하는 방법을 보여줍니다. 이 작업을 수행하려면 조직의 마스터 계정 관리자여야 합니다. 이 예시에서는 이전에 조직에서 모든 멤버 계정, OU 및 정책을 제거했다고 가정합니다.

```
aws organizations delete-organization
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteOrganization을 참조하세요.

delete-organizational-unit

다음 코드 예시에서는 delete-organizational-unit 코드를 사용하는 방법을 보여줍니다.

AWS CLI

OU 삭제

다음 예시에서는 OU를 삭제하는 방법을 보여줍니다. 이 예시에서는 이전에 OU에서 모든 계정과 다른 OU를 제거했다고 가정합니다.

```
aws organizations delete-organizational-unit --organizational-unit-id ou-
examplerootid111-exampleouid111
```

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteOrganizationalUnit](#)을 참조하세요.

delete-policy

다음 코드 예시에서는 `delete-policy` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

정책 삭제

다음 예시에서는 조직에서 정책을 삭제하는 방법을 보여줍니다. 이 예시에서는 이전에 정책을 모든 엔터티에서 분리했다고 가정합니다.

```
aws organizations delete-policy --policy-id p-examplepolicyid111
```

- API 세부 정보는 AWS CLI 명령 참조의 [DeletePolicy](#)를 참조하세요.

describe-account

다음 코드 예시에서는 `describe-account` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

계정에 대한 세부 정보를 가져오는 방법

다음 예시에서는 계정에 대한 세부 정보를 요청하는 방법을 설명합니다.

```
aws organizations describe-account --account-id 555555555555
```

출력에는 계정에 대한 세부 정보가 포함된 계정 객체가 표시됩니다.

```
{
        "Account": {
                "Id": "555555555555",
                "Arn": "arn:aws:organizations::111111111111:account/o-
exampleorgid/555555555555",
                "Name": "Beta account",
                "Email": "anika@example.com",
                "JoinedMethod": "INVITED",
                "JoinedTimeStamp": 1481756563.134,
                "Status": "ACTIVE"
```

```
        }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeAccount 섹션을 참조하세요.

describe-create-account-status

다음 코드 예시에서는 `describe-create-account-status` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

계정 생성 요청에 대한 최신 상태를 확인하는 방법

다음 예시에서는 조직에서 이전 계정 만들기 요청에 대한 최신 상태를 요청하는 방법을 보여줍니다. 지정된 --request-id는 create-account에 대한 원래 호출의 응답에서 가져옵니다. 계정 생성 요청은 상태 필드에 조직이 계정 생성을 성공적으로 완료했음을 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"
  }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeCreateAccountStatus 섹션을 참조하세요.

describe-handshake

다음 코드 예시에서는 `describe-handshake` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

핸드셰이크에 대한 정보 가져오기

다음 예시에서는 핸드셰이크에 대한 세부 정보를 요청하는 방법을 설명합니다. 핸드셰이크 ID는 원래 호출에서 InviteAccountToOrganization으로 연결되거나 ListHandshakesForAccount 또는 ListHandshakesForOrganization으로 연결되는 호출에서 나옵니다.

```
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"
        }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeHandshake 섹션을 참조하세요.

describe-organization

다음 코드 예시에서는 describe-organization 코드를 사용하는 방법을 보여줍니다.

AWS CLI

현재 조직에 대한 정보를 가져오는 방법

다음 예시에서는 조직의 계정 목록을 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"
```

```
                    }
                ]
        }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeOrganization](https://docs.aws.amazon.com/goto/aws-cli/organizations-2016-11-28/DescribeOrganization) 섹션을 참조하세요.

`describe-organizational-unit`

다음 코드 예시에서는 `describe-organizational-unit` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

OU에 대한 정보 가져오기

다음 `describe-organizational-unit` 예시에서는 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeOrganizationalUnit](https://docs.aws.amazon.com/goto/aws-cli/organizations-2016-11-28/DescribeOrganizationalUnit) 섹션을 참조하세요.

`describe-policy`

다음 코드 예시에서는 `describe-policy` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

정책 정보 가져오기

다음 예시에서는 정책 정보를 요청하는 방법을 보여줍니다.

```
aws organizations describe-policy --policy-id p-examplepolicyid111
```

출력에는 정책의 세부 정보가 있는 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"
                }
        }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribePolicy를 참조하세요.

detach-policy

다음 코드 예시에서는 detach-policy 코드를 사용하는 방법을 보여줍니다.

AWS CLI

정책을 루트, OU 또는 계정에서 분리

다음 예시에서는 OU에서 정책을 분리하는 방법을 보여줍니다.

```
aws organizations  detach-policy  --target-id ou-examplerootid111-exampleouid111 --
policy-id p-examplepolicyid111
```

- API 세부 정보는 AWS CLI 명령 참조의 DetachPolicy를 참조하세요.

disable-policy-type

다음 코드 예시에서는 disable-policy-type 코드를 사용하는 방법을 보여줍니다.

AWS CLI

루트에서 정책 유형을 비활성화하는 방법

다음 예시에서는 루트에서 서비스 제어 정책(SCP) 정책 유형을 비활성화하는 방법을 보여줍니다.

```
aws organizations disable-policy-type --root-id r-examplerootid111 --policy-type SERVICE_CONTROL_POLICY
```

출력에는 PolicyTypes 응답 요소에 더 이상 SERVICE_CONTROL_POLICY가 포함되어 있지 않음을 보여줍니다.

```
{
        "Root": {
                "PolicyTypes": [],
                "Name": "Root",
                "Id": "r-examplerootid111",
                "Arn": "arn:aws:organizations::111111111111:root/o-exampleorgid/r-examplerootid111"
        }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DisablePolicyType 섹션을 참조하세요.

`enable-all-features`

다음 코드 예시에서는 `enable-all-features` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

조직의 모든 기능 활성화

이 예시에서는 관리자가 조직의 모든 초대된 계정에 조직의 활성화된 모든 기능을 승인하도록 요청하는 것을 보여줍니다. AWS 조직은 초대된 모든 멤버 계정에 등록된 주소로 이메일을 보내 소유자가 전송된 핸드셰이크를 수락하여 모든 기능의 변경을 승인하도록 요청합니다. 초대된 모든 멤버 계정이 핸드셰이크를 수락하면 조직 관리자는 모든 기능에 대한 변경을 완료할 수 있으며, 적절한 권한이 있는 계정은 정책을 생성하고 루트, OU 및 계정에 적용할 수 있습니다.

```
aws organizations enable-all-features
```

출력은 승인을 위해 초대된 모든 멤버 계정으로 전송되는 핸드셰이크 객체가 있습니다.

```
{
        "Handshake": {
                "Action": "ENABLE_ALL_FEATURES",
                "Arn":"arn:aws:organizations::111111111111:handshake/o-exampleorgid/
enable_all_features/h-examplehandshakeid111",
                "ExpirationTimestamp":1.483127868609E9,
                "Id":"h-examplehandshakeid111",
                "Parties": [
                        {
                                "id":"o-exampleorgid",
                                "type":"ORGANIZATION"
                        }
                ],
                "requestedTimestamp":1.481831868609E9,
                "resources": [
                        {
                                "type":"ORGANIZATION",
                                "value":"o-exampleorgid"
                        }
                ],
                "state":"REQUESTED"
        }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 EnableAllFeatures 섹션을 참조하세요.

enable-policy-type

다음 코드 예시에서는 enable-policy-type 코드를 사용하는 방법을 보여줍니다.

AWS CLI

루트에서 정책 유형 사용을 활성화하는 방법

다음 예시에서는 루트에서 서비스 제어 정책(SCP) 정책 유형을 활성화하는 방법을 보여줍니다.

```
aws organizations enable-policy-type --root-id r-examplerootid111 --policy-
type SERVICE_CONTROL_POLICY
```

출력에는 SCP가 이제 활성화되었음을 나타내는 policyTypes 응답 요소가 있는 루트 객체가 표시됩니다.

```
{
        "Root": {
                "PolicyTypes": [
                        {
                                "Status":"ENABLED",
                                "Type":"SERVICE_CONTROL_POLICY"
                        }
                ],
                "Id": "r-examplerootid111",
                "Name": "Root",
                "Arn": "arn:aws:organizations::111111111111:root/o-exampleorgid/r-
examplerootid111"
        }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [EnablePolicyType](https://docs.aws.amazon.com/goto/aws-cli/organizations-2016-11-28/EnablePolicyType) 섹션을 참조하세요.

invite-account-to-organization

다음 코드 예시에서는 invite-account-to-organization 코드를 사용하는 방법을 보여줍니다.

AWS CLI

조직에 가입하도록 계정 초대

다음 예시에서는 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 전송된 내용을 보여주는 핸드셰이크 구조가 포함되어 있습니다.

```
{
        "Handshake": {
                "Action": "INVITE",
                "Arn": "arn:aws:organizations::111111111111:handshake/o-
exampleorgid/invite/h-examplehandshakeid111",
                "ExpirationTimestamp": 1482952459.257,
                "Id": "h-examplehandshakeid111",
                "Parties": [
```

```
                {
                        "Id": "o-exampleorgid",
                        "Type": "ORGANIZATION"
                },
                {
                        "Id": "juan@example.com",
                        "Type": "EMAIL"
                }
        ],
        "RequestedTimestamp": 1481656459.257,
        "Resources": [
                {
                        "Resources": [
                                {
                                        "Type": "MASTER_EMAIL",
                                        "Value": "bill@amazon.com"
                                },
                                {
                                        "Type": "MASTER_NAME",
                                        "Value": "Org Master Account"
                                },
                                {
                                        "Type": "ORGANIZATION_FEATURE_SET",
                                        "Value": "FULL"
                                }
                        ],
                        "Type": "ORGANIZATION",
                        "Value": "o-exampleorgid"
                },
                {
                        "Type": "EMAIL",
                        "Value": "juan@example.com"
                }
        ],
        "State": "OPEN"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 InviteAccountToOrganization 섹션을 참조하세요.

leave-organization

다음 코드 예시에서는 `leave-organization` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

멤버 계정으로 조직에서 나가기

다음 예시에서는 현재 멤버인 조직을 떠나도록 요청하는 멤버 계정의 관리자를 보여줍니다.

```
aws organizations leave-organization
```

- API 세부 정보는 AWS CLI 명령 참조의 LeaveOrganization 섹션을 참조하세요.

list-accounts-for-parent

다음 코드 예시에서는 `list-accounts-for-parent` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

지정된 상위 루트 또는 OU의 모든 계정 목록을 검색하는 방법

다음 예시에서는 OU의 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"
                }
        ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListAccountsForParent 섹션을 참조하세요.

list-accounts

다음 코드 예시에서는 `list-accounts` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

조직의 모든 계정 목록 가져오기

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"
                }
        ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListAccounts를 참조하세요.

list-children

다음 코드 예시에서는 `list-children` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

상위 OUs 또는 루트의 하위 계정 및 OU를 검색하는 방법

다음 예시에서는 해당 계정 444444444444이 포함된 루트 또는 OU를 나열하는 방법을 보여줍니다.

```
aws organizations list-children --child-type ORGANIZATIONAL_UNIT --parent-id ou-
examplerootid111-exampleouid111
```

출력에는 부모에 포함된 두 개의 자식 OU가 표시됩니다.

```
{
```

```
        "Children": [
                {
                        "Id": "ou-examplerootid111-exampleouid111",
                        "Type":"ORGANIZATIONAL_UNIT"
                },
                {
                        "Id":"ou-examplerootid111-exampleouid222",
                        "Type":"ORGANIZATIONAL_UNIT"
                }
        ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListChildren 섹션을 참조하세요.

`list-create-account-status`

다음 코드 예시에서는 `list-create-account-status` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

예시 1: 현재 조직에서 수행된 계정 생성 요청 목록을 검색하는 방법

다음 예시에서는 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"
                }
        ]
}
```

예시 2: 현재 조직에서 진행 중인 계정 생성 요청 목록을 검색하는 방법

다음 예시에서는 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"
                }
        ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListCreateAccountStatus 섹션을 참조하세요.

list-handshakes-for-account

다음 코드 예시에서는 `list-handshakes-for-account` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

계정으로 전송된 핸드셰이크 목록을 검색하는 방법

다음 예시에서는 작업을 호출하는 데 사용된 자격 증명의 계정과 연결된 모든 핸드셰이크 목록을 가져오는 방법을 보여줍니다.

```
aws organizations list-handshakes-for-account
```

출력에는 현재 상태를 포함한 각 핸드셰이크에 대한 정보와 함께 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"
        }
}
```


- API 세부 정보는 AWS CLI 명령 참조의 [ListHandshakesForAccount](#) 섹션을 참조하세요.

`list-handshakes-for-organization`

다음 코드 예시에서는 `list-handshakes-for-organization` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

조직과 연결된 핸드셰이크 목록을 검색하는 방법

다음 예시에서는 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 OPEN 상태를 보여줍니다. 두 번째는 Anika의 계정에 대한 초대이며 수락됨 상태를 보여줍니다.

```
{
        "Handshakes": [
                {
                        "Action": "INVITE",
                        "Arn": "arn:aws:organizations::111111111111:handshake/o-
exampleorgid/invite/h-examplehandshakeid111",
                        "ExpirationTimestamp": 1482952459.257,
                        "Id": "h-examplehandshakeid111",
                        "Parties": [
                                {
                                        "Id": "o-exampleorgid",
                                        "Type": "ORGANIZATION"
                                },
                                {
                                        "Id": "juan@example.com",
                                        "Type": "EMAIL"
                                }
                        ],
                        "RequestedTimestamp": 1481656459.257,
                        "Resources": [
                                {
                                        "Resources": [
                                                {
                                                        "Type": "MASTER_EMAIL",
                                                        "Value": "bill@amazon.com"
                                                },
                                                {
```

```
                                                        "Type": "MASTER_NAME",
                                                        "Value": "Org Master
 Account"
                                                },
                                                {
                                                        "Type":
 "ORGANIZATION_FEATURE_SET",
                                                        "Value": "FULL"
                                                }
                                        ],
                                        "Type": "ORGANIZATION",
                                        "Value": "o-exampleorgid"
                                },
                                {
                                        "Type": "EMAIL",
                                        "Value": "juan@example.com"
                                },
                                {
                                        "Type":"NOTES",
                                        "Value":"This is an invitation to Juan's
 account to join Bill's organization."
                                }
                        ],
                        "State": "OPEN"
                },
                {
                        "Action": "INVITE",
                        "State":"ACCEPTED",
                        "Arn": "arn:aws:organizations::111111111111:handshake/o-
exampleorgid/invite/h-examplehandshakeid111",
                        "ExpirationTimestamp": 1.471797437427E9,
                        "Id": "h-examplehandshakeid222",
                        "Parties": [
                                {
                                        "Id": "o-exampleorgid",
                                        "Type": "ORGANIZATION"
                                },
                                {
                                        "Id": "anika@example.com",
                                        "Type": "EMAIL"
                                }
                        ],
                        "RequestedTimestamp": 1.469205437427E9,
                        "Resources": [
```

```
                                {
                                        "Resources": [
                                                {
                                                        "Type":"MASTER_EMAIL",
                                                        "Value":"bill@example.com"
                                                },
                                                {
                                                        "Type":"MASTER_NAME",
                                                        "Value":"Master Account"
                                                }
                                        ],
                                        "Type":"ORGANIZATION",
                                        "Value":"o-exampleorgid"
                                },
                                {
                                        "Type":"EMAIL",
                                        "Value":"anika@example.com"
                                },
                                {
                                        "Type":"NOTES",
                                        "Value":"This is an invitation to Anika's
 account to join Bill's organization."
                                }
                        ]
                }
        ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListHandshakesForOrganization 섹션을 참조하세요.

`list-organizational-units-for-parent`

다음 코드 예시에서는 `list-organizational-units-for-parent` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

상위 OUs 또는 루트에서 OU 목록을 검색하는 방법

다음 예시에서는 지정된 루트에서 OU 목록을 가져오는 방법을 보여줍니다.

```
aws organizations list-organizational-units-for-parent --parent-id r-
examplerootid111
```

출력은 지정된 루트에 두 개의 OU가 포함되어 있음과 각각의 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"
                }
        ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListOrganizationalUnitsForParent 섹션을 참조하세요.

`list-parents`

다음 코드 예시에서는 `list-parents` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

계정 또는 하위 OUs의 상위 OU 또는 루트를 나열하는 방법

다음 예시에서는 해당 계정 444444444444이 포함된 루트 또는 상위 OU를 나열하는 방법을 보여줍니다.

```
aws organizations list-parents --child-id 444444444444
```

출력에는 지정된 계정이 지정된 ID를 가진 OU에 있음을 보여줍니다.

```
{
  "Parents": [
        {
```

```
            "Id": "ou-examplerootid111-exampleouid111",
            "Type": "ORGANIZATIONAL_UNIT"
          }
      ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListParents 섹션을 참조하세요.

list-policies-for-target

다음 코드 예시에서는 `list-policies-for-target` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

계정에 직접 연결된 SCPs 목록을 검색하는 방법

다음 예시에서는 필터 파라미터에 지정된 대로 계정에 직접 연결된 모든 서비스 제어 정책(SCP) 목록을 가져오는 방법을 보여줍니다.

```
aws organizations list-policies-for-target --filter SERVICE_CONTROL_POLICY --target-id 444444444444
```

출력에는 정책에 대한 요약 정보와 함께 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."
                }
        ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListPoliciesForTarget 섹션을 참조하세요.

list-policies

다음 코드 예시에서는 `list-policies` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

특정 유형의 조직에 있는 모든 정책 목록 가져오기

다음 예시에서는 필터 파라미터로 지정된 SCP 목록을 가져오는 방법을 보여줍니다.

```
aws organizations list-policies --filter SERVICE_CONTROL_POLICY
```

출력에는 요약 정보가 있는 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"
```

```
                }
        ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListPolicies를 참조하세요.

list-roots

다음 코드 예시에서는 `list-roots` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

조직의 루트 목록 검색

이 예시에서는 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"
                                }
                        ]
                }
        ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListRoots 섹션을 참조하세요.

list-targets-for-policy

다음 코드 예시에서는 `list-targets-for-policy` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

정책이 첨부된 루트, OU 및 계정 목록 검색

다음 예시에서는 지정된 정책이 연결된 루트, OU 및 계정 목록을 가져오는 방법을 보여줍니다.

```
aws organizations list-targets-for-policy --policy-id p-FullAWSAccess
```

출력에는 정책이 첨부된 루트, OU 및 계정에 대한 요약 정보와 함께 첨부 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"
                }
        ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListTargetsForPolicy 섹션을 참조하세요.

move-account

다음 코드 예시에서는 `move-account` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

루트 또는 OUs 간에 계정을 이동하는 방법

다음 예시에서는 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
```

- API 세부 정보는 AWS CLI 명령 참조의 MoveAccount 섹션을 참조하세요.

`remove-account-from-organization`

다음 코드 예시에서는 `remove-account-from-organization` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

조직에서 마스터 계정으로 계정을 제거하는 방법

다음 예시에서는 조직의 계정 목록을 요청하는 방법을 보여줍니다.

```
aws organizations remove-account-from-organization --account-id 333333333333
```

- API 세부 정보는 AWS CLI 명령 참조의 RemoveAccountFromOrganization 섹션을 참조하세요.

`update-organizational-unit`

다음 코드 예시에서는 `update-organizational-unit` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

OU의 이름 바꾸기

이 예시에서는 OU의 이름을 바꾸는 방법을 보여줍니다. 이 예시에서는 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""
        }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateOrganizationalUnit 섹션을 참조하세요.

update-policy

다음 코드 예시에서는 update-policy 코드를 사용하는 방법을 보여줍니다.

AWS CLI

예시 1: 정책 이름 바꾸기

다음 update-policy 예시에서는 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"
        }
```

```
    }
}
```

예시 2: 정책의 JSON 텍스트 콘텐츠를 바꾸려면

다음 예시에서는 이전 예시에서 SCP의 JSON 텍스트를 EC2 대신 S3를 허용하는 새 JSON 정책 텍스트 문자열로 바꾸는 방법을 보여줍니다.

```
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"
        }
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdatePolicy 섹션을 참조하세요.

AWS CLI를 사용한 AWS Outposts 예시

다음 코드 예시에서는 AWS Outposts에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

get-outpost-instance-types

다음 코드 예시에서는 get-outpost-instance-types의 사용 방법을 보여줍니다.

AWS CLI

Outpost에서 인스턴스 유형 가져오기

다음 get-outpost-instance-types 예시에서는 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 AWS Outposts 사용자 안내서의 Outpost에서 인스턴스 시작을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetOutpostInstanceTypes를 참조하세요.

get-outpost

다음 코드 예시에서는 get-outpost의 사용 방법을 보여줍니다.

AWS CLI

Outpost 세부 정보 가져오기

다음 get-outpost 예시에서는 지정된 Outpost의 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 AWS Outposts 사용자 안내서의 Outposts 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetOutpost를 참조하세요.

list-outposts

다음 코드 예시에서는 list-outposts의 사용 방법을 보여줍니다.

AWS CLI

Outposts 나열

다음 `list-outposts` 예시에서는 AWS 계정의 Outposts를 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 AWS Outposts 사용자 안내서의 [Outposts 작업](#)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListOutposts를 참조하세요.

`list-sites`

다음 코드 예시에서는 `list-sites`의 사용 방법을 보여줍니다.

AWS CLI

사이트 나열

다음 `list-sites` 예시에서는 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 AWS Outposts 사용자 안내서의 Outposts 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListSites를 참조하세요.

AWS CLI를 사용한 AWS Payment Cryptography 예시

다음 코드 예시에서는 AWS Payment Cryptography에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

create-alias

다음 코드 예시에서는 create-alias의 사용 방법을 보여줍니다.

AWS CLI

키 별칭 생성

다음 create-alias 예시에서는 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 사용자 안내서의 별칭 정보를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateAlias를 참조하세요.

create-key

다음 코드 예시에서는 create-key의 사용 방법을 보여줍니다.

AWS CLI

키 생성

다음 `create-key` 예시에서는 CVV/CVV2 값을 생성하고 확인하는 데 사용할 수 있는 2KEY 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 사용자 안내서의 [키 생성]을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateKey를 참조하세요.

delete-alias

다음 코드 예시에서는 delete-alias의 사용 방법을 보여줍니다.

AWS CLI

별칭 삭제

다음 delete-alias 예시에서는 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 사용자 안내서의 별칭 정보를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteAlias를 참조하세요.

delete-key

다음 코드 예시에서는 delete-key의 사용 방법을 보여줍니다.

AWS CLI

키 삭제

다음 delete-key 예시에서는 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 사용자 안내서의 키 삭제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteKey를 참조하세요.

export-key

다음 코드 예시에서는 export-key의 사용 방법을 보여줍니다.

AWS CLI

키 내보내기

다음 export-key 예시에서는 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":
 "308205A106092A864886F70D010702A08205923082058E020101310D300B06096086480165030402013082031F
        "WrappedKeyMaterialFormat": "TR34_KEY_BLOCK"
    }
}
```


자세한 내용은 AWS Payment Cryptography 사용자 안내서의 키 내보내기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ExportKey를 참조하세요.

get-alias

다음 코드 예시에서는 get-alias의 사용 방법을 보여줍니다.

AWS CLI

별칭 가져오기

다음 get-alias 예시에서는 별칭에 연결된 키의 ARN을 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 사용자 안내서의 별칭 정보를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetAlias를 참조하세요.

get-key

다음 코드 예시에서는 get-key의 사용 방법을 보여줍니다.

AWS CLI

키의 메타데이터 가져오기

다음 get-key 예시에서는 별칭에 연결된 키의 메타데이터를 반환합니다. 이 작업은 암호화 구성 요소를 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 사용자 안내서의 키 가져오기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetKey를 참조하세요.

get-parameters-for-export

다음 코드 예시에서는 get-parameters-for-export의 사용 방법을 보여줍니다.

AWS CLI

내보내기 프로세스 초기화

다음 get-parameters-for-export 예시에서는 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 사용자 안내서의 키 내보내기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetParametersForExport를 참조하세요.

`get-parameters-for-import`

다음 코드 예시에서는 `get-parameters-for-import`의 사용 방법을 보여줍니다.

AWS CLI

가져오기 프로세스 초기화

다음 get-parameters-for-import 예시에서는 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 사용자 안내서의 키 가져오기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetParametersForImport를 참조하세요.

get-public-key-certificate

다음 코드 예시에서는 get-public-key-certificate의 사용 방법을 보여줍니다.

AWS CLI

퍼블릭 키 반환

다음 get-public-key-certificate 예시에서는 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 사용자 안내서의 키 페어에 연결된 퍼블릭 키/인증서 가져오기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetPublicKeyCertificate를 참조하세요.

`import-key`

다음 코드 예시에서는 `import-key`의 사용 방법을 보여줍니다.

AWS CLI

TR-34 키 가져오기

다음 `import-key` 예시에서는 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:

```
3082059806092A864886F70D010702A08205893082058502010131OD300B0609608648016503040201308203160
```

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 사용자 안내서의 키 가져오기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ImportKey를 참조하세요.

list-aliases

다음 코드 예시에서는 list-aliases의 사용 방법을 보여줍니다.

AWS CLI

별칭 목록 가져오기

다음 `list-aliases` 예시에서는 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 사용자 안내서의 별칭 정보를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListAliases를 참조하세요.

`list-keys`

다음 코드 예시에서는 `list-keys`의 사용 방법을 보여줍니다.

AWS CLI

키 목록 가져오기

다음 `list-keys` 예시에서는 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 사용자 안내서의 키 나열을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListKeys를 참조하세요.

`list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`의 사용 방법을 보여줍니다.

AWS CLI

키의 태그 목록 가져오기

다음 `list-tags-for-resource` 예시에서는 키의 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 사용자 안내서의 API 작업을 사용한 키 태그 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource를 참조하세요.

`restore-key`

다음 코드 예시에서는 `restore-key`의 사용 방법을 보여줍니다.

AWS CLI

삭제 예약된 키 복원

다음 `restore-key` 예시에서는 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 사용자 안내서의 키 삭제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RestoreKey를 참조하세요.

start-key-usage

다음 코드 예시에서는 `start-key-usage`의 사용 방법을 보여줍니다.

AWS CLI

키 활성화

다음 `start-key-usage` 예시에서는 사용할 키를 활성화합니다.

```
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 사용자 안내서의 키 활성화 및 비활성화를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartKeyUsage를 참조하세요.

stop-key-usage

다음 코드 예시에서는 stop-key-usage의 사용 방법을 보여줍니다.

AWS CLI

키 비활성화

다음 stop-key-usage 예시에서는 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 사용자 안내서의 키 활성화 및 비활성화를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StopKeyUsage를 참조하세요.

tag-resource

다음 코드 예시에서는 tag-resource의 사용 방법을 보여줍니다.

AWS CLI

키에 태그 지정

다음 tag-resource 예시에서는 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 사용자 안내서의 키 태그 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

untag-resource

다음 코드 예시에서는 untag-resource의 사용 방법을 보여줍니다.

AWS CLI

키에서 태그 제거

다음 `untag-resource` 예시에서는 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 사용자 안내서의 키 태그 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

update-alias

다음 코드 예시에서는 `update-alias`의 사용 방법을 보여줍니다.

AWS CLI

별칭 업데이트

다음 `update-alias` 예시에서는 별칭을 다른 키에 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 사용자 안내서의 별칭 정보를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateAlias를 참조하세요.

AWS CLI를 사용한 AWS Payment Cryptography 데이터 플레인 예시

다음 코드 예시는 AWS Payment Cryptography 데이터 플레인과 함께 AWS Command Line Interface 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

작업

decrypt-data

다음 코드 예시에서는 decrypt-data의 사용 방법을 보여줍니다.

AWS CLI

사이퍼텍스트 복호화

다음 decrypt-data 예시에서는 대칭 키를 사용하여 사이퍼텍스트 데이터를 복호화합니다. 이 작업을 수행하려면 키의 KeyModesOfUse을 Decrypt로 설정하고 KeyUsage를 TR31_D0_SYMMETRIC_DATA_ENCRYPTION_KEY로 설정해야 합니다.

```
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 사용자 안내서의 데이터 복호화를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DecryptData를 참조하세요.

encrypt-data

다음 코드 예시에서는 encrypt-data의 사용 방법을 보여줍니다.

AWS CLI

데이터 암호화

다음 encrypt-data 예시에서는 대칭 키를 사용하여 일반 텍스트 데이터를 암호화합니다. 이 작업을 수행하려면 키의 KeyModesOfUse을 Encrypt로 설정하고 KeyUsage를 TR31_D0_SYMMETRIC_DATA_ENCRYPTION_KEY로 설정해야 합니다.

```
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 사용자 안내서의 데이터 암호화를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 EncryptData를 참조하세요.

generate-card-validation-data

다음 코드 예시에서는 generate-card-validation-data의 사용 방법을 보여줍니다.

AWS CLI

CVV 생성

다음 `generate-card-validation-data` 예시에서는 CVV/CVV2를 생성합니다.

```
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 사용자 안내서의 카드 데이터 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GenerateCardValidationData를 참조하세요.

generate-mac

다음 코드 예시에서는 `generate-mac`의 사용 방법을 보여줍니다.

AWS CLI

MAC 생성

다음 `generate-card-validation-data` 예시에서는 HMAC_SHA256 알고리즘과 HMAC 암호화 키를 사용하여 카드 데이터 인증을 위한 해시 기반 메시지 인증 코드(HMAC)를 생성합니다. 키의 `KeyModesOfUse`을 `TR31_M7_HMAC_KEY`로 설정하고 `KeyUsage`를 `Generate`로 설정해야 합니다.

```
aws payment-cryptography-data generate-mac \
    --key-identifier arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h \
```

```
    --message-
data "3b313038383439303031303733393431353d32343038323236303030373030303f33" \
    --generation-attributes Algorithm=HMAC_SHA256
```

출력:

```
{
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h,
    "KeyCheckValue": "2976E7",
    "Mac": "ED87F26E961C6D0DDB78DA5038AA2BDDEA0DCE03E5B5E96BDDD494F4A7AA470C"
}
```

자세한 내용은 AWS Payment Cryptography 사용자 안내서의 MAC 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GenerateMac을 참조하세요.

generate-pin-data

다음 코드 예시에서는 generate-pin-data의 사용 방법을 보여줍니다.

AWS CLI

PIN 생성

다음 generate-card-validation-data 예시에서는 Visa PIN 체계를 사용하여 새 무작위 PIN을 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 사용자 안내서의 PIN 데이터 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GeneratePinData를 참조하세요.

re-encrypt-data

다음 코드 예시에서는 re-encrypt-data의 사용 방법을 보여줍니다.

AWS CLI

다른 키로 데이터 다시 암호화

다음 re-encrypt-data 예시에서는 AES 대칭 키를 사용하여 암호화된 사이퍼텍스트를 복호화하고 Derived Unique Key Per Transaction(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 사용자 안내서의 데이터 암호화 및 복호화를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ReEncryptData를 참조하세요.

translate-pin-data

다음 코드 예시에서는 `translate-pin-data`의 사용 방법을 보여줍니다.

AWS CLI

PIN 데이터 변환

다음 `translate-pin-data` 예시에서는 ISO 0 PIN 블록을 사용하는 PEK TDES 암호화의 PIN을 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 사용자 안내서의 PIN 데이터 변환을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TranslatePinData를 참조하세요.

verify-auth-request-cryptogram

다음 코드 예시에서는 verify-auth-request-cryptogram의 사용 방법을 보여줍니다.

AWS CLI

승인 요청 확인

다음 verify-auth-request-cryptogram 예시에서는 승인 요청 암호문(ARQC)을 확인합니다.

```
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 사용자 안내서의 승인 요청 암호문(ARQC) 확인을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 VerifyAuthRequestCryptogram을 참조하세요.

verify-card-validation-data

다음 코드 예시에서는 verify-card-validation-data의 사용 방법을 보여줍니다.

AWS CLI

CVV 확인

다음 `verify-card-validation-data` 예시에서는 PAN에 대한 CVV/CVV2를 확인합니다.

```
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 사용자 안내서의 카드 데이터 확인을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 VerifyCardValidationData를 참조하세요.

verify-mac

다음 코드 예시에서는 `verify-mac`의 사용 방법을 보여줍니다.

AWS CLI

MAC 확인

다음 `verify-mac` 예시에서는 HMAC_SHA256 알고리즘과 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 "3b343038383439303031303733393431353d32343038323236303030373030303f33" \
```

```
    --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 사용자 안내서의 MAC 확인을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 VerifyMac을 참조하세요.

verify-pin-data

다음 코드 예시에서는 verify-pin-data의 사용 방법을 보여줍니다.

AWS CLI

PIN 확인

다음 verify-pin-data 예시에서는 PAN의 PIN을 확인합니다.

```
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 사용자 안내서의 PIN 데이터 확인을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 VerifyPinData를 참조하세요.

AWS CLI를 사용한 Amazon Pinpoint 예시

다음 코드 예시는 Amazon Pinpoint와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`create-app`

다음 코드 예시에서는 `create-app`의 사용 방법을 보여줍니다.

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"
        }
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateApp을 참조하세요.

`create-sms-template`

다음 코드 예시에서는 `create-sms-template`의 사용 방법을 보여줍니다.

AWS CLI

SMS 채널을 통해 전송되는 메시지의 메시지 템플릿 생성

다음 `create-sms-template` 예시에서는 SMS 메시지 템플릿을 생성합니다.

```
aws pinpoint create-sms-template \
    --template-name TestTemplate \
    --sms-template-request file://myfile.json \
    --region us-east-1
```

`myfile.json`의 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 사용자 안내서의 Amazon Pinpoint SMS 채널을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateSmsTemplate을 참조하세요.

`delete-app`

다음 코드 예시에서는 `delete-app`의 사용 방법을 보여줍니다.

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": {}
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteApp을 참조하세요.

get-apns-channel

다음 코드 예시에서는 get-apns-channel의 사용 방법을 보여줍니다.

AWS CLI

애플리케이션의 APN 채널 상태 및 설정 정보 가져오기

다음 get-apns-channel 예시에서는 애플리케이션의 APN 채널 상태 및 설정에 대한 정보를 가져옵니다.

```
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
        }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetApnsChannel을 참조하세요.

get-app

다음 코드 예시에서는 get-app의 사용 방법을 보여줍니다.

AWS CLI

애플리케이션(프로젝트)의 정보 가져오기

다음 get-app 예시에서는 애플리케이션(프로젝트)의 정보를 가져옵니다.

```
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"
            }
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetApp을 참조하세요.

get-apps

다음 코드 예시에서는 get-apps의 사용 방법을 보여줍니다.

AWS CLI

모든 애플리케이션의 정보 가져오기

다음 get-apps 예시에서는 모든 애플리케이션(프로젝트)의 정보를 가져옵니다.

```
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 직접적으로 호출하고 해당 값을 NextToken 입력 파라미터로 입력합니다.

- API 세부 정보는 AWS CLI 명령 참조의 GetApps를 참조하세요.

get-campaign

다음 코드 예시에서는 get-campaign의 사용 방법을 보여줍니다.

AWS CLI

캠페인의 상태, 구성 및 기타 설정 정보 가져오기

다음 get-campaign 예시에서는 캠페인의 상태, 구성 및 기타 설정에 대한 정보를 가져옵니다.

```
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
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetCampaign을 참조하세요.

get-campaigns

다음 코드 예시에서는 get-campaigns의 사용 방법을 보여줍니다.

AWS CLI

애플리케이션에 연결된 모든 캠페인의 상태, 구성 및 기타 설정 정보 가져오기

다음 get-campaigns 예시에서는 애플리케이션에 연결된 모든 캠페인의 상태, 구성 및 기타 설정에 대한 정보를 가져옵니다.

```
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
            }
        ]
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetCampaigns를 참조하세요.

get-channels

다음 코드 예시에서는 get-channels의 사용 방법을 보여줍니다.

AWS CLI

애플리케이션의 각 채널 기록 및 상태 정보 가져오기

다음 get-channels 예시에서는 애플리케이션의 각 채널 기록 및 상태에 대한 정보를 가져옵니다.

```
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
            }
        }
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetChannels를 참조하세요.

`get-email-channel`

다음 코드 예시에서는 `get-email-channel`의 사용 방법을 보여줍니다.

AWS CLI

애플리케이션의 Email 채널 상태 및 설정 정보 가져오기

다음 `get-email-channel` 예시에서는 애플리케이션의 Email 채널 상태 및 설정을 가져옵니다.

```
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
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetEmailChannel을 참조합니다.

get-endpoint

다음 코드 예시에서는 get-endpoint의 사용 방법을 보여줍니다.

AWS CLI

애플리케이션의 특정 엔드포인트 설정 및 속성 정보 가져오기

다음 get-endpoint 예시에서는 애플리케이션의 특정 엔드포인트 설정 및 속성에 대한 정보를 가져옵니다.

```
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"
            }
        }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetEndpoint를 참조하세요.

get-gcm-channel

다음 코드 예시에서는 get-gcm-channel의 사용 방법을 보여줍니다.

AWS CLI

애플리케이션의 GCM 채널 상태 및 설정 정보 가져오기

다음 get-gcm-channel 예시에서는 애플리케이션의 GCM 채널 상태 및 설정에 대한 정보를 가져옵니다.

```
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
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetGcmChannel을 참조하세요.

get-sms-channel

다음 코드 예시에서는 get-sms-channel의 사용 방법을 보여줍니다.

AWS CLI

애플리케이션의 SMS 채널 상태 및 설정 정보 가져오기

다음 get-sms-channel 예시에서는 애플리케이션의 SMS 채널 상태 및 설정을 가져옵니다.

```
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
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetSmsChannel을 참조하세요.

get-sms-template

다음 코드 예시에서는 get-sms-template의 사용 방법을 보여줍니다.

AWS CLI

SMS 채널을 통해 전송되는 메시지에 대한 메시지 템플릿의 콘텐츠 및 설정 가져오기

다음 get-sms-template 예시에서는 SMS 메시지 템플릿의 콘텐츠와 설정을 가져옵니다.

```
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 사용자 안내서의 Amazon Pinpoint SMS 채널을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetSmsTemplate을 참조하세요.

get-voice-channel

다음 코드 예시에서는 get-voice-channel의 사용 방법을 보여줍니다.

AWS CLI

애플리케이션의 Voice 채널 상태 및 설정 정보 가져오기

다음 get-voice-channel 예시에서는 애플리케이션의 Voice 채널 상태 및 설정을 가져옵니다.

```
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
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetVoiceChannel을 참조합니다.

list-tags-for-resource

다음 코드 예시에서는 list-tags-for-resource의 사용 방법을 보여줍니다.

AWS CLI

리소스의 태그 목록 가져오기

다음 list-tags-for-resource 예시에서는 지정된 리소스에 연결된 모든 태그(키 이름 및 값)를 가져옵니다.

```
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 리소스 태그 지정'<https://docs.aws.amazon.com/pinpoint/latest/developerguide/tagging-resources.html>을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource를 참조하세요.

phone-number-validate

다음 코드 예시에서는 phone-number-validate의 사용 방법을 보여줍니다.

AWS CLI

전화번호 정보 가져오기

다음 phone-number-validate 예시에서는 전화번호 정보를 가져옵니다.

```
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 사용자 안내서의 Amazon Pinpoint SMS 채널을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PhoneNumberValidate를 참조하세요.

send-messages

다음 코드 예시에서는 send-messages의 사용 방법을 보여줍니다.

AWS CLI

애플리케이션의 엔드포인트를 사용하여 SMS 메시지 전송

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.json의 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 사용자 안내서의 Amazon Pinpoint SMS 채널을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 SendMessages를 참조하세요.

send-users-messages

다음 코드 예시에서는 send-users-messages의 사용 방법을 보여줍니다.

AWS CLI

애플리케이션 사용자에게 SMS 메시지 전송

다음 send-users-messages 예시에서는 애플리케이션 사용자에게 다이렉트 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 사용자 안내서의 Amazon Pinpoint SMS 채널을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 SendUsersMessages를 참조하세요.

`tag-resource`

다음 코드 예시에서는 `tag-resource`의 사용 방법을 보여줍니다.

AWS CLI

리소스에 태그 추가

다음 예시에서는 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 리소스 태그 지정'<https://docs.aws.amazon.com/pinpoint/latest/developerguide/tagging-resources.html>을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

untag-resource

다음 코드 예시에서는 `untag-resource`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 리소스에서 태그 제거

다음 `untag-resource` 예시에서는 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.

예시 2: 리소스에서 여러 태그 제거

다음 `untag-resource` 예시에서는 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 태그 지정'<https://docs.aws.amazon.com/pinpoint/latest/developerguide/tagging-resources.html>을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

update-sms-channel

다음 코드 예시에서는 update-sms-channel의 사용 방법을 보여줍니다.

AWS CLI

SMS 채널 활성화 또는 애플리케이션의 SMS 채널 상태 및 설정 업데이트

다음 update-sms-channel 예시에서는 애플리케이션의 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 사용자 안내서의 Amazon Pinpoint SMS 채널을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateSmsChannel을 참조하세요.

AWS CLI를 사용한 Amazon Polly 예시

다음 코드 예시는 Amazon Polly와 함께 AWS Command Line Interface를 사용하여 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

delete-lexicon

다음 코드 예시에서는 delete-lexicon의 사용 방법을 보여줍니다.

AWS CLI

어휘 삭제

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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteLexicon을 참조하세요.

get-lexicon

다음 코드 예시에서는 get-lexicon의 사용 방법을 보여줍니다.

AWS CLI

어휘의 콘텐츠 가져오기

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.

- API 세부 정보는 AWS CLI 명령 참조의 GetLexicon을 참조하세요.

get-speech-synthesis-task

다음 코드 예시에서는 get-speech-synthesis-task의 사용 방법을 보여줍니다.

AWS CLI

음성 합성 작업 정보 가져오기

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/amzn-s3-demo-
bucket/70b61c0f-57ce-4715-a247-cae8729dcce9.mp3",
        "CreationTime": 1603911042.689,
        "RequestCharacters": 1311,
        "OutputFormat": "mp3",
        "TextType": "text",
        "VoiceId": "Joanna"
    }
}
```

자세한 내용은 Amazon Polly 개발자 안내서의 긴 오디오 파일 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetSpeechSynthesisTask를 참조하세요.

list-lexicons

다음 코드 예시에서는 `list-lexicons`의 사용 방법을 보여줍니다.

AWS CLI

어휘 나열

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.

- API 세부 정보는 AWS CLI 명령 참조의 ListLexicons를 참조하세요.

list-speech-synthesis-tasks

다음 코드 예시에서는 `list-speech-synthesis-tasks`의 사용 방법을 보여줍니다.

AWS CLI

음성 합성 작업 나열

다음 `list-speech-synthesis-tasks` 예시에서는 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/amzn-s3-demo-
bucket/70b61c0f-57ce-4715-a247-cae8729dcce9.mp3",
            "CreationTime": 1603911042.689,
            "RequestCharacters": 1311,
            "OutputFormat": "mp3",
            "TextType": "text",
            "VoiceId": "Joanna"
        }
    ]
}
```

자세한 내용은 Amazon Polly 개발자 안내서의 긴 오디오 파일 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListComplianceSummaries를 참조하세요.

put-lexicon

다음 코드 예시에서는 put-lexicon의 사용 방법을 보여줍니다.

AWS CLI

어휘 저장

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.

- API 세부 정보는 AWS CLI 명령 참조의 PutLexicon을 참조하세요.

start-speech-synthesis-task

다음 코드 예시에서는 start-speech-synthesis-task의 사용 방법을 보여줍니다.

AWS CLI

텍스트 합성

다음 start-speech-synthesis-task 예시에서는 text_file.txt의 텍스트를 합성하고 결과 MP3 파일을 지정된 버킷에 저장합니다.

```
aws polly start-speech-synthesis-task \
    --output-format mp3 \
    --output-s3-bucket-name amzn-s3-demo-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/amzn-s3-demo-
bucket/70b61c0f-57ce-4715-a247-cae8729dcce9.mp3",
        "CreationTime": 1603911042.689,
        "RequestCharacters": 1311,
        "OutputFormat": "mp3",
        "TextType": "text",
        "VoiceId": "Joanna"
    }
}
```

자세한 내용은 Amazon Polly 개발자 안내서의 긴 오디오 파일 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartSpeechSynthesisTask를 참조하세요.

AWS CLI를 사용한 AWS 가격표 예시

다음 코드 예시에서는 AWS 가격표에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

describe-services

다음 코드 예시에서는 describe-services의 사용 방법을 보여줍니다.

AWS CLI

서비스 메타데이터 검색

이 예시에서는 Amazon EC2 서비스 코드의 메타데이터를 가져옵니다.

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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeServices를 참조하세요.

get-attribute-values

다음 코드 예시에서는 get-attribute-values의 사용 방법을 보여줍니다.

AWS CLI

속성 값 목록 가져오기

다음 get-attribute-values 예시에서는 지정된 속성에 사용할 수 있는 값 목록을 가져옵니다.

```
aws pricing get-attribute-values \
    --service-code AmazonEC2 \
    --attribute-name volumeType \
    --max-items 2
```

출력:

```
{
    "NextToken": "eyJOZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAyfQ==",
    "AttributeValues": [
        {
            "Value": "Cold HDD"
        },
        {
            "Value": "General Purpose"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetAttributeValues를 참조하세요.

get-products

다음 코드 예시에서는 get-products의 사용 방법을 보여줍니다.

AWS CLI

제품 목록 가져오기

이 예시에서는 지정된 기준과 일치하는 제품 목록을 가져옵니다.

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\"}"
  ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetProducts를 참조하세요.

AWS CLI를 사용한 AWS Private CA 예시

다음 코드 예시에서는 AWS Private CA에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

작업

`create-certificate-authority-audit-report`

다음 코드 예시는 `create-certificate-authority-audit-report`의 사용 방법을 보여 줍니다.

AWS CLI

인증 기관 감사 보고서를 생성하는 방법

다음 `create-certificate-authority-audit-report` 명령은 ARN에 의해 식별된 프라이빗 CA에 대한 감사 보고서를 생성합니다.

```
aws acm-pca create-certificate-authority-audit-report --certificate-authority-arn arn:aws:acm-pca:us-east-1:accountid:certificate-authority/12345678-1234-1234-1234-123456789012 --s3-bucket-name your-bucket-name --audit-report-response-format JSON
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateCertificateAuthorityAuditReport를 참조하세요.

`create-certificate-authority`

다음 코드 예시는 `create-certificate-authority`의 사용 방법을 보여 줍니다.

AWS CLI

프라이빗 인증 기관을 생성하는 방법

다음 `create-certificate-authority` 명령은 AWS 계정에서 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
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateCertificateAuthority를 참조하세요.

`delete-certificate-authority`

다음 코드 예시는 `delete-certificate-authority`의 사용 방법을 보여 줍니다.

AWS CLI

프라이빗 인증 기관을 삭제하는 방법

다음 `delete-certificate-authority` 명령은 ARN에 의해 식별된 인증 기관을 삭제합니다.

```
aws acm-pca delete-certificate-authority --certificate-authority-arn arn:aws:acm-pca:us-west-2:123456789012:certificate-authority/12345678-1234-1234-1234-123456789012
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteCertificateAuthority를 참조하세요.

describe-certificate-authority-audit-report

다음 코드 예시는 `describe-certificate-authority-audit-report`의 사용 방법을 보여 줍니다.

AWS CLI

인증 기관에 대한 감사 보고서를 설명하는 방법

다음 `describe-certificate-authority-audit-report` 명령은 ARN에 의해 식별된 CA의 지정된 감사 보고서에 대한 정보를 나열합니다.

```
aws acm-pca describe-certificate-authority-audit-report --certificate-authority-arn arn:aws:acm-pca:us-west-2:123456789012:certificate-authority/99999999-8888-7777-6666-555555555555 --audit-report-id 11111111-2222-3333-4444-555555555555
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeCertificateAuthorityAuditReport를 참조하세요.

describe-certificate-authority

다음 코드 예시는 `describe-certificate-authority`의 사용 방법을 보여 줍니다.

AWS CLI

프라이빗 인증 기관을 설명하는 방법

다음 `describe-certificate-authority` 명령은 ARN에 의해 식별된 프라이빗 CA에 대한 정보를 나열합니다.

```
aws acm-pca describe-certificate-authority --certificate-authority-arn arn:aws:acm-pca:us-west-2:123456789012:certificate-authority/12345678-1234-1234-1234-123456789012
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeCertificateAuthority를 참조하세요.

get-certificate-authority-certificate

다음 코드 예시는 get-certificate-authority-certificate의 사용 방법을 보여 줍니다.

AWS CLI

인증 기관(CA) 인증서를 검색하는 방법

다음 get-certificate-authority-certificate 명령은 ARN에 의해 지정된 프라이빗 CA에 대한 인증서 및 인증서 체인을 검색합니다.

```
aws acm-pca get-certificate-authority-certificate --certificate-
authority-arn arn:aws:acm-pca:us-west-2:123456789012:certificate-
authority/12345678-1234-1234-1234-123456789012 --output text
```

- API 세부 정보는 AWS CLI 명령 참조의 GetCertificateAuthorityCertificate를 참조하세요.

get-certificate-authority-csr

다음 코드 예시는 get-certificate-authority-csr의 사용 방법을 보여 줍니다.

AWS CLI

인증 기관에 대한 인증서 서명 요청을 검색하는 방법

다음 get-certificate-authority-csr 명령은 ARN에 의해 지정된 프라이빗 CA에 대한 CSR을 검색합니다.

```
aws acm-pca get-certificate-authority-csr --certificate-
authority-arn arn:aws:acm-pca:us-west-2:123456789012:certificate-
authority/12345678-1234-1234-1234-123456789012 --output text
```

- API 세부 정보는 AWS CLI 명령 참조의 GetCertificateAuthorityCsr을 참조하세요.

get-certificate

다음 코드 예시는 get-certificate의 사용 방법을 보여 줍니다.

AWS CLI

발급된 인증서를 검색하는 방법

다음 `get-certificate` 예제는 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 인증서로 연결되는 인증서 체인입니다. `--output text` 옵션을 사용하면 두 인증서 사이에 TAB 문자가 삽입된다는 점에 유의하세요(들여쓰기되는 텍스트의 원인). 이 출력을 가져와 다른 도구로 인증서를 구문 분석하려는 경우, 올바르게 처리되도록 TAB 문자를 제거해야 할 수 있습니다.

- API 세부 정보는 AWS CLI 명령 참조의 [GetCertificate](https://docs.aws.amazon.com/goto/aws-cli/acm-pca-2017-08-22/GetCertificate)를 참조하세요.

`import-certificate-authority-certificate`

다음 코드 예시는 `import-certificate-authority-certificate`의 사용 방법을 보여 줍니다.

AWS CLI

인증 기관 인증서를 ACM PCA로 가져오는 방법

다음 `import-certificate-authority-certificate` 명령은 ARN에 의해 지정된 CA에 대한 서명된 프라이빗 CA 인증서를 ACM PCA로 가져옵니다.

```
aws acm-pca import-certificate-authority-certificate --certificate-authority-arn arn:aws:acm-pca:us-west-2:123456789012:certificate-authority/12345678-1234-1234-1234-123456789012 --certificate file://C:\ca_cert.pem
 --certificate-chain file://C:\ca_cert_chain.pem
```

- API 세부 정보는 AWS CLI 명령 참조의 ImportCertificateAuthorityCertificate를 참조하세요.

`issue-certificate`

다음 코드 예시는 `issue-certificate`의 사용 방법을 보여 줍니다.

AWS CLI

프라이빗 인증서를 발급하는 방법

다음 `issue-certificate` 명령은 ARN에 의해 지정된 프라이빗 CA를 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
```

- API 세부 정보는 AWS CLI 명령 참조의 IssueCertificate를 참조하세요.

`list-certificate-authorities`

다음 코드 예시는 `list-certificate-authorities`의 사용 방법을 보여 줍니다.

AWS CLI

프라이빗 인증 기관을 나열하는 방법

다음 `list-certificate-authorities` 명령은 계정에서 모든 프라이빗 CA에 대한 정보를 나열합니다.

```
aws acm-pca list-certificate-authorities --max-results 10
```

- API 세부 정보는 AWS CLI 명령 참조의 ListCertificateAuthorities를 참조하세요.

list-tags

다음 코드 예시는 `list-tags`의 사용 방법을 보여 줍니다.

AWS CLI

인증 기관에 대한 태그를 나열하는 방법

다음 `list-tags` 명령은 ARN에 의해 지정된 프라이빗 CA와 연결된 태그를 나열합니다.

```
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
```

- API 세부 정보는 AWS CLI 명령 참조의 ListTags를 참조하세요.

revoke-certificate

다음 코드 예시는 `revoke-certificate`의 사용 방법을 보여 줍니다.

AWS CLI

프라이빗 인증서를 해지하는 방법

다음 `revoke-certificate` 명령은 ARN에 의해 식별된 CA에서 프라이빗 인증서를 해지합니다.

```
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"
```

- API 세부 정보는 AWS CLI 명령 참조의 RevokeCertificate를 참조하세요.

tag-certificate-authority

다음 코드 예시는 `tag-certificate-authority`의 사용 방법을 보여 줍니다.

AWS CLI

프라이빗 인증 기관에 태그를 연결하는 방법

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
```

- API 세부 정보는 AWS CLI 명령 참조의 [TagCertificateAuthority](https://docs.aws.amazon.com/goto/aws-cli/acm-pca-2017-08-22/TagCertificateAuthority)를 참조하세요.

`untag-certificate-authority`

다음 코드 예시는 `untag-certificate-authority`의 사용 방법을 보여 줍니다.

AWS CLI

프라이빗 인증 기관에서 하나 이상의 태그를 제거하는 방법

다음 `untag-certificate-authority` 명령은 ARN에 의해 식별된 프라이빗 CA에서 태그를 제거합니다.

```
aws acm-pca untag-certificate-authority --certificate-authority-
arn arn:aws:acm-pca:us-west-2:123456789012:certificate-
authority/12345678-1234-1234-1234-123456789012 --tags Key=Purpose,Value=Website
```

- API 세부 정보는 AWS CLI 명령 참조의 [UntagCertificateAuthority](https://docs.aws.amazon.com/goto/aws-cli/acm-pca-2017-08-22/UntagCertificateAuthority)를 참조하세요.

`update-certificate-authority`

다음 코드 예시는 `update-certificate-authority`의 사용 방법을 보여 줍니다.

AWS CLI

프라이빗 인증 기관의 구성을 업데이트하는 방법

다음 `update-certificate-authority` 명령은 ARN에 의해 식별된 프라이빗 CA의 상태와 구성을 업데이트합니다.

```
aws acm-pca update-certificate-authority --certificate-
authority-arn arn:aws:acm-pca:us-west-2:123456789012:certificate-
```

```
authority/12345678-1234-1234-1234-1232456789012 --revocation-configuration file://C:
\revoke_config.txt --status "DISABLED"
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateCertificateAuthority을 참조하세요.

AWS CLI를 사용한 AWS Proton 예시

다음 코드 예시에서는 AWS Proton에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`cancel-service-instance-deployment`

다음 코드 예시에서는 `cancel-service-instance-deployment`의 사용 방법을 보여줍니다.

AWS CLI

서비스 인스턴스 배포 취소

다음 `cancel-service-instance-deployment` 예시에서는 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 사용자 안내서의 서비스 인스턴스 업데이트를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CancelServiceInstanceDeployment를 참조하세요.

cancel-service-pipeline-deployment

다음 코드 예시에서는 cancel-service-pipeline-deployment의 사용 방법을 보여줍니다.

AWS CLI

서비스 파이프라인 배포 취소

다음 cancel-service-pipeline-deployment 예시에서는 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 사용자 안내서의 서비스 파이프라인 업데이트를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CancelServicePipelineDeployment를 참조하세요.

`create-service`

다음 코드 예시에서는 `create-service`의 사용 방법을 보여줍니다.

AWS CLI

서비스 생성

다음 `create-service` 예시에서는 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.yaml`의 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 사용자 안내서의 서비스 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateService를 참조하세요.

delete-service

다음 코드 예시에서는 `delete-service`의 사용 방법을 보여줍니다.

AWS CLI

서비스 삭제

다음 `delete-service` 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteService를 참조하세요.

get-service-instance

다음 코드 예시에서는 `get-service-instance`의 사용 방법을 보여줍니다.

AWS CLI

서비스 인스턴스 세부 정보 가져오기

다음 `get-service-instance` 예시에서는 서비스 인스턴스의 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 사용자 안내서의 [서비스 데이터 보기]를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [GetServiceInstance]를 참조하세요.

get-service

다음 코드 예시에서는 `get-service`의 사용 방법을 보여줍니다.

AWS CLI

서비스 세부 정보 가져오기

다음 `get-service` 예시에서는 서비스의 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 사용자 안내서의 서비스 데이터 보기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetService를 참조하세요.

list-service-instances

다음 코드 예시에서는 list-service-instances의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 모든 서비스 인스턴스 나열

다음 `list-service-instances` 예시에서는 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 사용자 안내서의 서비스 인스턴스 데이터 보기를 참조하세요.

예시 2: 지정된 서비스 인스턴스 나열

다음 `get-service-instance` 예시에서는 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 사용자 안내서의 서비스 인스턴스 데이터 보기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListServiceInstances를 참조하세요.

`update-service-instance`

다음 코드 예시에서는 `update-service-instance`의 사용 방법을 보여줍니다.

AWS CLI

서비스 인스턴스를 새 마이너 버전으로 업데이트

다음 `update-service-instance` 예시에서는 서비스 인스턴스를 서비스 템플릿의 새 마이너 버전으로 업데이트하여 'my-other-instance'라는 새 인스턴스를 새로운 필수 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.yaml`의 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 사용자 안내서의 [서비스 인스턴스 업데이트](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateServiceInstance를 참조하세요.

update-service-pipeline

다음 코드 예시에서는 update-service-pipeline의 사용 방법을 보여줍니다.

AWS CLI

서비스 파이프라인 업데이트

다음 update-service-pipeline 예시에서는 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 사용자 안내서의 서비스 파이프라인 업데이트를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateServicePipeline을 참조하세요.

update-service

다음 코드 예시에서는 update-service의 사용 방법을 보여줍니다.

AWS CLI

서비스 업데이트

다음 update-service 예시에서는 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 및 AWS Proton 사용자 안내서의 서비스 편집을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateService를 참조하세요.

AWS CLI를 사용한 QLDB 예시

다음 코드 예시는 QLDB와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

cancel-journal-kinesis-stream

다음 코드 예시에서는 `cancel-journal-kinesis-stream`의 사용 방법을 보여줍니다.

AWS CLI

저널 스트림 취소

다음 `cancel-journal-kinesis-stream` 예시에서는 원장에서 지정된 저널 스트림을 취소합니다.

```
aws qldb cancel-journal-kinesis-stream \
    --ledger-name myExampleLedger \
    --stream-id 7ISCkqwe4y25YyHLzYUFAf
```

출력:

```
{
    "StreamId": "7ISCkqwe4y25YyHLzYUFAf"
}
```

자세한 내용은 Amazon QLDB 개발자 안내서의 Amazon QLDB에서 저널 데이터 스트리밍을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CancelJournalKinesisStream을 참조하세요.

`create-ledger`

다음 코드 예시에서는 `create-ledger`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 기본 속성을 사용하여 원장 생성

다음 `create-ledger` 예시에서는 이름 `myExampleLedger` 및 권한 모드 `STANDARD`를 사용하여 원장을 생성합니다. 삭제 방지를 위한 선택적 파라미터와 AWS KMS 키는 지정되지 않으므로 각각 `true` 및 AWS 소유 KMS 키를 기본값으로 사용합니다.

```
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"
}
```

예시 2: 삭제 방지 특성을 비활성화하고 고객 관리형 KMS 키와 지정된 태그를 사용하여 원장 생성

다음 `create-ledger` 예시에서는 이름 `myExampleLedger2` 및 권한 모드 `STANDARD`를 사용하여 원장을 생성합니다. 삭제 방지 특성이 비활성화되고, 지정된 고객 관리 KMS 키는 저장 중 암호화에 사용되며, 지정된 태그는 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 QLDB 개발자 안내서의 Amazon QLDB 원장의 기본 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateLedger를 참조하세요.

delete-ledger

다음 코드 예시에서는 delete-ledger의 사용 방법을 보여줍니다.

AWS CLI

원장 삭제

다음 delete-ledger 예시에서는 지정된 원장을 삭제합니다.

```
aws qldb delete-ledger \
    --name myExampleLedger
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon QLDB 개발자 안내서의 Amazon QLDB 원장의 기본 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteLedger를 참조하세요.

describe-journal-kinesis-stream

다음 코드 예시에서는 describe-journal-kinesis-stream의 사용 방법을 보여줍니다.

AWS CLI

저널 스트림 설명

다음 `describe-journal-kinesis-stream` 예시에서는 원장에서 지정된 저널 스트림의 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 QLDB 개발자 안내서의 Amazon QLDB에서 저널 데이터 스트리밍을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeJournalKinesisStream을 참조하세요.

describe-journal-s3-export

다음 코드 예시에서는 `describe-journal-s3-export`의 사용 방법을 보여줍니다.

AWS CLI

저널 내보내기 작업 설명

다음 `describe-journal-s3-export` 예시에서는 원장에서 지정된 내보내기 작업의 세부 정보를 표시합니다.

```
aws qldb describe-journal-s3-export \
    --name myExampleLedger \
    --export-id ADR2ONPKN5LINYGb4dp7yZ
```

출력:

```
{
    "ExportDescription": {
        "S3ExportConfiguration": {
            "Bucket": "amzn-s3-demo-bucket",
            "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 QLDB 개발자 안내서의 Amazon QLDB에서 저널 내보내기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeJournalS3Export를 참조하세요.

`describe-ledger`

다음 코드 예시에서는 `describe-ledger`의 사용 방법을 보여줍니다.

AWS CLI

원장 설명

다음 describe-ledger 예시에서는 지정된 원장의 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 QLDB 개발자 안내서의 Amazon QLDB 원장의 기본 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeLedger를 참조하세요.

export-journal-to-s3

다음 코드 예시에서는 export-journal-to-s3의 사용 방법을 보여줍니다.

AWS CLI

저널 블록을 S3으로 내보내기

다음 export-journal-to-s3 예시에서는 이름이 myExampleLedger인 원장에서 지정된 날짜 및 시간 범위 내에 저널 블록의 내보내기 작업을 생성합니다. 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": "amzn-s3-demo-bucket",
    "Prefix": "ledgerexport1/",
    "EncryptionConfiguration": {
        "ObjectEncryptionType": "SSE_S3"
    }
}
```

출력:

```
{
    "ExportId": "ADR2ONPKN5LINYGb4dp7yZ"
}
```

자세한 내용은 Amazon QLDB 개발자 안내서의 Amazon QLDB에서 저널 내보내기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ExportJournalToS3을 참조하세요.

get-block

다음 코드 예시에서는 get-block의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 입력 파일을 사용하여 검증할 저널 블록 및 증명 가져오기

다음 get-block 예시에서는 블록 데이터 객체와 지정된 원장의 증명을 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 QLDB 개발자 안내서의 Amazon QLDB에서 데이터 확인을 참조하세요.

예시 2: 간편 구문을 사용하여 검증할 저널 블록 및 증명 가져오기

다음 `get-block` 예시에서는 간편 구문을 사용하여 지정된 원장에서 블록 데이터 객체와 증명을 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 QLDB 개발자 안내서의 Amazon QLDB에서 데이터 확인을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetBlock을 참조하세요.

get-digest

다음 코드 예시에서는 get-digest의 사용 방법을 보여줍니다.

AWS CLI

원장의 다이제스트 가져오기

다음 get-digest 예시에서는 저널의 가장 최근 커밋 블록에서 지정된 원장의 다이제스트를 요청합니다.

```
aws qldb get-digest \
    --name vehicle-registration
```

출력:

```
{
    "Digest": "6m6BMXobbJKpMhahwVthAEsN6awgnHK62Qq5McGP1Gk=",
    "DigestTipAddress": {
        "IonText": "{strandId:\"KmA3ZZca7vAIiJAK9S5Iwl\",sequenceNo:123}"
    }
}
```

자세한 내용은 Amazon QLDB 개발자 안내서의 Amazon QLDB에서 데이터 확인을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetDigest를 참조하세요.

get-revision

다음 코드 예시에서는 get-revision의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 입력 파일을 사용하여 검증할 문서 개정 및 증명 가져오기

다음 get-revision 예시에서는 지정된 원장의 개정 데이터 객체와 증명을 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 QLDB 개발자 안내서의 [Amazon QLDB에서 데이터 확인]을 참조하세요.

예시 2: 간편 구문을 사용하여 검증할 문서 개정 및 증명 가져오기

다음 `get-revision` 예시에서는 간편 구문을 사용하여 지정된 원장의 개정 데이터 객체와 증명을 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 QLDB 개발자 안내서의 Amazon QLDB에서 데이터 확인을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetRevision을 참조하세요.

list-journal-kinesis-streams-for-ledger

다음 코드 예시에서는 list-journal-kinesis-streams-for-ledger의 사용 방법을 보여줍니다.

AWS CLI

원장의 저널 스트림 나열

다음 list-journal-kinesis-streams-for-ledger 예시에서는 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 QLDB 개발자 안내서의 Amazon QLDB에서 저널 데이터 스트리밍을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListJournalKinesisStreamsForLedger를 참조하세요.

list-journal-s3-exports-for-ledger

다음 코드 예시에서는 `list-journal-s3-exports-for-ledger`의 사용 방법을 보여줍니다.

AWS CLI

원장의 저널 내보내기 작업 나열

다음 `list-journal-s3-exports-for-ledger` 예시에서는 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": "amzn-s3-demo-bucket",
                "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 QLDB 개발자 안내서의 Amazon QLDB에서 저널 내보내기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListJournalS3ExportsForLedger를 참조하세요.

`list-journal-s3-exports`

다음 코드 예시에서는 `list-journal-s3-exports`의 사용 방법을 보여줍니다.

AWS CLI

저널 내보내기 작업 나열

다음 `list-journal-s3-exports` 예시에서는 현재 AWS 계정 및 리전에 연결된 모든 원장의 저널 내보내기 작업을 나열합니다.

```
aws qldb list-journal-s3-exports
```

출력:

```
{
    "JournalS3Exports": [
        {
            "Status": "IN_PROGRESS",
            "LedgerName": "myExampleLedger",
            "S3ExportConfiguration": {
                "EncryptionConfiguration": {
                    "ObjectEncryptionType": "SSE_S3"
                },
```

```
                "Bucket": "amzn-s3-demo-bucket",
                "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": "amzn-s3-demo-bucket",
                "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 QLDB 개발자 안내서의 Amazon QLDB에서 저널 내보내기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListJournalS3Exports를 참조하세요.

list-ledgers

다음 코드 예시에서는 `list-ledgers`의 사용 방법을 보여줍니다.

AWS CLI

사용 가능한 원장 나열

다음 `list-ledgers` 예시에서는 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 QLDB 개발자 안내서의 Amazon QLDB 원장의 기본 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListLedgers를 참조하세요.

list-tags-for-resource

다음 코드 예시에서는 `list-tags-for-resource`의 사용 방법을 보여줍니다.

AWS CLI

원장에 연결된 태그 나열

다음 `list-tags-for-resource` 예시에서는 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 QLDB 개발자 안내서의 Amazon QLDB 리소스 태그 지정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource를 참조하세요.

stream-journal-to-kinesis

다음 코드 예시에서는 `stream-journal-to-kinesis`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 입력 파일을 사용하여 저널 데이터를 Kinesis Data Streams로 스트리밍

다음 `stream-journal-to-kinesis` 예시에서는 이름이 `myExampleLedger`인 원장에서 지정된 날짜 및 시간 범위 내에 저널 데이터 스트림을 생성합니다. 스트림은 지정된 Amazon Kinesis Data Streams로 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.json`의 콘텐츠:

```
{
    "StreamArn": "arn:aws:kinesis:us-east-1:123456789012:stream/stream-for-qldb",
    "AggregationEnabled": true
}
```

출력:

```
{
    "StreamId": "7ISCkqwe4y25YyHLzYUFAf"
}
```

자세한 내용은 Amazon QLDB 개발자 안내서의 Amazon QLDB에서 저널 데이터 스트리밍을 참조하세요.

예시 2: 간편 구문을 사용하여 저널 데이터를 Kinesis Data Streams로 스트리밍

다음 stream-journal-to-kinesis 예시에서는 이름이 myExampleLedger인 원장에서 지정된 날짜 및 시간 범위 내에 저널 데이터 스트림을 생성합니다. 스트림은 지정된 Amazon Kinesis Data Streams로 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 QLDB 개발자 안내서의 Amazon QLDB에서 저널 데이터 스트리밍을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StreamJournalToKinesis를 참조하세요.

tag-resource

다음 코드 예시에서는 tag-resource의 사용 방법을 보여줍니다.

AWS CLI

원장에 태그 지정

다음 tag-resource 예시에서는 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 QLDB 개발자 안내서의 Amazon QLDB 리소스 태그 지정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

untag-resource

다음 코드 예시에서는 untag-resource 코드를 사용하는 방법을 보여줍니다.

AWS CLI

리소스에서 태그 제거

다음 untag-resource 예시에서는 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 QLDB 개발자 안내서의 Amazon QLDB 리소스 태그 지정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

update-ledger-permissions-mode

다음 코드 예시에서는 update-ledger-permissions-mode의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 원장의 권한 모드를 STANDARD로 업데이트

다음 update-ledger-permissions-mode 예시에서는 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"
}
```

예시 2: 원장의 권한 모드를 ALLOW_ALL로 업데이트

다음 `update-ledger-permissions-mode` 예시에서는 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 QLDB 개발자 안내서의 [Amazon QLDB 원장의 기본 작업](#)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [UpdateLedgerPermissionsMode](#)를 참조하세요.

`update-ledger`

다음 코드 예시에서는 `update-ledger`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 원장의 삭제 방지 속성 업데이트

다음 `update-ledger` 예시에서는 지정된 원장을 업데이트하여 삭제 방지 특성을 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"
}
```

예시 2: 원장의 AWS KMS 키를 고객 관리형 키로 업데이트

다음 update-ledger 예시에서는 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"
    }
}
```

예시 3: 원장의 AWS KMS 키를 AWS 소유 키로 업데이트

다음 update-ledger 예시에서는 지정된 원장을 업데이트하여 AWS 소유 KMS 키를 저장 시 암호화에 사용하도록 합니다.

```
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 QLDB 개발자 안내서의 Amazon QLDB 원장의 기본 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateLedger를 참조하세요.

AWS CLI를 사용한 Amazon RDS 예시

다음 코드 예시는 Amazon RDS와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

add-option-to-option-group

다음 코드 예시에서는 `add-option-to-option-group`의 사용 방법을 보여줍니다.

AWS CLI

옵션 그룹에 옵션 추가

다음 `add-option-to-option-group` 예제에서는 옵션을 지정된 옵션 그룹에 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 추가 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AddOptionToOptionGroup 섹션을 참조하세요.

`add-role-to-db-cluster`

다음 코드 예시에서는 `add-role-to-db-cluster`의 사용 방법을 보여줍니다.

AWS CLI

AWS Identity 및 Access Management(IAM) 역할을 DB 클러스터와 연결

다음 `add-role-to-db-cluster`는 DB 클러스터와 연결할 역할을 지정합니다.

```
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 IAM 역할을 Amazon Aurora MySQL DB 클러스터와 연결 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AddRoleToDbCluster 섹션을 참조하세요.

add-role-to-db-instance

다음 코드 예시에서는 `add-role-to-db-instance`의 사용 방법을 보여줍니다.

AWS CLI

AWS Identity 및 Access Management(IAM) 역할을 DB 인스턴스와 연결

다음 `add-role-to-db-instance` 명령은 `test-instance` Oracle DB 인스턴스에 역할을 추가합니다.

```
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 RDS 사용 설명서의 Amazon RDS Oracle과 Amazon S3 통합을 위한 사전 요구 사항 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AddRoleToDbInstance 섹션을 참조하세요.

add-source-identifier-to-subscription

다음 코드 예시에서는 `add-source-identifier-to-subscription`의 사용 방법을 보여줍니다.

AWS CLI

구독에 소스 식별자 추가

다음 `add-source-identifier` 예시에서는 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"
        ]
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [AddSourceIdentifierToSubscription](#) 섹션을 참조하세요.

`add-tags-to-resource`

다음 코드 예시에서는 `add-tags-to-resource`의 사용 방법을 보여줍니다.

AWS CLI

태그를 리소스에 추가

다음 add-tags-to-resource 예시에서는 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 RDS 사용 설명서의 Amazon RDS 리소스에 태그 지정 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AddTagsToResource를 참조하세요.

apply-pending-maintenance-action

다음 코드 예시에서는 apply-pending-maintenance-action의 사용 방법을 보여줍니다.

AWS CLI

보류 중인 유지 관리 작업 적용

다음 apply-pending-maintenance-action 예시에서는 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 Aurora 사용 설명서의 DB 인스턴스의 유지 관리 섹션 및 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ApplyPendingMaintenanceAction 섹션을 참조하세요.

authorize-db-security-group-ingress

다음 코드 예시에서는 authorize-db-security-group-ingress의 사용 방법을 보여줍니다.

AWS CLI

AWS Identity 및 Access Management(IAM) 역할을 DB 인스턴스와 연결

다음 authorize-db-security-group-ingress 예시에서는 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 네트워크 액세스 권한 부여 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AuthorizeDbSecurityGroupIngress 섹션을 참조하세요.

backtrack-db-cluster

다음 코드 예시에서는 backtrack-db-cluster의 사용 방법을 보여줍니다.

AWS CLI

Aurora DB 클러스터 역추적

다음 backtrack-db-cluster 예시에서는 지정된 DB 클러스터 샘플 클러스터를 2018년 3월 19일 오전 10시로 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.

- API 세부 정보는 AWS CLI 명령 참조의 BacktrackDbCluster 섹션을 참조하세요.

cancel-export-task

다음 코드 예시에서는 cancel-export-task의 사용 방법을 보여줍니다.

AWS CLI

Amazon S3로 스냅샷 내보내기 취소

다음 cancel-export-task 예시에서는 스냅샷을 Amazon S3로 내보내는 진행 중인 내보내기 태스크를 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": "amzn-s3-demo-bucket",
    "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 RDS 사용 설명서의 스냅샷 내보내기 태스크 취소 또는 Amazon Aurora 사용 설명서의 스냅샷 내보내기 태스크 취소 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CancelExportTask 참조하세요.

copy-db-cluster-parameter-group

다음 코드 예시에서는 copy-db-cluster-parameter-group의 사용 방법을 보여줍니다.

AWS CLI

DB 클러스터 파라미터 그룹 복사

다음 copy-db-cluster-parameter-group 예시에서는 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 RDS 사용 설명서의 [DB 클러스터 파라미터 그룹 복사](#) 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [CopyDbClusterParameterGroup](#) 섹션을 참조하세요.

`copy-db-cluster-snapshot`

다음 코드 예시에서는 `copy-db-cluster-snapshot`의 사용 방법을 보여줍니다.

AWS CLI

DB 클러스터 스냅샷 복사

다음 `copy-db-cluster-snapshot` 예시에서는 태그를 포함하여 DB 클러스터 스냅샷의 복사본을 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 EC2 사용 설명서의 스냅샷 복사 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CopyDbClusterSnapshot 섹션을 참조하세요.

copy-db-parameter-group

다음 코드 예시에서는 copy-db-parameter-group의 사용 방법을 보여줍니다.

AWS CLI

DB 클러스터 파라미터 그룹 복사

다음 copy-db-parameter-group 예시에서는 DB 파라미터 그룹을 생성합니다.

```
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 복사 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CopyDbParameterGroup 섹션을 참조하세요.

copy-db-snapshot

다음 코드 예시에서는 copy-db-snapshot의 사용 방법을 보여줍니다.

AWS CLI

DB 스냅샷 복사

다음 copy-db-snapshot 예시에서는 DB 스냅샷 복사본을 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 EC2 사용 설명서의 스냅샷 복사 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CopyDbSnapshot 섹션을 참조하세요.

copy-option-group

다음 코드 예시에서는 copy-option-group의 사용 방법을 보여줍니다.

AWS CLI

옵션 그룹을 복사하는 방법

다음 copy-option-group 예시에서는 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 복사복 생성 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CopyOptionGroup 섹션을 참조하세요.

create-blue-green-deployment

다음 코드 예시에서는 `create-blue-green-deployment`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: RDS for MySQL DB 인스턴스에 대한 블루/그린 배포 생성

다음 `create-blue-green-deployment` 예시에서는 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 생성 섹션을 참조하세요.

예시 1: Aurora MySQL DB 클러스터에 대한 블루/그린 배포 생성

다음 `create-blue-green-deployment` 예시에서는 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 생성 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateBlueGreenDeployment 섹션을 참조하세요.

`create-db-cluster-endpoint`

다음 코드 예시에서는 `create-db-cluster-endpoint`의 사용 방법을 보여줍니다.

AWS CLI

사용자 지정 DB 클러스터 엔드포인트를 생성하는 방법

다음 `create-db-cluster-endpoint` 예시에서는 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 관리 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateDbClusterEndpoint 섹션을 참조하세요.

create-db-cluster-parameter-group

다음 코드 예시에서는 create-db-cluster-parameter-group의 사용 방법을 보여줍니다.

AWS CLI

DB 클러스터 파라미터 그룹 생성

다음 create-db-cluster-parameter-group 예시에서는 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 파라미터 그룹 생성 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateDbClusterParameterGroup 섹션을 참조하세요.

`create-db-cluster-snapshot`

다음 코드 예시에서는 `create-db-cluster-snapshot`의 사용 방법을 보여줍니다.

AWS CLI

DB 클러스터 스냅샷 생성

다음 `create-db-cluster-snapshot` 예시에서는 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 생성 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateDbClusterSnapshot 섹션을 참조하세요.

create-db-cluster

다음 코드 예시에서는 `create-db-cluster`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: MySQL 5.7 호환 DB 클러스터 생성

다음 `create-db-cluster` 기본 엔진 버전을 사용하여 MySQL 5.7 호환 DB 클러스터를 생성합니다. 샘플 암호 `secret99`를 안전한 암호로 바꿉니다. 콘솔을 사용하여 DB 클러스터를 생성하면 Amazon RDS가 자동으로 DB 클러스터에 대한 쓰기 DB 인스턴스를 생성합니다. 그러나 AWS CLI를 사용하여 DB 클러스터를 생성하는 경우, `create-db-instance` AWS CLI 명령을 사용하여 명시적으로 DB 클러스터에 대한 쓰기 DB 인스턴스를 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
```

```
    }
}
```

예시 2: PostgreSQL - 호환 DB 클러스터 생성

다음 `create-db-cluster` 예시에서는 기본 엔진 버전을 사용하여 PostgreSQL 호환 DB 클러스터를 생성합니다. 예시 암호 `secret99`를 안전한 암호로 바꿉니다. 콘솔을 사용하여 DB 클러스터를 생성하면 Amazon RDS가 자동으로 DB 클러스터에 대한 쓰기 DB 인스턴스를 생성합니다. 그러나 AWS CLI를 사용하여 DB 클러스터를 생성하는 경우, `create-db-instance` AWS CLI 명령을 사용하여 명시적으로 DB 클러스터에 대한 쓰기 DB 인스턴스를 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 생성 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateDbCluster 섹션을 참조하세요.

`create-db-instance-read-replica`

다음 코드 예시에서는 `create-db-instance-read-replica`의 사용 방법을 보여줍니다.

AWS CLI

DB 인스턴스 읽기 복제본 생성

이 예시에서는 `test-instance` 기존 DB 인스턴스의 읽기 전용 복제본을 생성합니다. 읽기 전용 복제본의 이름은 `test-instance-repl`입니다.

```
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...
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateDbInstanceReadReplica 섹션을 참조하세요.

create-db-instance

다음 코드 예시에서는 create-db-instance의 사용 방법을 보여줍니다.

AWS CLI

DB 인스턴스 생성

다음 create-db-instance 예시에서는 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 RDS 사용자 안내서의 Amazon RDS DB 인스턴스 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateDBInstance를 참조하세요.

create-db-parameter-group

다음 코드 예시에서는 create-db-parameter-group의 사용 방법을 보여줍니다.

AWS CLI

DB 파라미터 그룹 생성

다음 create-db-parameter-group 예시에서는 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 사용자 안내서의 DB 파라미터 그룹 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateDBParameterGroup을 참조하세요.

create-db-proxy-endpoint

다음 코드 예시에서는 create-db-proxy-endpoint의 사용 방법을 보여줍니다.

AWS CLI

RDS 데이터베이스에 대한 DB 프록시 엔드포인트를 생성하는 방법

다음 `create-db-proxy-endpoint` 예시에서는 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 섹션 및 Amazon Aurora 사용 설명서의 프록시 엔드포인트 생성 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateDbProxyEndpoint 섹션을 참조하세요.

`create-db-proxy`

다음 코드 예시에서는 `create-db-proxy`의 사용 방법을 보여줍니다.

AWS CLI

RDS 데이터베이스에 대한 DB 프록시를 생성하는 방법

다음 `create-db-proxy` 예시에서는 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 섹션 및 Amazon Aurora 사용 설명서의 RDS 프록시 생성 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateDbProxy 섹션을 참조하세요.

create-db-security-group

다음 코드 예시에서는 `create-db-security-group`의 사용 방법을 보여줍니다.

AWS CLI

Amazon RDS DB 보안 그룹 생성

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 예시에서는 새 DB 보안 그룹의 이름이 `mysecgroup`으로 지정되고 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [CreateDbSecurityGroup](#) 섹션을 참조하세요.

`create-db-shard-group`

다음 코드 예시에서는 `create-db-shard-group`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: Aurora PostgreSQL DB 클러스터 생성

다음 `create-db-cluster` 예시에서는 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
```

```
        }
    }
}
```

예시 2: 기본(라이터) DB 인스턴스 생성

다음 `create-db-instance` 예시에서는 Aurora Serverless v2 기본(라이터) DB 인스턴스를 생성합니다. 콘솔을 사용하여 DB 클러스터를 생성하면 Amazon RDS가 자동으로 DB 클러스터에 대한 쓰기 DB 인스턴스를 생성합니다. 그러나 AWS CLI를 사용하여 DB 클러스터를 생성하는 경우, `create-db-instance` AWS CLI 명령을 사용하여 명시적으로 DB 클러스터에 대한 쓰기 DB 인스턴스를 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
    }
}
```

예시 3: DB 샤드 그룹 생성

다음 `create-db-shard-group` 예시에서는 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 사용](#) 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의[CreateDbShardGroup](#) 섹션을 참조하세요.

`create-db-snapshot`

다음 코드 예시에서는 `create-db-snapshot`의 사용 방법을 보여줍니다.

AWS CLI

DB 스냅샷 생성

다음 `create-db-snapshot` 예시에서는 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 사용자 안내서의 DB 스냅샷 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateDBSnapshot을 참조하세요.

create-db-subnet-group

다음 코드 예시에서는 create-db-subnet-group의 사용 방법을 보여줍니다.

AWS CLI

DB 서브넷 그룹 생성

다음 create-db-subnet-group 예시에서는 기존 서브넷을 사용하여 mysubnetgroup 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 VPC에서 DB 인스턴스 생성 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateDbSubnetGroup 섹션을 참조하세요.

create-event-subscription

다음 코드 예시에서는 `create-event-subscription`의 사용 방법을 보여줍니다.

AWS CLI

이벤트 구독 생성

다음 `create-event-subscription` 예시에서는 현재 AWS 계정에서 RDS Custom DB 인스턴스의 백업 및 복구 이벤트에 대한 구독을 생성합니다. 알림은 `--sns-topic-arn`에서 지정한 Amazon Simple Notification Service 주제로 전송됩니다.

```
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
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateEventSubscription을 참조하세요.

`create-global-cluster`

다음 코드 예시에서는 `create-global-cluster`의 사용 방법을 보여줍니다.

AWS CLI

글로벌 DB 클러스터 생성

다음 `create-global-cluster` 예시에서는 새 Aurora MySQL 호환 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 생성 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateGlobalCluster 섹션을 참조하세요.

`create-option-group`

다음 코드 예시에서는 `create-option-group`의 사용 방법을 보여줍니다.

AWS CLI

Amazon RDS 옵션 그룹 생성

다음 `create-option-group` 명령은 `Oracle Enterprise Edition` 버전 `11.2`, is named ``MyOptionGroup`에 대한 새 Amazon RDS 옵션 그룹을 생성하고 설명을 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateOptionGroup 섹션을 참조하세요.

delete-blue-green-deployment

다음 코드 예시에서는 `delete-blue-green-deployment`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: RDS for MySQL DB 인스턴스의 녹색 환경에서 리소스 삭제

다음 `delete-blue-green-deployment` 예시에서는 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 삭제](#) 섹션을 참조하세요.

예시 2: Aurora MySQL DB 클러스터의 녹색 환경에서 리소스 삭제

다음 `delete-blue-green-deployment` 예시에서는 Aurora MySQL DB 클러스터의 그린 환경에서 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 삭제](#) 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteBlueGreenDeployment](#) 섹션을 참조하세요.

delete-db-cluster-endpoint

다음 코드 예시에서는 `delete-db-cluster-endpoint`의 사용 방법을 보여줍니다.

AWS CLI

사용자 지정 DB 클러스터 엔드포인트를 삭제하는 방법

다음 `delete-db-cluster-endpoint` 예시에서는 지정된 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 관리 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteDbClusterEndpoint 섹션을 참조하세요.

delete-db-cluster-parameter-group

다음 코드 예시에서는 `delete-db-cluster-parameter-group`의 사용 방법을 보여줍니다.

AWS CLI

DB 클러스터 파라미터 그룹 삭제

다음 `delete-db-cluster-parameter-group` 예시에서는 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 작업 및 DB 클러스터 파라미터 작업 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteDbClusterParameterGroup 섹션을 참조하세요.

delete-db-cluster-snapshot

다음 코드 예시에서는 `delete-db-cluster-snapshot`의 사용 방법을 보여줍니다.

AWS CLI

DB 클러스터 스냅샷 삭제

다음 `delete-db-cluster-snapshot` 예시에서는 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 삭제 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteDbClusterSnapshot 섹션을 참조하세요.

`delete-db-cluster`

다음 코드 예시에서는 `delete-db-cluster`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: DB 클러스터의 DB 인스턴스 삭제

다음 `delete-db-instance` 예시에서는 DB 클러스터의 최종 DB 인스턴스를 삭제합니다. deleting 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 [Amazon Aurora DB 클러스터에서 DB 인스턴스 삭제](#) 섹션을 참조하세요.

예시 2: DB 클러스터 삭제

다음 `delete-db-cluster` 예시에서는 `mycluster` DB 클러스터를 삭제하고 `mycluster-final-snapshot` 최종 스냅샷을 생성합니다. 스냅샷을 찍는 동안 DB 클러스터의 상태가 available일 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 오로라 클러스터 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteDbCluster 섹션을 참조하세요.

`delete-db-instance-automated-backup`

다음 코드 예시에서는 `delete-db-instance-automated-backup`의 사용 방법을 보여줍니다.

AWS CLI

리전에서 복제된 자동 백업을 삭제하는 방법

다음 `delete-db-instance-automated-backup` 예시에서는 지정된 Amazon 리소스 이름(ARN)으로 자동화된 백업을 삭제합니다.

```
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 삭제](#) 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteDbInstanceAutomatedBackup](#) 섹션을 참조하세요.

`delete-db-instance`

다음 코드 예시에서는 `delete-db-instance`의 사용 방법을 보여줍니다.

AWS CLI

DB 인스턴스 삭제

다음 `delete-db-instance` 예시에서는 `test-instance-final-snap`이라는 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...
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteDBInstance를 참조하세요.

delete-db-parameter-group

다음 코드 예시에서는 delete-db-parameter-group의 사용 방법을 보여줍니다.

AWS CLI

DB 파라미터 그룹 삭제

다음 command 예시에서는 DB 파라미터 그룹을 삭제합니다.

```
aws rds delete-db-parameter-group \
    --db-parameter-group-name mydbparametergroup
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon RDS 사용자 안내서의 DB 파라미터 그룹 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteDBParameterGroup을 참조하세요.

delete-db-proxy-endpoint

다음 코드 예시에서는 delete-db-proxy-endpoint의 사용 방법을 보여줍니다.

AWS CLI

RDS 데이터베이스의 DB 프록시 엔드포인트를 삭제하는 방법

다음 delete-db-proxy-endpoint 예시에서는 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] 섹션 및 Amazon Aurora 사용 설명서의 [프록시 엔드포인트 삭제] 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteDbProxyEndpoint] 섹션을 참조하세요.

delete-db-proxy

다음 코드 예시에서는 delete-db-proxy의 사용 방법을 보여줍니다.

AWS CLI

RDS 데이터베이스의 DB 프록시를 삭제하는 방법

다음 `delete-db-proxy` 예시에서는 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 RES 프록시 삭제 섹션 및 Amazon Aurora 사용 설명서의 RDS 프록시 삭제 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteDbProxy 섹션을 참조하세요.

delete-db-security-group

다음 코드 예시에서는 delete-db-security-group의 사용 방법을 보여줍니다.

AWS CLI

DB 보안 그룹 삭제

다음 delete-db-security-group 예시에서는 mysecuritygroup DB 보안 그룹을 삭제합니다.

```
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) 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteDbSecurityGroup 섹션을 참조하세요.

delete-db-shard-group

다음 코드 예시에서는 delete-db-shard-group의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 비정상적으로 DB 샤드 그룹 삭제

다음 delete-db-shard-group 예시에서는 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.
```

예시 2: DB 샤드 그룹을 성공적으로 삭제

다음 `delete-db-shard-group` 예시에서는 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 삭제 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteDbShardGroup 섹션을 참조하세요.

`delete-db-snapshot`

다음 코드 예시에서는 `delete-db-snapshot`의 사용 방법을 보여줍니다.

AWS CLI

DB 스냅샷 삭제

다음 delete-db-snapshot 예시에서는 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 EC2 사용 설명서의 스냅샷 삭제 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteDbSnapshot 섹션을 참조하세요.

delete-db-subnet-group

다음 코드 예시에서는 delete-db-subnet-group의 사용 방법을 보여줍니다.

AWS CLI

DB 서브넷 그룹 삭제

다음 `delete-db-subnet-group` 예시에서는 `mysubnetgroup` DB 서브넷 그룹을 삭제합니다.

```
aws rds delete-db-subnet-group --db-subnet-group-name mysubnetgroup
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon RDS 사용 설명서의 [VPC에서 DB 인스턴스 작업](#) 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteDbSubnetGroup](#) 섹션을 참조하세요.

delete-event-subscription

다음 코드 예시에서는 `delete-event-subscription`의 사용 방법을 보여줍니다.

AWS CLI

이벤트 구독 삭제

다음 `delete-event-subscription` 예시에서는 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"
        }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteEventSubscription을 참조하세요.

`delete-global-cluster`

다음 코드 예시에서는 `delete-global-cluster`의 사용 방법을 보여줍니다.

AWS CLI

글로벌 DB 클러스터 삭제

다음 `delete-global-cluster` 예시에서는 Aurora MySQL 호환 글로벌 DB 클러스터를 삭제합니다. 출력에는 삭제 중인 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 삭제 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteGlobalCluster](#) 섹션을 참조하세요.

delete-option-group

다음 코드 예시에서는 delete-option-group의 사용 방법을 보여줍니다.

AWS CLI

옵션 그룹 삭제

다음 delete-option-group 예시에서는 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 삭제](#) 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteOptionGroup](#) 섹션을 참조하세요.

deregister-db-proxy-targets

다음 코드 예시에서는 deregister-db-proxy-targets의 사용 방법을 보여줍니다.

AWS CLI

데이터베이스 대상 그룹에서 DB 프록시 대상을 등록 취소하는 방법

다음 deregister-db-proxy-targets 예시에서는 proxyExample 프록시와 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 [RES 프록시 삭제](#) 섹션 및 Amazon Aurora 사용 설명서의 [RDS 프록시 삭제](#) 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DeregisterDbProxyTargets](#) 섹션을 참조하세요.

describe-account-attributes

다음 코드 예시에서는 describe-account-attributes의 사용 방법을 보여줍니다.

AWS CLI

계정 속성 설명

다음 describe-account-attributes 예시에서는 현재 AWS 계정의 속성을 가져옵니다.

```
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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeAccountAttributes를 참조하세요.

describe-blue-green-deployments

다음 코드 예시에서는 describe-blue-green-deployments의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 생성이 완료된 후 RDS DB 인스턴스의 블루/그린 배포 설명

다음 describe-blue-green-deployment 예시에서는 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 보기] 섹션을 참조하세요.

예시 2: Aurora MySQL DB 클러스터의 블루/그린 배포 설명

다음 `describe-blue-green-deployment` 예시에서는 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 보기](#) 섹션을 참조하세요.

예시 3: 전환 후 Aurora MySQL 클러스터의 블루/그린 배포 설명

다음 `describe-blue-green-deployment` 예시에서는 그린 환경이 프로덕션 환경으로 승격된 후 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-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 보기 섹션을 참조하세요.

예시 4: 결합된 블루/그린 배포 설명

다음 `describe-blue-green-deployment` 예시에서는 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 섹션 및 Amazon Aurora 사용 설명서의 블루/그린 배포 보기 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeBlueGreenDeployments 섹션을 참조하세요d

`describe-certificates`

다음 코드 예시에서는 `describe-certificates`의 사용 방법을 보여줍니다.

AWS CLI

인증서 설명

다음 `describe-certificates` 예시에서는 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 인스턴스에 대한 연결 암호화 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeCertificates를 참조하세요.

describe-db-cluster-backtracks

다음 코드 예시에서는 describe-db-cluster-backtracks의 사용 방법을 보여줍니다.

AWS CLI

DB 클러스터의 역추적을 설명하는 방법

다음 describe-db-cluster-backtracks 예시에서는 지정된 DB 클러스터의 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 역추적 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDbClusterBacktracks 섹션을 참조하세요.

`describe-db-cluster-endpoints`

다음 코드 예시에서는 `describe-db-cluster-endpoints`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: DB 클러스터 엔드포인트 설명

다음 `describe-db-cluster-endpoints` 예시에서는 DB 클러스터 엔드포인트의 세부 정보를 검색합니다. 가장 일반적인 종류의 Aurora 클러스터에는 두 개의 엔드포인트가 있습니다. 하나의 엔드포인트에는 `WRITER` 유형이 있습니다. 이 엔드포인트는 모든 SQL 문에 사용할 수 있습니다. 다른 엔드포인트에는 `READER` 유형이 있습니다. 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"
        }
    ]
}
```

예시 2: 단일 DB 클러스터의 DB 클러스터 엔드포인트 설명

다음 `describe-db-cluster-endpoints` 예시에서는 지정된 단일 DB 클러스터의 DB 클러스터 엔드포인트의 세부 정보를 검색합니다. Aurora Serverless 클러스터에는 유형이 `WRITER`인 단일 엔드포인트만 있습니다.

```
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 관리](#) 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeDbClusterEndpoints](#) 섹션을 참조하세요.

`describe-db-cluster-parameter-groups`

다음 코드 예시에서는 `describe-db-cluster-parameter-groups`의 사용 방법을 보여줍니다.

AWS CLI

DB 클러스터 파라미터 그룹 설명

다음 describe-db-cluster-parameter-groups 예시에서는 DB 클러스터 파라미터 그룹의 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 작업 및 DB 클러스터 파라미터 작업 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDbClusterParameterGroups 섹션을 참조하세요.

describe-db-cluster-parameters

다음 코드 예시에서는 `describe-db-cluster-parameters`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: DB 클러스터 파라미터 그룹에 파라미터 설명

다음 `describe-db-cluster-parameters` 예시에서는 DB 클러스터 파라미터 그룹에 있는 파라미터의 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...
    ]
}
```

예시 2: DB 클러스터 파라미터 그룹의 파라미터 이름만 나열

다음 `describe-db-cluster-parameters` 예시에서는 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...
    }
]
```

예시 3: DB 클러스터 파라미터 그룹에서 수정 가능한 파라미터만 설명

다음 `describe-db-cluster-parameters` 예시에서는 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...
    }
]
```

예시 4: DB 클러스터 파라미터 그룹에서 수정 가능한 부울 파라미터만 설명

다음 `describe-db-cluster-parameters` 예시에서는 DB 클러스터 파라미터 그룹에서 수정할 수 있고 부울 데이터 형식을 가진 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 작업 및 DB 클러스터 파라미터 작업 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDbClusterParameters 섹션을 참조하세요.

describe-db-cluster-snapshot-attributes

다음 코드 예시에서는 describe-db-cluster-snapshot-attributes의 사용 방법을 보여줍니다.

AWS CLI

DB 클러스터 스냅샷의 속성 이름 및 값을 설명하는 방법

다음 describe-db-cluster-snapshot-attributes 예시에서는 지정된 DB 클러스터 스냅샷의 속성 이름 및 값의 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 공유 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDbClusterSnapshotAttributes 섹션을 참조하세요.

describe-db-cluster-snapshots

다음 코드 예시에서는 describe-db-cluster-snapshots의 사용 방법을 보여줍니다.

AWS CLI

DB 클러스터에 대한 DB 클러스터 스냅샷을 설명하는 방법

다음 describe-db-cluster-snapshots 예시에서는 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 생성 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDbClusterSnapshots 섹션을 참조하세요.

describe-db-clusters

다음 코드 예시에서는 describe-db-clusters의 사용 방법을 보여줍니다.

AWS CLI

예시 1: DB 클러스터 설명

다음 describe-db-clusters 예시에서는 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
        }
```

```
    ]
}
```

예시 2: 모든 DB 클러스터의 특정 속성 나열

다음 describe-db-clusters 예시에서는 현재 AWS 리전에 있는 모든 DB 클러스터의 DBClusterIdentifier, Endpoint, 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...
    }
]
```

예시 3: 특정 속성이 있는 DB 클러스터 나열

다음 describe-db-clusters 예시에서는 aurora-postgresql DB 엔진을 사용하는 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 클러스터 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDbClusters 섹션을 참조하세요.

describe-db-engine-versions

다음 코드 예시에서는 describe-db-engine-versions의 사용 방법을 보여줍니다.

AWS CLI

MySQL DB 엔진의 DB 엔진 버전 설명

다음 describe-db-engine-versions 예시에서는 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 RDS 사용자 안내서의 Amazon Relational Database Service(Amazon RDS)란?을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDBEngineVersions를 참조하세요.

describe-db-instance-automated-backups

다음 코드 예시에서는 `describe-db-instance-automated-backups`의 사용 방법을 보여줍니다.

AWS CLI

DB 인스턴스의 자동 백업 설명

다음 describe-db-instance-automated-backups 예시에서는 지정된 DB 인스턴스에 대한 자동화된 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 찾기](#) 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDbInstanceAutomatedBackups 섹션을 참조하세요.

`describe-db-instances`

다음 코드 예시에서는 `describe-db-instances`의 사용 방법을 보여줍니다.

AWS CLI

DB 인스턴스 설명

다음 `describe-db-instances` 예시에서는 지정된 DB 인스턴스의 세부 정보를 가져옵니다.

```
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...
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDBInstances를 참조하세요.

`describe-db-log-files`

다음 코드 예시에서는 `describe-db-log-files`의 사용 방법을 보여줍니다.

AWS CLI

DB 인스턴스의 로그 파일을 설명하는 방법

다음 describe-db-log-files 예시에서는 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDbLogFiles 섹션을 참조하세요.

describe-db-parameter-groups

다음 코드 예시에서는 describe-db-parameter-groups의 사용 방법을 보여줍니다.

AWS CLI

DB 파라미터 그룹 설명

다음 describe-db-parameter-groups 예시에서는 DB 파라미터 그룹의 세부 정보를 가져옵니다.

```
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 사용자 안내서의 DB 파라미터 그룹 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDBParameterGroups를 참조하세요.

`describe-db-parameters`

다음 코드 예시에서는 `describe-db-parameters`의 사용 방법을 보여줍니다.

AWS CLI

DB 파라미터 그룹의 파라미터 설명

다음 `describe-db-parameters` 예시에서는 지정된 DB 파라미터 그룹의 세부 정보를 가져옵니다.

```
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 사용자 안내서의 [DB 파라미터 그룹 작업]을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeDBParameters]를 참조하세요.

describe-db-proxies

다음 코드 예시에서는 describe-db-proxies의 사용 방법을 보여줍니다.

AWS CLI

RDS 데이터베이스의 DB 프록시를 설명하는 방법

다음 describe-db-proxies 예시에서는 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 RES 프록시 보기 섹션 및 Amazon Aurora 사용 설명서의 RDS 프록시 보기 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDbProxies 섹션을 참조하세요.

describe-db-proxy-endpoints

다음 코드 예시에서는 describe-db-proxy-endpoints의 사용 방법을 보여줍니다.

AWS CLI

DB 프록시 엔드포인트를 설명하는 방법

다음 describe-db-proxy-endpoints 예시에서는 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 섹션 및 Amazon Aurora 사용 설명서의 프록시 엔드포인트 생성 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDbProxyEndpoints 섹션을 참조하세요.

`describe-db-proxy-target-groups`

다음 코드 예시에서는 `describe-db-proxy-target-groups`의 사용 방법을 보여줍니다.

AWS CLI

DB 프록시 엔드포인트를 설명하는 방법

다음 `describe-db-proxy-target-groups` 예시에서는 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 RES 프록시 보기 섹션 및 Amazon Aurora 사용 설명서의 RDS 프록시 보기 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDbProxyTargetGroups 섹션을 참조하세요.

`describe-db-proxy-targets`

다음 코드 예시에서는 `describe-db-proxy-targets`의 사용 방법을 보여줍니다.

AWS CLI

DB 프록시 대상을 설명하는 방법

다음 `describe-db-proxy-targets` 예시에서는 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 [RES 프록시 보기] 섹션 및 Amazon Aurora 사용 설명서의 [RDS 프록시 보기] 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeDbProxyTargets] 섹션을 참조하세요.

describe-db-recommendations

다음 코드 예시에서는 `describe-db-recommendations`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 모든 DB 권장 사항 나열

다음 `describe-db-recommendations` 예시에서는 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 RDS 사용 설명서의 [Amazon RDS 권장 사항 보기 및 대응](#) 섹션 및 Amazon Aurora 사용 설명서의 [Amazon RDS 권장 사항 보기 및 대응](#) 섹션을 참조하세요.

예시 2: 심각도가 높은 DB 권장 사항 나열

다음 `describe-db-recommendations` 예시에서는 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 RDS 사용 설명서의 Amazon RDS 권장 사항 보기 및 대응 섹션 및 Amazon Aurora 사용 설명서의 Amazon RDS 권장 사항 보기 및 대응 섹션을 참조하세요.

예시 3: 지정된 DB 인스턴스에 대한 DB 권장 사항 나열

다음 `describe-db-recommendations` 예시에서는 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 RDS 사용 설명서의 [Amazon RDS 권장 사항 보기 및 대응](#) 섹션 및 Amazon Aurora 사용 설명서의 [Amazon RDS 권장 사항 보기 및 대응](#) 섹션을 참조하세요.

예시 4: 모든 활성 DB 권장 사항 나열

다음 `describe-db-recommendations` 예시에서는 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 RDS 사용 설명서의 Amazon RDS 권장 사항 보기 및 대응 섹션 및 Amazon Aurora 사용 설명서의 Amazon RDS 권장 사항 보기 및 대응 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDbRecommendations 섹션을 참조하세요.

describe-db-security-groups

다음 코드 예시에서는 describe-db-security-groups의 사용 방법을 보여줍니다.

AWS CLI

DB 보안 그룹을 나열하는 방법

다음 describe-db-security-groups 예시에서는 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 나열](#) 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeDbSecurityGroups](#) 섹션을 참조하세요.

`describe-db-shard-groups`

다음 코드 예시에서는 `describe-db-shard-groups`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: DB 샤드 그룹 설명

다음 `describe-db-shard-groups` 예시에서는 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 클러스터 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDbShardGroups 섹션을 참조하세요.

describe-db-snapshot-attributes

다음 코드 예시에서는 describe-db-snapshot-attributes의 사용 방법을 보여줍니다.

AWS CLI

DB 스냅샷의 속성 이름 및 값을 설명하는 방법

다음 describe-db-snapshot-attributes 예시에서는 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 Aurora 사용 설명서의 DB 스냅샷 공유 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDbSnapshotAttributes 섹션을 참조하세요.

describe-db-snapshots

다음 코드 예시에서는 describe-db-snapshots의 사용 방법을 보여줍니다.

AWS CLI

예시 1: DB 인스턴스의 DB 스냅샷 설명

다음 describe-db-snapshots 예시에서는 DB 인스턴스의 DB 스냅샷에 대한 세부 정보를 가져옵니다.

```
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 사용자 안내서의 DB 스냅샷 생성을 참조하세요.

예시 2: 생성한 수동 스냅샷의 수 확인

다음 describe-db-snapshots 예시에서는 --query 옵션의 length 연산자를 사용하여 특정 AWS 리전에서 생성된 수동 스냅샷의 수를 반환합니다.

```
aws rds describe-db-snapshots \
    --snapshot-type manual \
    --query "length(*[].{DBSnapshots:SnapshotType})" \
    --region eu-central-1
```

출력:

```
35
```

자세한 내용은 Amazon RDS 사용자 안내서의 DB 스냅샷 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDBSnapshots를 참조하세요.

describe-db-subnet-groups

다음 코드 예시에서는 describe-db-subnet-groups의 사용 방법을 보여줍니다.

AWS CLI

DB 서브넷 그룹 설명

다음 `describe-db-subnet-groups` 예시에서는 지정된 DB 서브넷 그룹의 상세 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 RDS 사용 설명서의 Amazon Virtual Private Cloud VPC 및 Amazon RDS 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDbSubnetGroups 섹션을 참조하세요.

describe-engine-default-cluster-parameters

다음 코드 예시에서는 describe-engine-default-cluster-parameters의 사용 방법을 보여줍니다.

AWS CLI

Aurora 데이터베이스 엔진의 기본 엔진 및 시스템 파라미터 정보 설명

다음 describe-engine-default-cluster-parameters 예시에서는 MySQL 5.7 호환성을 갖춘 Aurora DB 클러스터의 기본 엔진 및 시스템 파라미터 정보에 대한 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 작업 및 DB 클러스터 파라미터 작업 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeEngineDefaultClusterParameters 섹션을 참조하세요.

`describe-engine-default-parameters`

다음 코드 예시에서는 `describe-engine-default-parameters`의 사용 방법을 보여줍니다.

AWS CLI

데이터베이스 엔진의 기본 엔진 및 시스템 파라미터 정보 설명

다음 `describe-engine-default-parameters` 예시에서는 MySQL 5.7 DB 인스턴스의 기본 엔진 및 시스템 파라미터 정보의 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 사용자 안내서의 [DB 파라미터 그룹 작업](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_WorkingWithParamGroups.html)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeEngineDefaultParameters](https://awscli.amazonaws.com/v2/documentation/api/latest/reference/rds/describe-engine-default-parameters.html) 섹션을 참조하세요.

`describe-event-categories`

다음 코드 예시에서는 `describe-event-categories`의 사용 방법을 보여줍니다.

AWS CLI

이벤트 카테고리 설명

다음 `describe-event-categories` 예시에서는 사용 가능한 모든 이벤트 소스에 대한 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"
            ]
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeEventCategories를 참조하세요.

describe-event-subscriptions

다음 코드 예시에서는 describe-event-subscriptions 코드를 사용하는 방법을 보여줍니다.

AWS CLI

이벤트 구독 설명

이 예시에서는 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...
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeEventSubscriptions를 참조하세요.

describe-events

다음 코드 예시에서는 describe-events의 사용 방법을 보여줍니다.

AWS CLI

이벤트 설명

다음 describe-events 예시에서는 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeEvents 섹션을 참조하세요.

`describe-export-tasks`

다음 코드 예시에서는 `describe-export-tasks`의 사용 방법을 보여줍니다.

AWS CLI

스냅샷 내보내기 태스크 설명

다음 `describe-export-tasks` 예시에서는 Amazon S3로 스냅샷 내보내기에 대한 정보를 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": "amzn-s3-demo-bucket",
            "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": "amzn-s3-demo-bucket",
            "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 모니터링 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeExportTasks 섹션을 참조하세요.

`describe-global-clusters`

다음 코드 예시에서는 `describe-global-clusters`의 사용 방법을 보여줍니다.

AWS CLI

글로벌 DB 클러스터를 설명하는 방법

다음 `describe-global-clusters` 예시에서는 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 관리 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeGlobalClusters 섹션을 참조하세요.

describe-option-group-options

다음 코드 예시에서는 describe-option-group-options의 사용 방법을 보여줍니다.

AWS CLI

사용 가능한 모든 옵션 설명

다음 describe-option-group-options 예시에서는 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 나열 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeOptionGroupOptions 섹션을 참조하세요.

describe-option-groups

다음 코드 예시에서는 `describe-option-groups`의 사용 방법을 보여줍니다.

AWS CLI

사용 가능한 옵션 그룹 설명

다음 `describe-option-groups` 예시에서는 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 나열 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeOptionGroups 섹션을 참조하세요.

`describe-orderable-db-instance-options`

다음 코드 예시에서는 `describe-orderable-db-instance-options`의 사용 방법을 보여줍니다.

AWS CLI

주문 가능한 DB 인스턴스 옵션 설명

다음 `describe-orderable-db-instance-options` 예시에서는 MySQL DB 엔진을 실행 중인 DB 인스턴스의 주문 가능한 옵션에 대한 세부 정보를 가져옵니다.

```
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...
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeOrderableDBInstanceOptions를 참조하세요.

describe-pending-maintenance-actions

다음 코드 예시에서는 describe-pending-maintenance-actions의 사용 방법을 보여줍니다.

AWS CLI

하나 이상의 보류 중인 유지 관리 작업이 있는 리소스를 나열하는 방법

다음 describe-pending-maintenance-actions 예시에서는 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 관리 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribePendingMaintenanceActions 섹션을 참조하세요.

describe-reserved-db-instances-offerings

다음 코드 예시에서는 describe-reserved-db-instances-offerings의 사용 방법을 보여줍니다.

AWS CLI

예약 DB 인스턴스 오퍼링 설명

다음 describe-reserved-db-instances-offerings 예시에서는 oracle에 대해 정기 DB 인스턴스 옵션에 대한 세부 정보를 검색합니다.

```
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...
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeReservedDbInstancesOfferings 섹션을 참조하세요.

describe-reserved-db-instances

다음 코드 예시에서는 describe-reserved-db-instances의 사용 방법을 보여줍니다.

AWS CLI

예약 DB 인스턴스 설명

다음 `describe-reserved-db-instances` 예시에서는 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 RDS 사용 설명서의 [Amazon RDS용 정기 DB 인스턴스](#) 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeReservedDbInstances](#) 섹션을 참조하세요.

describe-source-regions

다음 코드 예시에서는 describe-source-regions의 사용 방법을 보여줍니다.

AWS CLI

소스 리전을 설명하는 방법

다음 describe-source-regions 예시에서는 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 찾기 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeSourceRegions 섹션을 참조하세요.

describe-valid-db-instance-modifications

다음 코드 예시에서는 describe-valid-db-instance-modifications의 사용 방법을 보여줍니다.

AWS CLI

DB 인스턴스에 대한 유효한 수정 사항을 설명하는 방법

다음 describe-valid-db-instance-modifications 예시에서는 지정된 DB 인스턴스에 대한 유효한 수정 사항에 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"
            }
        ]
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeValidDbInstanceModifications](#) 섹션을 참조하세요.

download-db-log-file-portion

다음 코드 예시는 download-db-log-file-portion의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: DB 로그 파일의 최신 부분 다운로드

다음 download-db-log-file-portion 예시에서는 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
```

저장된 파일에 빈 줄이 있을 수 있습니다. 다운로드하는 동안 로그 파일의 각 부분 끝에 표시됩니다.

예제 2: 전체 DB 로그 파일 다운로드

다음 download-db-log-file-portion 예제에서는 --starting-token 0 파라미터를 사용하여 전체 로그 파일을 다운로드하고 출력을 full.txt라는 로컬 파일에 저장합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 DownloadDbLogFilePortion 섹션을 참조하세요.

generate-auth-token

다음 코드 예시에서는 generate-auth-token의 사용 방법을 보여줍니다.

AWS CLI

인증 토큰을 생성하는 방법

다음 generate-db-auth-token 예시에서는 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
```

- API 세부 정보는 AWS CLI 명령 참조의 GenerateAuthToken 섹션을 참조하세요.

`generate-db-auth-token`

다음 코드 예시에서는 `generate-db-auth-token`의 사용 방법을 보여줍니다.

AWS CLI

IAM 인증 토큰을 생성하는 방법

다음 `generate-db-auth-token` 예시에서는 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 섹션 및 Amazon Aurora 사용 설명서의 IAM 인증을 사용하여 DB 클러스터에 연결 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GenerateDbAuthToken 섹션을 참조하세요.

`list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`의 사용 방법을 보여줍니다.

AWS CLI

Amazon RDS 리소스에 태그 나열

다음 `list-tags-for-resource` 예시에서는 DB 인스턴스에 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 RDS 사용 설명서의 Amazon RDS 리소스에 태그 지정 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource를 참조하세요.

`modify-certificates`

다음 코드 예시에서는 `modify-certificates`의 사용 방법을 보여줍니다.

AWS CLI

새 DB 인스턴스에 대한 시스템 기본 SSL/TLS 인증서를 일시적으로 재정의하는 방법

다음 modify-certificates 예시에서는 새 DB 인스턴스에 대한 시스템 기본 SSL/TLS 인증서를 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 섹션 및 Amazon Aurora 사용 설명서의 SSL/TLS 인증서 교체 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyCertificates 섹션을 참조하세요.

modify-current-db-cluster-capacity

다음 코드 예시에서는 modify-current-db-cluster-capacity의 사용 방법을 보여줍니다.

AWS CLI

Aurora Serverless DB 클러스터의 용량을 조정하는 방법

다음 modify-current-db-cluster-capacity 예시에서는 Aurora 서버리스 DB 클러스터의 용량을 8로 확장합니다.

```
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 Aurora Serverless v1 DB 클러스터 용량 수동으로 확장 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyCurrentDbClusterCapacity 섹션을 참조하세요.

modify-db-cluster-endpoint

다음 코드 예시에서는 modify-db-cluster-endpoint의 사용 방법을 보여줍니다.

AWS CLI

사용자 지정 DB 클러스터 엔드포인트를 수정하는 방법

다음 modify-db-cluster-endpoint 예시에서는 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 관리](#) 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ModifyDbClusterEndpoint](#) 섹션을 참조하세요.

modify-db-cluster-parameter-group

다음 코드 예시에서는 `modify-db-cluster-parameter-group`의 사용 방법을 보여줍니다.

AWS CLI

DB 클러스터 파라미터 그룹에서 파라미터 수정

다음 `modify-db-cluster-parameter-group` 예시에서는 DB 파라미터 그룹의 파라미터 값을 변경합니다.

```
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 작업 및 DB 클러스터 파라미터 작업](#) 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ModifyDbClusterParameterGroup](#) 섹션을 참조하세요.

modify-db-cluster-snapshot-attribute

다음 코드 예시에서는 `modify-db-cluster-snapshot-attribute`의 사용 방법을 보여줍니다.

AWS CLI

DB 클러스터 스냅샷 속성 수정

다음 `modify-db-cluster-snapshot-attribute` 예시에서는 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 복원](#) 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ModifyDbClusterSnapshotAttribute](#) 섹션을 참조하세요.

modify-db-cluster

다음 코드 예시에서는 `modify-db-cluster`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: DB 클러스터 수정

다음 `modify-db-cluster` 예시에서는 `cluster-2` DB 클러스터의 마스터 사용자 암호를 변경하고 백업 보존 기간을 14일로 설정합니다. `--apply-immediately` 파라미터를 사용하면 다음 유지 관리 기간까지 기다리지 않고 즉시 변경이 적용됩니다.

```
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 수정 섹션을 참조하세요.

예시 2: VPC 보안 그룹을 DB 클러스터와 연결

다음 `modify-db-instance` 예시에서는 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 통한 액세스 제어 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyDbCluster 섹션을 참조하세요.

modify-db-instance

다음 코드 예시에서는 `modify-db-instance`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: DB 인스턴스 수정

다음 `modify-db-instance` 예시에서는 옵션 그룹과 파라미터 그룹을 호환되는 Microsoft SQL Server DB 인스턴스에 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 RDS 사용자 안내서의 Amazon RDS DB 인스턴스 수정을 참조하세요.

예시 2: DB 인스턴스에 VPC 보안 그룹을 연결

다음 `modify-db-instance` 예시에서는 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 사용자 안내서의 보안 그룹을 통한 액세스 제어를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyDBInstance를 참조하세요.

modify-db-parameter-group

다음 코드 예시에서는 `modify-db-parameter-group`의 사용 방법을 보여줍니다.

AWS CLI

DB 파라미터 그룹 수정

다음 modify-db-parameter-group 예시에서는 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 사용자 안내서의 DB 파라미터 그룹의 파라미터 수정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyDBParameterGroup을 참조하세요.

modify-db-proxy-endpoint

다음 코드 예시에서는 modify-db-proxy-endpoint의 사용 방법을 보여줍니다.

AWS CLI

RDS 데이터베이스의 DB 프록시 엔드포인트를 수정하는 방법

다음 modify-db-proxy-endpoint 예시에서는 DB 프록시 엔드포인트 proxyEndpoint를 수정하여 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 섹션 및 Amazon Aurora 사용 설명서의 프록시 엔드포인트 수정 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyDbProxyEndpoint 섹션을 참조하세요.

modify-db-proxy-target-group

다음 코드 예시에서는 modify-db-proxy-target-group의 사용 방법을 보여줍니다.

AWS CLI

DB 프록시 엔드포인트를 수정하는 방법

다음 modify-db-proxy-target-group 예시에서는 DB 프록시 대상 그룹을 수정하여 최대 연결을 80%로, 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 섹션 및 Amazon Aurora 사용 설명서의 RDS 프록시 수정 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyDbProxyTargetGroup 섹션을 참조하세요.

modify-db-proxy

다음 코드 예시에서는 modify-db-proxy의 사용 방법을 보여줍니다.

AWS CLI

RDS 데이터베이스의 DB 프록시를 수정하는 방법

다음 modify-db-proxy 예시에서는 연결을 위해 SSL이 필요하도록 proxyExample DB 프록시를 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 섹션 및 Amazon Aurora 사용 설명서의 RDS 프록시 수정 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyDbProxy 섹션을 참조하세요.

`modify-db-shard-group`

다음 코드 예시에서는 `modify-db-shard-group`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: DB 샤드 그룹 수정

다음 modify-db-shard-group 예시에서는 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 클러스터](#) 섹션을 참조하세요.

예시 2: DB 샤드 그룹 설명

다음 describe-db-shard-groups 예시에서는 modify-db-shard-group 명령을 실행한 후 DB 샤드 그룹의 세부 정보를 검색합니다. DB 샤드 그룹 my-db-shard-group의 최대 용량은 이제 이제 1000개의 Aurora 용량 단위(ACU)입니다.

```
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 클러스터 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyDbShardGroup 섹션을 참조하세요.

modify-db-snapshot-attribute

다음 코드 예시에서는 modify-db-snapshot-attribute의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 두 AWS 계정이 DB 스냅샷을 복원하도록 활성화

다음 modify-db-snapshot-attribute 예시에서는 식별자가 111122223333과 444455556666인 두 AWS 계정에 권한을 부여하여 mydbsnapshot DB 스냅샷을 복원할 수 있도록 합니다.

```
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 공유 섹션을 참조하세요.

예시 2: AWS 계정이 DB 스냅샷의 복원 방지

다음 `modify-db-snapshot-attribute` 예시에서는 mydbsnapshot라는 DB 스냅샷을 복원할 수 있는 권한을 특정 AWS 계정에서 제거합니다. 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 공유 섹션을 참조하세요.

• API 세부 정보는 AWS CLI 명령 참조의 ModifyDbSnapshotAttribute 섹션을 참조하세요.

`modify-db-snapshot-attributes`

다음 코드 예시에서는 `modify-db-snapshot-attributes`의 사용 방법을 보여줍니다.

AWS CLI

DB 스냅샷 속성을 수정하는 방법

다음 `modify-db-snapshot-attribute` 예시에서는 식별자가 `111122223333` 및 `444455556666`인 두 AWS 계정에 대해 `mydbsnapshot` 스냅샷 복원을 허용합니다.

```
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 공유 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyDbSnapshotAttributes 섹션을 참조하세요.

modify-db-snapshot

다음 코드 예시에서는 `modify-db-snapshot`의 사용 방법을 보여줍니다.

AWS CLI

DB 스냅샷 수정

다음 `modify-db-snapshot` 예시에서는 이름이 `db5-snapshot-upg-test`인 PostgeSQL PostgreSQL 10.6 스냅샷을 업그레이드합니다. 새 DB 엔진 버전은 스냅샷 업그레이드가 완료되고 해당 상태가 available이면 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 업그레이드 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyDbSnapshot 섹션을 참조하세요.

modify-db-subnet-group

다음 코드 예시에서는 modify-db-subnet-group의 사용 방법을 보여줍니다.

AWS CLI

DB 서브넷 그룹 수정

다음 modify-db-subnet-group 예시에서는 ID가 subnet-08e41f9e230222222인 서브넷을 mysubnetgroup DB 서브넷 그룹에 추가합니다. 서브넷 그룹에 기존 서브넷을 유지하려면 --subnet-ids 옵션에 해당 IDs 값으로 포함합니다. 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 생성 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyDbSubnetGroup 섹션을 참조하세요.

`modify-event-subscription`

다음 코드 예시에서는 `modify-event-subscription`의 사용 방법을 보여줍니다.

AWS CLI

이벤트 구독 수정

다음 modify-event-subscription 예시에서는 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
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ModifyEventSubscription을 참조하세요.

modify-global-cluster

다음 코드 예시에서는 modify-global-cluster의 사용 방법을 보여줍니다.

AWS CLI

DB 클러스터 수정

다음 modify-global-cluster 예시에서는 Aurora MySQL 호환 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 관리 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyGlobalCluster 섹션을 참조하세요.

promote-read-replica-db-cluster

다음 코드 예시에서는 `promote-read-replica-db-cluster`의 사용 방법을 보여줍니다.

AWS CLI

DB 클러스터 읽기 전용 복제본 승격

다음 `promote-read-replica-db-cluster` 예시에서는 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 승격를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PromoteReadReplicaDbCluster 섹션을 참조하세요.

promote-read-replica

다음 코드 예시에서는 promote-read-replica의 사용 방법을 보여줍니다.

AWS CLI

읽기 전용 복제본 승격

다음 promote-read-replica 예시에서는 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...
```

```
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 PromoteReadReplica 섹션을 참조하세요.

`purchase-reserved-db-instance`

다음 코드 예시에서는 `purchase-reserved-db-instance`의 사용 방법을 보여줍니다.

AWS CLI

예약 DB 인스턴스 오퍼링 구매

다음 `purchase-reserved-db-instances-offering` 예시에서는 정기 DB 인스턴스 제품을 구매합니다. `reserved-db-instances-offering-id`는 `describe-reserved-db-instances-offering` 명령으로 반환된 유효한 제공 ID여야 합니다.

aws rds purchase-reserved-db-instances-offering --reserved-db-instances-offering-id 438012d3-4a52-4cc7-b2e3-8dff72e0e706

- API 세부 정보는 AWS CLI 명령 참조의 PurchaseReservedDbInstance 섹션을 참조하세요.

`purchase-reserved-db-instances-offerings`

다음 코드 예시에서는 `purchase-reserved-db-instances-offerings`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 구매할 정기 DB 인스턴스 찾기

다음 `describe-reserved-db-instances-offerings` 예시에서는 db.t2.micro 인스턴스 클래스와 1년의 기간이 있는 사용 가능한 예약 MySQL DB 인스턴스를 나열합니다. 정기 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 RDS 사용 설명서의 [Amazon RDS용 정기 DB 인스턴스](#) 섹션을 참조하세요.

예시 2: 예약 DB 인스턴스 구매

다음 `purchase-reserved-db-instances-offering` 이전 예시에서 예약된 DB 인스턴스 오퍼링을 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 RDS 사용 설명서의 Amazon RDS용 정기 DB 인스턴스 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PurchaseReservedDbInstancesOfferings 섹션을 참조하세요.

reboot-db-instance

다음 코드 예시에서는 `reboot-db-instance`의 사용 방법을 보여줍니다.

AWS CLI

DB 인스턴스 재부팅

다음 `reboot-db-instance` 예시에서는 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 사용자 안내서의 DB 인스턴스 재부팅을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RebootDBInstance를 참조하세요.

reboot-db-shard-group

다음 코드 예시에서는 reboot-db-shard-group의 사용 방법을 보여줍니다.

AWS CLI

예시 1: DB 샤드 그룹 재부팅

다음 reboot-db-shard-group 예시에서는 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 재부팅 섹션을 참조하세요.

예시 2: DB 샤드 그룹 설명

다음 `describe-db-shard-groups` 예시에서는 `reboot-db-shard-group` 명령을 실행한 후 DB 샤드 그룹의 세부 정보를 검색합니다. 이제 DB 샤드 그룹 `my-db-shard-group`이 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 재부팅 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RebootDbShardGroup 섹션을 참조하세요.

`register-db-proxy-targets`

다음 코드 예시에서는 `register-db-proxy-targets`의 사용 방법을 보여줍니다.

AWS CLI

데이터베이스에 DB 프록시를 등록하는 방법

다음 `register-db-proxy-targets` 예시에서는 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 섹션 및 Amazon Aurora 사용 설명서의 RDS 프록시 생성 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RegisterDbProxyTargets 섹션을 참조하세요.

`remove-from-global-cluster`

다음 코드 예시에서는 `remove-from-global-cluster`의 사용 방법을 보여줍니다.

AWS CLI

Aurora 글로벌 데이터베이스 클러스터에서 Aurora 보조 클러스터 분리

다음 `remove-from-global-cluster` 예시에서는 Aurora 글로벌 데이터베이스 클러스터에서 Aurora 보조 클러스터를 분리하는 예시입니다. 클러스터가 읽기 전용에서 읽기-쓰기 기능을 갖춘 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 Aurora 글로벌 데이터베이스에서 클러스터 제거 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RemoveFromGlobalCluster 섹션을 참조하세요.

remove-option-from-option-group

다음 코드 예시에서는 `remove-option-from-option-group`의 사용 방법을 보여줍니다.

AWS CLI

옵션 그룹에서 옵션을 삭제하는 방법

다음 `remove-option-from-option-group` 예시에서는 `myoptiongroup`에서 `OEM` 옵션을 제거합니다.

```
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 제거 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RemoveOptionFromOptionGroup 섹션을 참조하세요.

`remove-role-from-db-cluster`

다음 코드 예시에서는 `remove-role-from-db-cluster`의 사용 방법을 보여줍니다.

AWS CLI

DB 클러스터에서 AWS Identity and Access Management(IAM) 역할 연결 해제

다음 `remove-role-from-db-cluster` 예시에서는 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 IAM 역할을 Amazon Aurora MySQL DB 클러스터와 연결 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RemoveRoleFromDbCluster 섹션을 참조하세요.

remove-role-from-db-instance

다음 코드 예시에서는 remove-role-from-db-instance의 사용 방법을 보여줍니다.

AWS CLI

DB 인스턴스에서 AWS Identity and Access Management(IAM) 역할 연결 해제

다음 remove-role-from-db-instance 예시에서는 test-instance라는 Oracle DB 인스턴스에서 rds-s3-integration-role 역할을 제거합니다.

```
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 Server 통합 비활성화 섹션을 사용하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RemoveRoleFromDbInstance 섹션을 참조하세요.

remove-source-identifier-from-subscription

다음 코드 예시에서는 remove-source-identifier-from-subscription의 사용 방법을 보여줍니다.

AWS CLI

구독에서 소스 식별자 제거

다음 remove-source-identifier 예시에서는 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
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 RemoveSourceIdentifierFromSubscription 섹션을 참조하세요.

remove-tags-from-resource

다음 코드 예시에서는 remove-tags-from-resource의 사용 방법을 보여줍니다.

AWS CLI

리소스에서 태그 제거

다음 remove-tags-from-resource 예시에서는 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 RDS 사용 설명서의 Amazon RDS 리소스에 태그 지정 섹션 및 Amazon Aurora 사용 설명서의 Amazon RDS 리소스 태그 지정 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RemoveTagsFromResource를 참조하세요.

reset-db-cluster-parameter-group

다음 코드 예시에서는 reset-db-cluster-parameter-group의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 모든 파라미터를 기본값으로 재설정

다음 reset-db-cluster-parameter-group 예시에서는 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 작업 및 DB 클러스터 파라미터 작업](#) 섹션을 참조하세요.

예시 2: 특정 파라미터를 기본값으로 재설정

다음 reset-db-cluster-parameter-group 예시에서는 고객이 생성한 DB 클러스터 파라미터 그룹에서 특정 파라미터의 파라미터 값을 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 작업 및 DB 클러스터 파라미터 작업 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ResetDbClusterParameterGroup 섹션을 참조하세요.

reset-db-parameter-group

다음 코드 예시에서는 `reset-db-parameter-group`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 모든 파라미터를 기본값으로 재설정

다음 `reset-db-parameter-group` 예시에서는 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 섹션 및 Amazon Aurora 사용 설명서의 DB 파라미터 그룹 및 DB 클러스터 파라미터 그룹 작업 섹션을 참조하세요.

예시 2: 특정 파라미터를 기본값으로 재설정

다음 `reset-db-parameter-group` 예시에서는 고객이 생성한 DB 파라미터 그룹에서 특정 파라미터의 파라미터 값을 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 섹션 및 Amazon Aurora 사용 설명서의 DB 파라미터 그룹 및 DB 클러스터 파라미터 그룹 작업 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ResetDbParameterGroup 섹션을 참조하세요.

`restore-db-cluster-from-s3`

다음 코드 예시에서는 `restore-db-cluster-from-s3`의 사용 방법을 보여줍니다.

AWS CLI

Amazon S3에서 Amazon Aurora DB 클러스터 복원

다음 `restore-db-cluster-from-s3` 예시에서는 Amazon S3의 MySQL 5.7 DB 백업 파일에서 Amazon Aurora MySQL 버전 5.7과 호환되는 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 amzn-s3-demo-bucket \
    --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 MySQL에서 데이터 마이그레이션 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RestoreDbClusterFromS3 섹션을 참조하세요.

`restore-db-cluster-from-snapshot`

다음 코드 예시에서는 `restore-db-cluster-from-snapshot`의 사용 방법을 보여줍니다.

AWS CLI

스냅샷에서 DB 클러스터 복원

다음 `restore-db-cluster-from-snapshot`은 이름이 `test-instance-snapshot`인 DB 클러스터 스냅샷에서 PostgreSQL 버전 10.7과 호환되는 Aurora PostgreSQL DB 클러스터를 복원합니다.

```
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 복원 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RestoreDbClusterFromSnapshot 섹션을 참조하세요.

`restore-db-cluster-to-point-in-time`

다음 코드 예시에서는 `restore-db-cluster-to-point-in-time`의 사용 방법을 보여줍니다.

AWS CLI

지정된 시간으로 DB 클러스터 복원

다음 `restore-db-cluster-to-point-in-time` 예시에서는 `database-4` DB 클러스터를 가능한 가장 늦은 시간으로 복원합니다. `copy-on-write` 복원 유형을 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 복원 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RestoreDbClusterToPointInTime 섹션을 참조하세요.

restore-db-instance-from-db-snapshot

다음 코드 예시에서는 `restore-db-instance-from-db-snapshot`의 사용 방법을 보여줍니다.

AWS CLI

DB 스냅샷에서 DB 인스턴스 복원

다음 `restore-db-instance-from-db-snapshot` 예시에서는 지정된 DB 스냅샷에서 `db.t3.small` DB 인스턴스 클래스를 통해 이름이 `db7-new-instance`인 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 복원 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RestoreDbInstanceFromDbSnapshot 섹션을 참조하세요.

`restore-db-instance-from-s3`

다음 코드 예시에서는 `restore-db-instance-from-s3`의 사용 방법을 보여줍니다.

AWS CLI

Amazon S3의 백업에서 DB 인스턴스를 복원하는 방법

다음 `restore-db-instance-from-s3` 예시에서는 `my-backups` S3 버킷의 기존 백업에서 이름이 `restored-test-instance`인 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 세부 정보는 AWS CLI 명령 참조의 RestoreDbInstanceFromS3 섹션을 참조하세요.

restore-db-instance-to-point-in-time

다음 코드 예시에서는 restore-db-instance-to-point-in-time의 사용 방법을 보여줍니다.

AWS CLI

예시 1: DB 인스턴스를 특정 시점으로 복원

다음 restore-db-instance-to-point-in-time 예시에서는 지정된 시간을 기준으로 test-instance를 restored-test-instance라는 새로운 DB 인스턴스로 복원합니다.

```
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 지정된 시간으로 DB 인스턴스 복원 섹션을 참조하세요.

예시 2: 복제된 백업에서 DB 인스턴스를 지정된 시간으로 복원

다음 `restore-db-instance-to-point-in-time` 예시에서는 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 복원 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RestoreDbInstanceToPointInTime 섹션을 참조하세요.

start-activity-stream

다음 코드 예시에서는 start-activity-stream의 사용 방법을 보여줍니다.

AWS CLI

데이터베이스 활동 스트림 시작

다음 start-activity-stream 예시에서는 my-pg-cluster라는 Aurora 클러스터를 모니터링하기 위해 비동기 활동 스트림을 시작합니다.

```
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 시작 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartActivityStream 섹션을 참조하세요.

start-db-cluster

다음 코드 예시에서는 start-db-cluster의 사용 방법을 보여줍니다.

AWS CLI

DB 클러스터를 시작하는 방법

다음 `start-db-cluster` 예시에서는 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 시작 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartDbCluster 섹션을 참조하세요.

`start-db-instance-automated-backups-replication`

다음 코드 예시에서는 `start-db-instance-automated-backups-replication`의 사용 방법을 보여줍니다.

AWS CLI

리전 간 자동 백업을 활성화하는 방법

다음 `start-db-instance-automated-backups-replication` 예시에서는 미국 동부(노스 버지니아) 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 활성화 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartDbInstanceAutomatedBackupsReplication 섹션을 참조하세요.

start-db-instance

다음 코드 예시에서는 start-db-instance의 사용 방법을 보여줍니다.

AWS CLI

DB 인스턴스 시작

다음 start-db-instance 예시에서는 지정된 DB 인스턴스를 시작합니다.

```
aws rds start-db-instance \
    --db-instance-identifier test-instance
```

출력:

```
{
    "DBInstance": {
        "DBInstanceStatus": "starting",
        ...some output truncated...
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 StartDbInstance 섹션을 참조하세요.

start-export-task

다음 코드 예시에서는 start-export-task의 사용 방법을 보여줍니다.

AWS CLI

Amazon S3로 스냅샷 내보내기

다음 start-export-task 예시에서는 db5-snapshot-test DB 스냅샷을 amzn-s3-demo-bucket Amazon S3 버킷으로 내보냅니다.

```
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 amzn-s3-demo-bucket \
    --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": "amzn-s3-demo-bucket",
    "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 Amazon S3 버킷으로 스냅샷 내보내기 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartExportTask 섹션을 참조하세요.

stop-activity-stream

다음 코드 예시에서는 `stop-activity-stream`의 사용 방법을 보여줍니다.

AWS CLI

데이터베이스 활동 스트림 중지

다음 `stop-activity-stream` 예시에서는 my-pg-cluster라는 이름의 Aurora 클러스터에서 활동 스트림을 중지합니다.

```
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 중지 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StopActivityStream 섹션을 참조하세요.

stop-db-cluster

다음 코드 예시에서는 `stop-db-cluster`의 사용 방법을 보여줍니다.

AWS CLI

DB 클러스터 중지

다음 `stop-db-cluster` 예시에서는 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 시작 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StopDbCluster 섹션을 참조하세요.

stop-db-instance-automated-backups-replication

다음 코드 예시에서는 `stop-db-instance-automated-backups-replication`의 사용 방법을 보여줍니다.

AWS CLI

자동 백업 복제를 중지하는 방법

다음 `stop-db-instance-automated-backups-replication`은 미국 서부(오레곤) 리전으로 자동 백업 복제를 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 중지 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StopDbInstanceAutomatedBackupsReplication 섹션을 참조하세요.

stop-db-instance

다음 코드 예시에서는 `stop-db-instance`의 사용 방법을 보여줍니다.

AWS CLI

DB 인스턴스 중지

다음 `stop-db-instance` 예시에서는 지정된 DB 인스턴스를 중지합니다.

```
aws rds stop-db-instance \
    --db-instance-identifier test-instance
```

출력:

```
{
    "DBInstance": {
        "DBInstanceStatus": "stopping",
        ...some output truncated...
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 StopDbInstance 섹션을 참조하세요.

switchover-blue-green-deployment

다음 코드 예시에서는 `switchover-blue-green-deployment`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: RDS DB 인스턴스에 대한 블루/그린 배포 전환

다음 `switchover-blue-green-deployment` 예시에서는 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 전환 섹션을 참조하세요.

예시 2: Aurora MySQL DB 클러스터에 대한 블루/그린 배포 승격

다음 `switchover-blue-green-deployment` 예시에서는 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 전환 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 SwitchoverBlueGreenDeployment 섹션을 참조하세요.

AWS CLI를 사용한 Amazon RDS 예시

다음 코드 예시에서는 Amazon RDS와 함께 AWS Command Line Interface를 사용하여 일반적인 시나리오를 구현하고 개별 서비스 작업을 수행하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

batch-execute-statement

다음 코드 예시에서는 batch-execute-statement의 사용 방법을 보여줍니다.

AWS CLI

배치 SQL 문 실행

다음 batch-execute-statement 예시에서는 파라미터 세트를 사용하여 데이터 배열에 대해 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 사용자 안내서의 Aurora Serverless에 데이터 API 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 BatchExecuteStatement를 참조하세요.

begin-transaction

다음 코드 예시에서는 begin-transaction의 사용 방법을 보여줍니다.

AWS CLI

SQL 트랜잭션 시작

다음 begin-transaction 예시에서는 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 사용자 안내서의 Aurora Serverless에 데이터 API 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 BeginTransaction을 참조하세요.

commit-transaction

다음 코드 예시에서는 `commit-transaction`의 사용 방법을 보여줍니다.

AWS CLI

SQL 트랜잭션 커밋

다음 `commit-transaction` 예시에서는 지정된 SQL 트랜잭션을 종료하고 그 일부로 적용한 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 사용자 안내서의 Aurora Serverless에 데이터 API 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CommitTransaction을 참조하세요.

execute-statement

다음 코드 예시에서는 `execute-statement`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 트랜잭션의 일부인 SQL 문 실행

다음 `execute-statement` 예시에서는 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
}
```

예 2: 파라미터를 사용하여 SQL 문을 실행하는 방법

다음 `execute-statement` 예시에서는 파라미터를 사용하여 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 사용자 안내서의 Aurora Serverless에 데이터 API 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ExecuteStatement를 참조하세요.

`rollback-transaction`

다음 코드 예시에서는 `rollback-transaction`의 사용 방법을 보여줍니다.

AWS CLI

SQL 트랜잭션 롤백

다음 `rollback-transaction` 예시에서는 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 사용자 안내서의 Aurora Serverless에 데이터 API 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RollbackTransaction을 참조하세요.

AWS CLI를 사용한 Amazon RDS Performance Insights 예시

다음 코드 예시는 Amazon RDS Performance Insights와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`create-performance-analysis-report`

다음 코드 예시는 `create-performance-analysis-report`의 사용 방법을 보여 줍니다.

AWS CLI

성능 분석 보고서를 생성하는 방법

다음 `create-performance-analysis-report` 예시에서는 데이터베이스 `db-abcdefg123456789`에 대해 시작 시간 `1682969503`과 종료 시간 `1682979503`을 사용하여 성능 분석 보고서를 생성합니다.

```
aws pi create-performance-analysis-report \
    --service-type RDS \
    --identifier db-abcdefg123456789 \
    --start-time 1682969503 \
    --end-time 1682979503
```

출력:

```
{
    "AnalysisReportId": "report-0234d3ed98e28fb17"
}
```

성능 분석 보고서 생성에 대한 자세한 내용은 Amazon RDS 사용 설명서의 성능 개선 도우미에서 성능 분석 보고서 생성 및 Amazon Aurora 사용 설명서의 성능 개선 도우미에서 성능 분석 보고서 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreatePerformanceAnalysisReport를 참조하세요.

`delete-performance-analysis-report`

다음 코드 예시는 `delete-performance-analysis-report`의 사용 방법을 보여 줍니다.

AWS CLI

성능 분석 보고서를 삭제하는 방법

다음 `delete-performance-analysis-report` 예시에서는 보고서 ID가 `report-0d99cc91c4422ee61`인 성능 분석 보고서를 삭제합니다.

```
aws pi delete-performance-analysis-report \
    --service-type RDS \
    --identifier db-abcdefg123456789 \
    --analysis-report-id report-0d99cc91c4422ee61
```

이 명령은 출력을 생성하지 않습니다.

성능 분석 보고서 삭제에 대한 자세한 내용은 Amazon RDS 사용 설명서의 성능 개선 도우미에서 성능 분석 보고서 삭제 및 Amazon Aurora 사용 설명서의 성능 개선 도우미에서 성능 분석 보고서 삭제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeletePerformanceAnalysisReport를 참조하세요.

describe-dimension-keys

다음 코드 예시는 describe-dimension-keys의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 차원 키를 설명하는 방법

이 예시에서는 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
        }
    ]
}
```

예시 2: DB 로드에 가장 큰 영향을 미치는 명령문의 SQL ID를 찾는 방법

다음 `describe-dimension-keys`에서는 DB 로드에 가장 큰 영향을 미친 10개 명령문의 SQL 문과 SQL ID를 요청합니다.

```
aws pi describe-dimension-keys \
    --service-type RDS \
    --identifier db-abcdefg123456789 \
    --start-time 2023-05-01T00:00:00Z \
    --end-time 2023-05-01T01:00:00Z \
    --metric db.load.avg \
    --group-by '{"Group": "db.sql", "Dimensions": ["db.sql.id",
 "db.sql.statement"],"Limit": 10}'
```

출력:

```
{
    "AlignedEndTime": 1.5270804E9,
    "AlignedStartTime": 1.5270264E9,
    "Identifier": "db-abcdefg123456789",
    "MetricList": [
        {
            "Keys": [
                {
                    "Dimensions": {"db.sql.id": "AKIAIOSFODNN7EXAMPLE",
 "db.sql.statement": "SELECT * FROM customers WHERE customer_id = 123"},
                    "Total": 25.5,"Partitions": [12.3, 13.2]
                }
            ]
        }
    ]
}
```

Performance Insights의 차원에 대한 자세한 내용은 Amazon RDS 사용 설명서의 데이터베이스 로드 및 Amazon Aurora 사용 설명서의 데이터베이스 로드를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDimensionKeys를 참조하세요.

`get-dimension-key-details`

다음 코드 예시는 `get-dimension-key-details`의 사용 방법을 보여 줍니다.

AWS CLI

DB 인스턴스의 지정된 차원 그룹에 대한 세부 정보를 가져오는 방법

다음 `get-dimension-key-details` 예시에서는 DB 인스턴스 `db-10BCD2EFGHIJ3KL4M5NO6PQRS5`에 대한 SQL 문의 전체 텍스트를 검색합니다. `--group`은 `db.sql`이고 `--group-identifier`는 `db.sql.id`입니다. 이 예시에서 `example-sql-id`는 `get-resource-metrics` 또는 `describe-dimension-keys` 작업을 사용하여 검색된 SQL ID를 나타냅니다. 이 예제에서는 차원 세부 정보를 사용할 수 있습니다. 따라서 성능 개선 도우미는 잘리지 않은 SQL 문의 전체 텍스트를 검색합니다.

```
aws pi get-dimension-key-details \
    --service-type RDS \
    --identifier db-10BCD2EFGHIJ3KL4M5NO6PQRS5 \
    --group db.sql \
    --group-identifier example-sql-id \
    --requested-dimensions statement
```

출력:

```
{
    "Dimensions":[
        {
            "Value": "SELECT e.last_name, d.department_name FROM employees e,
 departments d WHERE e.department_id=d.department_id",
            "Dimension": "db.sql.statement",
            "Status": "AVAILABLE"
        },
    ...
    ]
}
```

Performance Insights의 차원에 대한 자세한 내용은 Amazon RDS 사용 설명서의 데이터베이스 로드 및 Amazon Aurora 사용 설명서의 데이터베이스 로드를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetDimensionKeyDetails를 참조하세요.

get-performance-analysis-report

다음 코드 예시는 `get-performance-analysis-report`의 사용 방법을 보여 줍니다.

AWS CLI

성능 분석 보고서를 가져오는 방법

다음 `get-performance-analysis-report` 예시에서는 보고서 ID가 `report-0d99cc91c4422ee61`인 데이터베이스 `db-abcdefg123456789`에 대한 성능 분석 보고서를 가져옵니다. 응답은 보고서 상태, ID, 시간 세부 정보, 인사이트를 제공합니다.

```
aws pi get-performance-analysis-report \
    --service-type RDS \
    --identifier db-abcdefg123456789 \
    --analysis-report-id report-0d99cc91c4422ee61
```

출력:

```
{
    "AnalysisReport": {
        "Status": "Succeeded",
        "ServiceType": "RDS",
        "Identifier": "db-abcdefg123456789",
        "StartTime": 1680583486.584,
        "AnalysisReportId": "report-0d99cc91c4422ee61",
        "EndTime": 1680587086.584,
        "CreateTime": 1680587087.139,
        "Insights": [
            ... (Condensed for space)
        ]
    }
}
```

성능 분석 보고서에 대한 자세한 내용은 Amazon RDS 사용 설명서의 일정 기간 동안의 데이터베이스 성능 분석 및 Amazon Aurora 사용 설명서의 일정 기간 동안의 데이터베이스 성능 분석을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetPerformanceAnalysisReport를 참조하세요.

`get-resource-metadata`

다음 코드 예시는 `get-resource-metadata`의 사용 방법을 보여 줍니다.

AWS CLI

데이터베이스의 리소스 메타데이터를 가져오는 방법

다음 `get-resource-metadata` 예시에서는 데이터베이스 db-abcdefg123456789에 대한 리소스 메타데이터를 가져옵니다. 이 응답은 SQL 다이제스트 통계가 활성화되었음을 보여줍니다.

```
aws pi get-resource-metadata \
    --service-type RDS \
    --identifier db-abcdefg123456789
```

출력:

```
{
    "Identifier": "db-abcdefg123456789",
    "Features":{
        "SQL_DIGEST_STATISTICS":{
            "Status": "ENABLED"
        }
    }
}
```

Performance Insights의 SQL 통계에 대한 자세한 내용은 Amazon RDS 사용 설명서의 성능 개선 도우미에 대한 SQL 통계 및 Amazon Aurora 사용 설명서의 성능 개선 도우미에 대한 SQL 통계를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetResourceMetadata를 참조하세요.

get-resource-metrics

다음 코드 예시는 `get-resource-metrics`의 사용 방법을 보여 줍니다.

AWS CLI

리소스 지표 가져오기

이 예시에서는 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
```

`--metric-queries`의 인수는 `metric-queries.json` JSON 파일에 저장됩니다. 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...>
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetResourceMetrics를 참조하세요.

list-available-resource-dimensions

다음 코드 예시는 list-available-resource-dimensions의 사용 방법을 보여 줍니다.

AWS CLI

DB 인스턴스에서 지표 유형에 대해 쿼리할 수 있는 차원을 나열하는 방법

다음 list-available-resource-dimensions 예시에서는 데이터베이스 db-abcdefg123456789에 대해 쿼리할 수 있는 db.load 지표를 나열합니다.

```
aws pi list-available-resource-dimensions \
    --service-type RDS \
    --identifier db-abcdefg123456789 \
    --metrics db.load
```

출력:

```
{
```

```
    "MetricDimensions": [
        {
            "Metric": "db.load",
            "Groups": [
                {
                    "Group": "db.user",
                    "Dimensions": [
                        {
                            "Identifier": "db.user.id"
                        },
                        {
                            "Identifier": "db.user.name"
                        }
                    ]
                },
                {
                    "Group": "db.sql_tokenized",
                    "Dimensions": [
                        {
                            "Identifier": "db.sql_tokenized.id"
                        },
                        {
                            "Identifier": "db.sql_tokenized.db_id"
                        },
                        {
                            "Identifier": "db.sql_tokenized.statement"
                        }
                    ]
                },
                ...
            ]
        }
    ]
}
```

Performance Insights의 차원에 대한 자세한 내용은 Amazon RDS 사용 설명서의 데이터베이스 로드 및 Amazon Aurora 사용 설명서의 데이터베이스 로드를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListAvailableResourceDimensions를 참조하세요.

`list-available-resource-metrics`

다음 코드 예시는 `list-available-resource-metrics`의 사용 방법을 보여 줍니다.

AWS CLI

DB 인스턴스에서 지표 유형에 대해 쿼리할 수 있는 지표를 나열하는 방법

다음 `list-available-resource-metrics` 예시에서는 데이터베이스 db-abcdefg123456789에 대해 쿼리할 수 있는 `db.load` 지표를 나열합니다.

```
aws pi list-available-resource-metrics \
    --service-type RDS \
    --identifier db-abcdefg123456789 \
    --metric-types "os" "db"
```

출력:

```
{
    "Metrics": [
        {
            "Description": "The number of virtual CPUs for the DB instance",
            "Metric": "os.general.numVCPUs",
            "Unit": "vCPUs"
        },
        ......,
        {
            "Description": "Time spent reading data file blocks by backends in this
 instance",
            "Metric": "db.IO.read_latency",
            "Unit": "Milliseconds per block"
        },
        ......
    ]
}
```

Performance Insights의 지표에 대한 자세한 내용은 Amazon RDS 사용 설명서의 데이터베이스 로드 및 Amazon Aurora 사용 설명서의 데이터베이스 로드를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListAvailableResourceMetrics를 참조하세요.

`list-performance-analysis-reports`

다음 코드 예시는 `list-performance-analysis-reports`의 사용 방법을 보여 줍니다.

AWS CLI

데이터베이스에 대한 성능 분석 보고서를 나열하는 방법

다음 `list-performance-analysis-reports` 예시에서는 데이터베이스 db-abcdefg123456789에 대한 성능 분석 보고서를 나열합니다. 응답에는 보고서 ID, 상태 및 기간 세부 정보와 함께 모든 보고서가 나열됩니다.

```
aws pi list-performance-analysis-reports \
    --service-type RDS \
    --identifier db-abcdefg123456789
```

출력:


```
{
    "AnalysisReports": [
        {
            "Status": "Succeeded",
            "EndTime": 1680587086.584,
            "CreateTime": 1680587087.139,
            "StartTime": 1680583486.584,
            "AnalysisReportId": "report-0d99cc91c4422ee61"
        },
        {
            "Status": "Succeeded",
            "EndTime": 1681491137.914,
            "CreateTime": 1681491145.973,
            "StartTime": 1681487537.914,
            "AnalysisReportId": "report-002633115cc002233"
        },
        {
            "Status": "Succeeded",
            "EndTime": 1681493499.849,
            "CreateTime": 1681493507.762,
            "StartTime": 1681489899.849,
            "AnalysisReportId": "report-043b1e006b47246f9"
        },
        {
            "Status": "InProgress",
            "EndTime": 1682979503.0,
            "CreateTime": 1682979618.994,
            "StartTime": 1682969503.0,
            "AnalysisReportId": "report-01ad15f9b88bcbd56"
```

```
        }
    ]
}
```

성능 분석 보고서에 대한 자세한 내용은 Amazon RDS 사용 설명서의 [일정 기간 동안의 데이터베이스 성능 분석](#) 및 Amazon Aurora 사용 설명서의 [일정 기간 동안의 데이터베이스 성능 분석](#)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListPerformanceAnalysisReports를 참조하세요.

`list-tags-for-resource`

다음 코드 예시는 `list-tags-for-resource`의 사용 방법을 보여 줍니다.

AWS CLI

성능 분석 보고서의 태그를 나열하는 방법

다음 `list-tags-for-resource` 예시에서는 보고서 ID가 `report-0d99cc91c4422ee61`인 성능 분석 보고서의 태그를 나열합니다.

```
aws pi list-tags-for-resource \
    --service-type RDS \
    --resource-arn arn:aws:pi:us-west-2:123456789012:perf-reports/RDS/db-
abcdefg123456789/report-0d99cc91c4422ee61
```

출력:

```
{
    "Tags": [
        {
            "Value": "test-tag",
            "Key": "name"
        }
    ]
}
```

성능 분석 보고서 태그 지정에 대한 자세한 내용은 Amazon RDS 사용 설명서의 성능 개선 도우미에서 성능 분석 보고서에 태그 추가 및 Amazon Aurora 사용 설명서의 성능 개선 도우미에서 성능 분석 보고서에 태그 추가를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource를 참조하세요.

tag-resource

다음 코드 예시는 tag-resource의 사용 방법을 보여 줍니다.

AWS CLI

성능 분석 보고서에 태그를 추가하는 방법

다음 tag-resource 예시에서는 태그 값이 test-tag인 태그 키 name을 보고서 ID가 report-0d99cc91c4422ee61인 성능 분석 보고서에 추가합니다.

```
aws pi tag-resource \
    --service-type RDS \
    --resource-arn arn:aws:pi:us-west-2:123456789012:perf-reports/RDS/db-
abcdefg123456789/report-0d99cc91c4422ee61 \
    --tags Key=name,Value=test-tag
```

이 명령은 출력을 생성하지 않습니다.

성능 분석 보고서 태그 지정에 대한 자세한 내용은 Amazon RDS 사용 설명서의 성능 개선 도우미에서 성능 분석 보고서에 태그 추가 및 Amazon Aurora 사용 설명서의 성능 개선 도우미에서 성능 분석 보고서에 태그 추가를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

untag-resource

다음 코드 예시는 untag-resource의 사용 방법을 보여 줍니다.

AWS CLI

성능 분석 보고서의 태그를 삭제하는 방법

다음 untag-resource 예시에서는 보고서 ID가 report-0d99cc91c4422ee61인 성능 분석 보고서의 태그 name을 삭제합니다.

```
aws pi untag-resource \
    --service-type RDS \
    --resource-arn arn:aws:pi:us-west-2:123456789012:perf-reports/RDS/db-
abcdefg123456789/report-0d99cc91c4422ee61 \
    --tag-keys name
```

이 명령은 출력을 생성하지 않습니다.

성능 분석 보고서 태그 지정에 대한 자세한 내용은 Amazon RDS 사용 설명서의 성능 개선 도우미에서 성능 분석 보고서에 태그 추가 및 Amazon Aurora 사용 설명서의 성능 개선 도우미에서 성능 분석 보고서에 태그 추가를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

AWS CLI를 사용한 Amazon Redshift 예시

다음 코드 예시는 Amazon Redshift와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`accept-reserved-node-exchange`

다음 코드 예시에서는 `accept-reserved-node-exchange`의 사용 방법을 보여줍니다.

AWS CLI

예약 노드 교환 수락

다음 `accept-reserved-node-exchange` 예시에서는 DC1 예약 노드를 DC2 예약 노드로 교환하는 것을 수락합니다.

```
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 AWS CLI를 사용하여 예약 노드 업그레이드를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AcceptReservedNodeExchange를 참조하세요.

authorize-cluster-security-group-ingress

다음 코드 예시에서는 `authorize-cluster-security-group-ingress`의 사용 방법을 보여줍니다.

AWS CLI

EC2 보안 그룹에 대한 액세스 권한 부여 이 예시에서는 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 범위에 대한 액세스 권한 부여 이 예시에서는 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
```

- API 세부 정보는 AWS CLI 명령 참조의 AuthorizeClusterSecurityGroupIngress를 참조하세요.

authorize-snapshot-access

다음 코드 예시에서는 authorize-snapshot-access의 사용 방법을 보여줍니다.

AWS CLI

AWS 계정에 스냅샷 복원 권한 부여 이 예시에서는 AWS 계정 444455556666에 my-snapshot-id 스냅샷을 복원할 수 있는 권한을 부여합니다. 기본 출력 형식은 JSON입니다. 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 AuthorizeSnapshotAccess를 참조하세요.

batch-delete-cluster-snapshots

다음 코드 예시에서는 batch-delete-cluster-snapshots의 사용 방법을 보여줍니다.

AWS CLI

클러스터 스냅샷 세트 삭제

다음 batch-delete-cluster-snapshots 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 BatchDeleteClusterSnapshots를 참조하세요.

batch-modify-cluster-snapshots

다음 코드 예시에서는 batch-modify-cluster-snapshots의 사용 방법을 보여줍니다.

AWS CLI

클러스터 스냅샷 세트 수정

다음 `batch-modify-cluster-snapshots` 예시에서는 클러스터 스냅샷 세트의 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.

- API 세부 정보는 AWS CLI 명령 참조의 BatchModifyClusterSnapshots를 참조하세요.

`cancel-resize`

다음 코드 예시에서는 `cancel-resize`의 사용 방법을 보여줍니다.

AWS CLI

클러스터 크기 조정 취소

다음 cancel-resize 예시에서는 클러스터에 대한 기존 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.

- API 세부 정보는 AWS CLI 명령 참조의 CancelResize를 참조하세요.

copy-cluster-snapshot

다음 코드 예시에서는 copy-cluster-snapshot의 사용 방법을 보여줍니다.

AWS CLI

모든 클러스터 버전에 대한 설명 가져오기 이 예시에서는 모든 클러스터 버전에 대한 설명을 반환합니다. 기본 출력 형식은 JSON입니다. 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CopyClusterSnapshot을 참조하세요.

`create-cluster-parameter-group`

다음 코드 예시에서는 `create-cluster-parameter-group`의 사용 방법을 보여줍니다.

AWS CLI

클러스터 파라미터 그룹 생성 이 예시에서는 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateClusterParameterGroup을 참조하세요.

create-cluster-security-group

다음 코드 예시에서는 create-cluster-security-group의 사용 방법을 보여줍니다.

AWS CLI

클러스터 보안 그룹 생성 이 예시에서는 새 클러스터 보안 그룹을 생성합니다. 기본 출력 형식은 JSON입니다. 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
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateClusterSecurityGroup을 참조하세요.

`create-cluster-snapshot`

다음 코드 예시에서는 `create-cluster-snapshot`의 사용 방법을 보여줍니다.

AWS CLI

클러스터 스냅샷 생성 이 예시에서는 새 클러스터 스냅샷을 생성합니다. 기본 출력 형식은 JSON입니다. 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateClusterSnapshot을 참조하세요.

`create-cluster-subnet-group`

다음 코드 예시에서는 `create-cluster-subnet-group`의 사용 방법을 보여줍니다.

AWS CLI

클러스터 서브넷 그룹 생성 이 예시에서는 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateClusterSubnetGroup을 참조하세요.

`create-cluster`

다음 코드 예시에서는 `create-cluster`의 사용 방법을 보여줍니다.

AWS CLI

최소 파라미터를 사용하여 클러스터 생성 이 예시에서는 최소 파라미터 세트로 클러스터를 만듭니다. 기본 출력 형식은 JSON입니다. 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateCluster를 참조하세요.

`create-event-subscription`

다음 코드 예시에서는 `create-event-subscription`의 사용 방법을 보여줍니다.

AWS CLI

이벤트 알림 구독 생성

다음 `create-event-subscription` 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateEventSubscription을 참조하세요.

create-hsm-client-certificate

다음 코드 예시에서는 `create-hsm-client-certificate`의 사용 방법을 보여줍니다.

AWS CLI

HSM 클라이언트 인증서 생성

다음 `create-hsm-client-certificate` 예시에서는 클러스터가 HSM에 연결하는 데 사용할 수 있는 HSM 클라이언트 인증서를 생성합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateHsmClientCertificate를 참조하세요.

`create-hsm-configuration`

다음 코드 예시에서는 `create-hsm-configuration`의 사용 방법을 보여줍니다.

AWS CLI

HSM 구성 생성

다음 `create-hsm-configuration` 예시에서는 클러스터가 하드웨어 보안 모듈(HSM)에 데이터베이스 암호화 키를 저장하고 사용하는 데 필요한 정보가 포함된 지정된 HSM 구성을 생성합니다.

```
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": []
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateHsmConfiguration을 참조하세요.

`create-snapshot-copy-grant`

다음 코드 예시에서는 `create-snapshot-copy-grant`의 사용 방법을 보여줍니다.

AWS CLI

스냅샷 복사 권한 생성

다음 `create-snapshot-copy-grant` 예시에서는 스냅샷 복사 권한을 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 암호화](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [CreateSnapshotCopyGrant](#)를 참조하세요.

`create-snapshot-schedule`

다음 코드 예시에서는 `create-snapshot-schedule`의 사용 방법을 보여줍니다.

AWS CLI

스냅샷 일정 생성

다음 `create-snapshot-schedule` 예시에서는 지정된 설명을 포함하여 12시간마다 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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateSnapshotSchedule을 참조하세요.

create-tags

다음 코드 예시에서는 `create-tags`의 사용 방법을 보여줍니다.

AWS CLI

클러스터의 태그 생성

다음 `create-tags` 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateTags를 참조하세요.

delete-cluster-parameter-group

다음 코드 예시에서는 `delete-cluster-parameter-group`의 사용 방법을 보여줍니다.

AWS CLI

클러스터 파라미터 그룹 삭제 이 예시에서는 새 클러스터 파라미터 그룹을 삭제합니다. 명령:

```
aws redshift delete-cluster-parameter-group --parameter-group-name
 myclusterparametergroup
```

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteClusterParameterGroup](#)을 참조하세요.

`delete-cluster-security-group`

다음 코드 예시에서는 `delete-cluster-security-group`의 사용 방법을 보여줍니다.

AWS CLI

클러스터 보안 그룹 삭제 이 예시에서는 클러스터 보안 그룹을 삭제합니다. 명령:

```
aws redshift delete-cluster-security-group --cluster-security-group-name
 mysecuritygroup
```

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteClusterSecurityGroup](#)을 참조하세요.

`delete-cluster-snapshot`

다음 코드 예시에서는 `delete-cluster-snapshot`의 사용 방법을 보여줍니다.

AWS CLI

클러스터 스냅샷 삭제 이 예시에서는 클러스터 스냅샷을 삭제합니다. 명령:

```
aws redshift delete-cluster-snapshot --snapshot-identifier my-snapshot-id
```

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteClusterSnapshot](#)을 참조하세요.

`delete-cluster-subnet-group`

다음 코드 예시에서는 `delete-cluster-subnet-group`의 사용 방법을 보여줍니다.

AWS CLI

클러스터 서브넷 그룹 삭제 이 예시에서는 클러스터 서브넷 그룹을 삭제합니다. 명령:

```
aws redshift delete-cluster-subnet-group --cluster-subnet-group-name mysubnetgroup
```

결과:

```
{
   "ResponseMetadata": {
      "RequestId": "253fbffd-6993-11e2-bc3a-47431073908a"
   }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteClusterSubnetGroup을 참조하세요.

delete-cluster

다음 코드 예시에서는 delete-cluster의 사용 방법을 보여줍니다.

AWS CLI

클러스터 삭제 및 최종 클러스터 스냅샷 생성 방지 이 예시에서는 클러스터를 삭제하고 최종 클러스터 스냅샷이 생성되지 않도록 데이터를 강제로 삭제합니다. 명령:

```
aws redshift delete-cluster --cluster-identifier mycluster --skip-final-cluster-snapshot
```

클러스터 삭제, 최종 클러스터 스냅샷 허용 이 예시에서는 클러스터를 삭제하지만 최종 클러스터 스냅샷을 지정합니다. 명령:

```
aws redshift delete-cluster --cluster-identifier mycluster --final-cluster-snapshot-identifier myfinalsnapshot
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteCluster를 참조하세요.

delete-event-subscription

다음 코드 예시에서는 delete-event-subscription의 사용 방법을 보여줍니다.

AWS CLI

이벤트 구독 삭제

다음 delete-event-subscription 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteEventSubscription을 참조하세요.

`delete-hsm-client-certificate`

다음 코드 예시에서는 `delete-hsm-client-certificate`의 사용 방법을 보여줍니다.

AWS CLI

HSM 클라이언트 인증서 삭제

다음 `delete-hsm-client-certificate` 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteHsmClientCertificate를 참조하세요.

`delete-hsm-configuration`

다음 코드 예시에서는 `delete-hsm-configuration`의 사용 방법을 보여줍니다.

AWS CLI

HSM 구성 삭제

다음 `delete-hsm-configuration` 예시에서는 현재 AWS 계정에서 지정된 HSM 구성을 삭제합니다.

```
aws redshift delete-hsm-configuration /
    --hsm-configuration-identifier myhsmconnection
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteHsmConfiguration을 참조하세요.

delete-scheduled-action

다음 코드 예시에서는 delete-scheduled-action의 사용 방법을 보여줍니다.

AWS CLI

예약된 작업 삭제

다음 delete-scheduled-action 예시에서는 지정된 예약된 작업을 삭제합니다.

```
aws redshift delete-scheduled-action \
    --scheduled-action-name myscheduledaction
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteScheduledAction을 참조하세요.

delete-snapshot-copy-grant

다음 코드 예시에서는 delete-snapshot-copy-grant의 사용 방법을 보여줍니다.

AWS CLI

스냅샷 복사본 권한 삭제

다음 delete-snapshot-copy-grant 예시에서는 지정된 스냅샷 복사 권한을 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.

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteSnapshotCopyGrant]를 참조하세요.

delete-snapshot-schedule

다음 코드 예시에서는 delete-snapshot-schedule의 사용 방법을 보여줍니다.

AWS CLI

스냅샷 일정 삭제

다음 delete-snapshot-schedule 예시에서는 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 일정]을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteSnapshotSchedule]을 참조하세요.

delete-tags

다음 코드 예시에서는 delete-tags의 사용 방법을 보여줍니다.

AWS CLI

클러스터에서 태그 삭제

다음 delete-tags 예시에서는 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 지정]을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteTags를 참조하세요.

describe-account-attributes

다음 코드 예시에서는 describe-account-attributes의 사용 방법을 보여줍니다.

AWS CLI

AWS 계정의 속성 설명

다음 describe-account-attributes 예시에서는 호출 AWS 계정에 연결된 속성을 표시합니다.

```
aws redshift describe-account-attributes
```

출력:

```
{
    "AccountAttributes": [
        {
            "AttributeName": "max-defer-maintenance-duration",
            "AttributeValues": [
                {
                    "AttributeValue": "45"
                }
            ]
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeAccountAttributes를 참조하세요.

describe-cluster-db-revisions

다음 코드 예시에서는 describe-cluster-db-revisions의 사용 방법을 보여줍니다.

AWS CLI

클러스터의 DB 개정 설명

다음 describe-cluster-db-revisions 예시에서는 지정된 클러스터의 ClusterDbRevision 객체 배열에 대한 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": []
        }
    ]
}
```


- API 세부 정보는 AWS CLI 명령 참조의 DescribeClusterDbRevisions를 참조하세요.

`describe-cluster-parameter-groups`

다음 코드 예시에서는 `describe-cluster-parameter-groups`의 사용 방법을 보여줍니다.

AWS CLI

모든 클러스터 파라미터 그룹의 설명 가져오기 이 예시에서는 계정의 모든 클러스터 파라미터 그룹에 대한 설명을 열 헤더와 함께 반환합니다. 기본 출력 형식은 JSON입니다. 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
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeClusterParameterGroups를 참조하세요.

describe-cluster-parameters

다음 코드 예시에서는 describe-cluster-parameters의 사용 방법을 보여줍니다.

AWS CLI

지정된 클러스터 파라미터 그룹의 파라미터 가져오기 이 예시에서는 명명된 파라미터 그룹의 파라미터를 가져옵니다. 기본 출력 형식은 JSON입니다. 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
```

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeClusterParameters](https://docs.aws.amazon.com/goto/aws-cli/redshift-2012-12-01/DescribeClusterParameters)를 참조하세요.

`describe-cluster-security-groups`

다음 코드 예시에서는 `describe-cluster-security-groups`의 사용 방법을 보여줍니다.

AWS CLI

모든 클러스터 보안 그룹의 설명 가져오기 이 예시에서는 계정의 모든 클러스터 보안 그룹에 대한 설명을 반환합니다. 기본 출력 형식은 JSON입니다. 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...)
   ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeClusterSecurityGroups](https://docs.aws.amazon.com/goto/aws-cli/redshift-2012-12-01/DescribeClusterSecurityGroups)를 참조하세요.

`describe-cluster-snapshots`

다음 코드 예시에서는 `describe-cluster-snapshots`의 사용 방법을 보여줍니다.

AWS CLI

모든 클러스터 스냅샷의 설명 가져오기 이 예시에서는 계정의 모든 클러스터 스냅샷에 대한 설명을 반환합니다. 기본 출력 형식은 JSON입니다. 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...)
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeClusterSnapshots를 참조하세요.

`describe-cluster-subnet-groups`

다음 코드 예시에서는 `describe-cluster-subnet-groups`의 사용 방법을 보여줍니다.

AWS CLI

모든 클러스터 서브넷 그룹의 설명 가져오기 이 예시에서는 모든 클러스터 서브넷 그룹의 설명을 반환합니다. 기본 출력 형식은 JSON입니다. 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeClusterSubnetGroups를 참조하세요.

describe-cluster-tracks

다음 코드 예시에서는 describe-cluster-tracks의 사용 방법을 보여줍니다.

AWS CLI

클러스터 트랙 설명

다음 describe-cluster-tracks 예시에서는 사용 가능한 유지 관리 트랙의 세부 정보를 표시합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeClusterTracks를 참조하세요.

describe-cluster-versions

다음 코드 예시에서는 `describe-cluster-versions`의 사용 방법을 보여줍니다.

AWS CLI

모든 클러스터 버전에 대한 설명 가져오기 이 예시에서는 모든 클러스터 버전에 대한 설명을 반환합니다. 기본 출력 형식은 JSON입니다. 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"
   }
}
```


- API 세부 정보는 AWS CLI 명령 참조의 DescribeClusterVersions를 참조하세요.

describe-clusters

다음 코드 예시에서는 describe-clusters의 사용 방법을 보여줍니다.

AWS CLI

모든 클러스터의 설명 가져오기 이 예시에서는 계정의 모든 클러스터에 대한 설명을 반환합니다. 기본 출력 형식은 JSON입니다. 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
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeClusters를 참조하세요.

describe-default-cluster-parameters

다음 코드 예시에서는 `describe-default-cluster-parameters`의 사용 방법을 보여줍니다.

AWS CLI

기본 클러스터 파라미터의 설명 가져오기 이 예시에서는 `redshift-1.0` 패밀리의 기본 클러스터 파라미터에 대한 설명을 반환합니다. 기본 출력 형식은 JSON입니다. 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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDefaultClusterParameters를 참조하세요.

describe-event-categories

다음 코드 예시에서는 describe-event-categories의 사용 방법을 보여줍니다.

AWS CLI

클러스터의 이벤트 범주 설명

다음 describe-event-categories 예시에서는 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"
                }
            ]
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeEventCategories를 참조하세요.

describe-event-subscriptions

다음 코드 예시에서는 describe-event-subscriptions 코드를 사용하는 방법을 보여줍니다.

AWS CLI

이벤트 구독 설명

다음 describe-event-subscriptions 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeEventSubscriptions를 참조하세요.

`describe-events`

다음 코드 예시에서는 `describe-events`의 사용 방법을 보여줍니다.

AWS CLI

모든 이벤트 설명 이 예시는 모든 이벤트를 반환합니다. 기본 출력 형식은 JSON입니다. 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
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeEvents를 참조하세요.

`describe-hsm-client-certificates`

다음 코드 예시에서는 `describe-hsm-client-certificates`의 사용 방법을 보여줍니다.

AWS CLI

HSM 클라이언트 인증서 설명

다음 `describe-hsm-client-certificates` 예시에서는 지정된 HSM 클라이언트 인증서의 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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeHsmClientCertificates를 참조하세요.

describe-hsm-configurations

다음 코드 예시에서는 `describe-hsm-configurations`의 사용 방법을 보여줍니다.

AWS CLI

HSM 구성 설명

다음 `describe-hsm-configurations` 예시에서는 호출 AWS 계정에 사용할 수 있는 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": []
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeHsmConfigurations를 참조하세요.

describe-logging-status

다음 코드 예시에서는 describe-logging-status의 사용 방법을 보여줍니다.

AWS CLI

클러스터의 로깅 상태 설명

다음 describe-logging-status 예시에서는 클러스터에 대한 쿼리 및 연결 시도 등의 정보가 기록되고 있는지 여부를 표시합니다.

```
aws redshift describe-logging-status \
    --cluster-identifier mycluster
```

출력:

```
{
    "LoggingEnabled": false
}
```

자세한 내용은 Amazon Redshift 클러스터 관리 안내서의 데이터베이스 감사 로깅을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeLoggingStatus를 참조하세요.

describe-node-configuration-options

다음 코드 예시에서는 describe-node-configuration-options의 사용 방법을 보여줍니다.

AWS CLI

노드 구성 옵션 설명

다음 `describe-node-configuration-options` 예시에서는 지정된 클러스터 스냅샷의 노드 유형, 노드 수 및 디스크 사용량 등 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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeNodeConfigurationOptions를 참조하세요.

`describe-orderable-cluster-options`

다음 코드 예시에서는 `describe-orderable-cluster-options`의 사용 방법을 보여줍니다.

AWS CLI

주문 가능한 모든 클러스터 옵션 설명 이 예시에서는 주문 가능한 모든 클러스터 옵션의 설명을 반환합니다. 기본 출력 형식은 JSON입니다. 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
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeOrderableClusterOptions를 참조하세요.

`describe-reserved-node-offerings`

다음 코드 예시에서는 `describe-reserved-node-offerings`의 사용 방법을 보여줍니다.

AWS CLI

예약 노드 오퍼링 설명 이 예시에서는 구매할 수 있는 모든 예약 노드 오퍼링을 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"
   }
}
```

예약 노드 제품을 구매하려면 유효한 ReservedNodeOfferingId를 사용하여 `purchase-reserved-node-offering`을 직접적으로 호출할 수 있습니다.

유효한 ReservedNodeOfferingId를 사용하는 `purchase-reserved-node-offering`

유효한 ReservedNodeOfferingId를 사용

ReservedNodeOfferingId

.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeReservedNodeOfferings를 참조하세요.

describe-reserved-nodes

다음 코드 예시에서는 describe-reserved-nodes의 사용 방법을 보여줍니다.

AWS CLI

예약 노드 설명 이 예시에서는 구매한 예약 노드 오퍼링을 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"
      }
   ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeReservedNodes를 참조하세요.

describe-resize

다음 코드 예시에서는 describe-resize의 사용 방법을 보여줍니다.

AWS CLI

크기 조정 설명 이 예시에서는 클러스터의 최근 크기 조정을 설명합니다. 요청은 dw.hs1.8xlarge 유형의 노드 3개에 대한 것이었습니다. 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeResize를 참조하세요.

describe-scheduled-actions

다음 코드 예시에서는 describe-scheduled-actions의 사용 방법을 보여줍니다.

AWS CLI

예약된 작업 설명

다음 describe-scheduled-actions 예시에서는 현재 예약된 작업의 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"
            ]
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeScheduledActions를 참조하세요.

`describe-snapshot-copy-grants`

다음 코드 예시에서는 `describe-snapshot-copy-grants`의 사용 방법을 보여줍니다.

AWS CLI

스냅샷 복사본 권한 설명

다음 `describe-snapshot-copy-grants` 예시에서는 지정된 클러스터 스냅샷 복사본 권한의 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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeSnapshotCopyGrants를 참조하세요.

describe-snapshot-schedules

다음 코드 예시에서는 `describe-snapshot-schedules`의 사용 방법을 보여줍니다.

AWS CLI

스냅샷 일정 설명

다음 `describe-snapshot-schedules` 예시에서는 지정된 클러스터 스냅샷 일정의 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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeSnapshotSchedules를 참조하세요.

describe-storage

다음 코드 예시에서는 describe-storage의 사용 방법을 보여줍니다.

AWS CLI

스토리지 설명

다음 describe-storage 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeStorage를 참조하세요.

describe-table-restore-status

다음 코드 예시에서는 describe-table-restore-status의 사용 방법을 보여줍니다.

AWS CLI

클러스터 스냅샷에서 테이블 복원 요청의 상태 설명

다음 `describe-table-restore-status` 예시에서는 지정된 클러스터에 대한 테이블 복원 요청의 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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeTableRestoreStatus를 참조하세요.

describe-tags

다음 코드 예시에서는 `describe-tags`의 사용 방법을 보여줍니다.

AWS CLI

태그 설명

다음 describe-tags 예시에서는 지정된 태그 이름 및 값에 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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeTags를 참조하세요.

disable-snapshot-copy

다음 코드 예시에서는 disable-snapshot-copy의 사용 방법을 보여줍니다.

AWS CLI

클러스터의 스냅샷 복사본 비활성화

다음 disable-snapshot-copy 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 DisableSnapshotCopy를 참조하세요.

enable-snapshot-copy

다음 코드 예시에서는 enable-snapshot-copy의 사용 방법을 보여줍니다.

AWS CLI

클러스터의 스냅샷 복사 활성화

다음 enable-snapshot-copy 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 EnableSnapshotCopy를 참조하세요.

get-cluster-credentials

다음 코드 예시에서는 `get-cluster-credentials`의 사용 방법을 보여줍니다.

AWS CLI

AWS 계정의 클러스터 자격 증명 가져오기

다음 `get-cluster-credentials` 예시에서는 Amazon Redshift 데이터베이스에 액세스할 수 있는 임시 자격 증명을 가져옵니다.

```
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 Redshift 클러스터 관리 안내서의 Amazon Redshift CLI 또는 API를 사용하여 IAM 데이터베이스 자격 증명 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetClusterCredentials를 참조하세요.

get-reserved-node-exchange-offerings

다음 코드 예시에서는 `get-reserved-node-exchange-offerings`의 사용 방법을 보여줍니다.

AWS CLI

예약 노드 교환 오퍼링 가져오기

다음 `get-reserved-node-exchange-offerings` 예시에서는 지정된 DC1 예약 노드와 일치하는 DC2 `ReservedNodeOfferings`의 배열을 가져옵니다.

```
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 AWS CLI를 사용하여 예약 노드 업그레이드를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetReservedNodeExchangeOfferings를 참조하세요.

modify-cluster-iam-roles

다음 코드 예시에서는 modify-cluster-iam-roles의 사용 방법을 보여줍니다.

AWS CLI

클러스터의 IAM 역할 수정

다음 modify-cluster-iam-roles 예시에서는 지정된 클러스터에서 지정된 AWS IAM 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 ID 기반 정책(IAM 정책) 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyClusterIamRoles를 참조하세요.

modify-cluster-maintenance

다음 코드 예시에서는 `modify-cluster-maintenance`의 사용 방법을 보여줍니다.

AWS CLI

클러스터 유지 관리 수정

다음 `modify-cluster-maintenance` 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyClusterMaintenance를 참조하세요.

`modify-cluster-parameter-group`

다음 코드 예시에서는 `modify-cluster-parameter-group`의 사용 방법을 보여줍니다.

AWS CLI

파라미터 그룹의 파라미터 수정

다음 `modify-cluster-parameter-group` 예시에서는 워크로드 관리를 위한 wlm_json_configuration 파라미터를 수정합니다. 아래 표시된 JSON 콘텐츠가 포함된 파일의 파라미터를 수락합니다.

```
aws redshift modify-cluster-parameter-group \
    --parameter-group-name myclusterparametergroup \
    --parameters file://modify_pg.json
```

`modify_pg.json`의 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ModifyClusterParameterGroup을 참조하세요.

modify-cluster-snapshot-schedule

다음 코드 예시에서는 modify-cluster-snapshot-schedule의 사용 방법을 보여줍니다.

AWS CLI

클러스터 스냅샷 일정 수정

다음 modify-cluster-snapshot-schedule 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyClusterSnapshotSchedule을 참조하세요.

modify-cluster-snapshot

다음 코드 예시에서는 modify-cluster-snapshot의 사용 방법을 보여줍니다.

AWS CLI

클러스터 스냅샷 수정

다음 modify-cluster-snapshot 예시에서는 지정된 클러스터 스냅샷의 수동 보존 기간 설정의 값을 10일로 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-b1cel7t9",
        "Encrypted": false,
        "EncryptedWithHSM": false,
        "OwnerAccount": "123456789012",
        "TotalBackupSizeInMegaBytes": 64384.0,
        "ActualIncrementalBackupSizeInMegaBytes": 24.0,
        "BackupProgressInMegaBytes": 24.0,
        "CurrentBackupRateInMegaBytesPerSecond": 13.0011,
        "EstimatedSecondsToCompletion": 0,
        "ElapsedTimeInSeconds": 1,
        "Tags": [
            {
                "Key": "mytagkey",
                "Value": "mytagvalue"
            }
```

```
        ],
        "EnhancedVpcRouting": false,
        "MaintenanceTrackName": "current",
        "ManualSnapshotRetentionPeriod": 10,
        "ManualSnapshotRemainingDays": 6,
        "SnapshotRetentionStartTime": "2019-12-07T00:34:07.479Z"
    }
}
```

자세한 내용은 Amazon Redshift 클러스터 관리 안내서의 Amazon Redshift 스냅샷을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyClusterSnapshot을 참조하세요.

modify-cluster-subnet-group

다음 코드 예시에서는 modify-cluster-subnet-group의 사용 방법을 보여줍니다.

AWS CLI

클러스터 서브넷 그룹의 서브넷 수정 이 예시에서는 캐시 서브넷 그룹의 서브넷 목록을 수정하는 방법을 보여줍니다. 기본 출력 형식은 JSON입니다. 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"
   }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ModifyClusterSubnetGroup을 참조하세요.

modify-cluster

다음 코드 예시에서는 modify-cluster의 사용 방법을 보여줍니다.

AWS CLI

클러스터에 보안 그룹 연결 이 예시에서는 클러스터 보안 그룹을 지정된 클러스터에 연결하는 방법을 보여줍니다. 명령:

```
aws redshift modify-cluster --cluster-identifier mycluster --cluster-security-groups
 mysecuritygroup
```

클러스터의 유지 관리 기간 수정 여기에서는 클러스터의 주간 기본 유지 관리 기간을 일요일 오후 11시 15분에 시작하여 월요일 오전 3시 15분에 끝나는 최소 4시간으로 변경하는 방법을 보여줍니다. 명령:

```
aws redshift modify-cluster --cluster-identifier mycluster --preferred-maintenance-
window Sun:23:15-Mon:03:15
```

클러스터의 마스터 암호 변경 이 예시에서는 클러스터의 마스터 암호를 변경하는 방법을 보여 줍니다. 명령:

```
aws redshift modify-cluster --cluster-identifier mycluster --master-user-password
 A1b2c3d4
```

- API 세부 정보는 AWS CLI 명령 참조의 ModifyCluster를 참조하세요.

modify-event-subscription

다음 코드 예시에서는 modify-event-subscription의 사용 방법을 보여줍니다.

AWS CLI

이벤트 구독 수정

다음 modify-event-subscription 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyEventSubscription을 참조하세요.

modify-scheduled-action

다음 코드 예시에서는 modify-scheduled-action의 사용 방법을 보여줍니다.

AWS CLI

예약된 작업 수정

다음 modify-scheduled-action 예시에서는 지정된 기존 예약된 작업에 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"
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ModifyScheduledAction을 참조하세요.

modify-snapshot-copy-retention-period

다음 코드 예시에서는 modify-snapshot-copy-retention-period의 사용 방법을 보여줍니다.

AWS CLI

스냅샷 복사본 보존 기간 수정

다음 modify-snapshot-copy-retention-period 예시에서는 지정된 클러스터의 스냅샷을 소스 AWS 리전에서 복사한 후 대상 AWS 리전에서 유지할 일 수를 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.

- API 세부 정보는 AWS CLI 명령 참조의 ModifySnapshotCopyRetentionPeriod를 참조하세요.

modify-snapshot-schedule

다음 코드 예시에서는 `modify-snapshot-schedule`의 사용 방법을 보여줍니다.

AWS CLI

스냅샷 일정 수정

다음 modify-snapshot-schedule 예시에서는 지정된 스냅샷 일정 간격을 10시간으로 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.

- API 세부 정보는 AWS CLI 명령 참조의 ModifySnapshotSchedule을 참조하세요.

purchase-reserved-node-offering

다음 코드 예시에서는 purchase-reserved-node-offering의 사용 방법을 보여줍니다.

AWS CLI

예약 노드 구매 이 예시에서는 예약 노드 오퍼링을 구매하는 방법을 보여줍니다. reserved-node-offering-id는 describe-reserved-node-offerings를 직접적으로 호출하여 가져옵니다. 명령:

```
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"
     }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 PurchaseReservedNodeOffering을 참조하세요.

reboot-cluster

다음 코드 예시에서는 `reboot-cluster`의 사용 방법을 보여줍니다.

AWS CLI

클러스터 재부팅 이 예시에서는 클러스터를 재부팅합니다. 기본 출력 형식은 JSON입니다. 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 RebootCluster를 참조하세요.

`reset-cluster-parameter-group`

다음 코드 예시에서는 `reset-cluster-parameter-group`의 사용 방법을 보여줍니다.

AWS CLI

파라미터 그룹의 파라미터 재설정 이 예시에서는 파라미터 그룹의 모든 파라미터를 재설정하는 방법을 보여줍니다. 명령:

```
aws redshift reset-cluster-parameter-group --parameter-group-name
 myclusterparametergroup --reset-all-parameters
```

- API 세부 정보는 AWS CLI 명령 참조의 ResetClusterParameterGroup을 참조하세요.

resize-cluster

다음 코드 예시에서는 resize-cluster의 사용 방법을 보여줍니다.

AWS CLI

클러스터 크기 조정

다음 resize-cluster 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 ResizeCluster를 참조하세요.

`restore-from-cluster-snapshot`

다음 코드 예시에서는 `restore-from-cluster-snapshot`의 사용 방법을 보여줍니다.

AWS CLI

스냅샷에서 클러스터 복원 이 예시에서는 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 RestoreFromClusterSnapshot을 참조하세요.

`restore-table-from-cluster-snapshot`

다음 코드 예시에서는 `restore-table-from-cluster-snapshot`의 사용 방법을 보여줍니다.

AWS CLI

클러스터 스냅샷에서 테이블 복원

다음 `restore-table-from-cluster-snapshot` 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 RestoreTableFromClusterSnapshot을 참조하세요.

revoke-cluster-security-group-ingress

다음 코드 예시에서는 `revoke-cluster-security-group-ingress`의 사용 방법을 보여줍니다.

AWS CLI

EC2 보안 그룹에서 액세스 권한 취소 이 예시에서는 명명된 Amazon EC2 보안 그룹에 대한 액세스 권한을 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 범위에 대한 액세스 권한 취소 이 예시에서는 CIDR 범위에 대한 액세스 권한을 취소합니다. 명령:

```
aws redshift revoke-cluster-security-group-ingress --cluster-security-group-name
 mysecuritygroup --cidrip 192.168.100.100/32
```

- API 세부 정보는 AWS CLI 명령 참조의 RevokeClusterSecurityGroupIngress를 참조하세요.

revoke-snapshot-access

다음 코드 예시에서는 `revoke-snapshot-access`의 사용 방법을 보여줍니다.

AWS CLI

AWS 계정의 스냅샷 복원 권한 취소 이 예시에서는 AWS 계정 444455556666에서 스냅샷 my-snapshot-id를 복원할 수 있는 권한을 취소합니다. 기본 출력 형식은 JSON입니다. 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"
   }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 RevokeSnapshotAccess를 참조하세요.

rotate-encryption-key

다음 코드 예시에서는 rotate-encryption-key의 사용 방법을 보여줍니다.

AWS CLI

클러스터의 암호화 키 교체

다음 `rotate-encryption-key` 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 RotateEncryptionKey를 참조하세요.

AWS CLI를 사용한 Amazon Rekognition 예시

다음 코드 예시는 Amazon Rekognition과 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`compare-faces`

다음 코드 예시에서는 `compare-faces` 코드를 사용하는 방법을 보여줍니다.

자세한 내용은 이미지 내 얼굴 비교를 참조하세요.

AWS CLI

두 이미지에서 얼굴 비교

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 이미지 내 얼굴 비교를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CompareFaces를 참조하세요.

`create-collection`

다음 코드 예시에서는 `create-collection` 코드를 사용하는 방법을 보여줍니다.

자세한 내용은 컬렉션 생성을 참조하세요.

AWS CLI

컬렉션 생성

다음 `create-collection` 명령은 지정된 이름을 가진 컬렉션을 생성합니다.

```
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 컬렉션 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateCollection을 참조하세요.

`create-stream-processor`

다음 코드 예시에서는 `create-stream-processor` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

새 스트림 프로세서를 생성하는 방법

다음 `create-stream-processor` 예시에서는 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 Working with Streaming Videos를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateStreamProcessor를 참조하세요.

delete-collection

다음 코드 예시에서는 delete-collection 코드를 사용하는 방법을 보여줍니다.

자세한 내용은 컬렉션 삭제를 참조하세요.

AWS CLI

컬렉션 삭제

다음 delete-collection 명령은 지정된 컬렉션을 삭제합니다.

```
aws rekognition delete-collection \
    --collection-id MyCollection
```

출력:

```
{
    "StatusCode": 200
}
```

자세한 내용은 Amazon Rekognition 개발자 안내서의 컬렉션 삭제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteCollection을 참조하세요.

delete-faces

다음 코드 예시에서는 delete-faces 코드를 사용하는 방법을 보여줍니다.

자세한 내용은 컬렉션에서 얼굴 삭제를 참조하세요.

AWS CLI

컬렉션에서 얼굴 삭제

다음 delete-faces 명령은 컬렉션에서 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 컬렉션에서 얼굴 삭제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteFaces를 참조하세요.

delete-stream-processor

다음 코드 예시에서는 delete-stream-processor 코드를 사용하는 방법을 보여줍니다.

AWS CLI

스트림 프로세서를 삭제하는 방법

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 Working with Streaming Videos를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteStreamProcessor을 참조하세요.

describe-collection

다음 코드 예시에서는 describe-collection 코드를 사용하는 방법을 보여줍니다.

자세한 내용은 컬렉션 설명을 참조하세요.

AWS CLI

컬렉션 설명

다음 describe-collection 예시에서는 지정된 컬렉션의 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 컬렉션 설명을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeCollection을 참조하세요.

describe-stream-processor

다음 코드 예시에서는 describe-stream-processor 코드를 사용하는 방법을 보여줍니다.

AWS CLI

스트림 프로세서에 대한 정보를 가져오는 방법

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 Working with Streaming Videos를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeStreamProcessor](https://docs.aws.amazon.com/goto/aws-cli/rekognition-2016-06-27/DescribeStreamProcessor)를 참조하세요.

detect-faces

다음 코드 예시에서는 detect-faces 코드를 사용하는 방법을 보여줍니다.

자세한 내용은 [이미지에서 얼굴 감지]를 참조하세요.

AWS CLI

이미지에서 얼굴 감지

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.

- API 세부 정보는 AWS CLI 명령 참조의 DetectFaces를 참조하세요.

detect-labels

다음 코드 예시에서는 detect-labels 코드를 사용하는 방법을 보여줍니다.

자세한 내용은 이미지에서 레이블 감지를 참조하세요.

AWS CLI

이미지에서 레이블 감지

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.

- API 세부 정보는 AWS CLI 명령 참조의 DetectLabels를 참조하세요.

detect-moderation-labels

다음 코드 예시에서는 detect-moderation-labels 코드를 사용하는 방법을 보여줍니다.

자세한 내용은 부적절한 이미지 감지를 참조하세요.

AWS CLI

이미지에서 안전하지 않은 콘텐츠 감지

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.

- API 세부 정보는 AWS CLI 명령 참조의 DetectModerationLabels를 참조하세요.

detect-text

다음 코드 예시에서는 detect-text 코드를 사용하는 방법을 보여줍니다.

자세한 내용은 이미지에서 텍스트 감지를 참조하세요.

AWS CLI

이미지에서 텍스트 감지

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
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DetectText를 참조하세요.

disassociate-faces

다음 코드 예시에서는 `disassociate-faces` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

```
aws rekognition disassociate-faces --face-ids list-of-face-ids
  --user-id user-id --collection-id collection-name --region region-name
```

- API 세부 정보는 AWS CLI 명령 참조의 DisassociateFaces를 참조하세요.

get-celebrity-info

다음 코드 예시에서는 `get-celebrity-info` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

유명인의 정보 가져오기

다음 `get-celebrity-info` 명령은 지정된 유명인의 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 유명인의 정보 가져오기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetCelebrityInfo를 참조하세요.

`get-celebrity-recognition`

다음 코드 예시에서는 `get-celebrity-recognition` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

유명인 인정 작업의 결과를 얻으려면

다음 `get-celebrity-recognition` 명령은 이전에 `start-celebrity-recognition`을 호출하여 시작한 유명인 인식 작업의 결과를 표시합니다.

```
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 Recognizing Celebrities in a Stored Video을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetCelebrityRecognition을 참조하세요.

get-content-moderation

다음 코드 예시에서는 `get-content-moderation` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

안전하지 않은 콘텐츠 작업의 결과를 얻으려면

다음 `get-content-moderation` 명령은 이전에 `start-content-moderation`을 호출하여 시작한 안전하지 않은 콘텐츠 작업의 결과를 표시합니다.

```
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 Detecting Unsafe Stored Videos를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [GetContentModeration](https://docs.aws.amazon.com/goto/aws-cli/rekognition-2016-06-27/GetContentModeration)를 참조하세요.

get-face-detection

다음 코드 예시에서는 get-face-detection 코드를 사용하는 방법을 보여줍니다.

AWS CLI

얼굴 감지 작업의 결과를 얻으려면

다음 get-face-detection 명령은 이전에 start-face-detection을 호출하여 시작한 얼굴 감지 작업의 결과를 표시합니다.

```
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 Detecting Faces in a Stored Video를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetFaceDetection를 참조하세요.

get-face-search

다음 코드 예시에서는 get-face-search 코드를 사용하는 방법을 보여줍니다.

AWS CLI

얼굴 검색 작업의 결과를 가져오는 방법

다음 get-face-search 명령은 이전에 start-face-search을 호출하여 시작한 얼굴 검색 작업의 결과를 표시합니다.

```
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 Searching Stored Videos for Faces를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetFaceSearch을 참조하세요.

get-label-detection

다음 코드 예시에서는 `get-label-detection` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

객체 및 장면 감지 작업의 결과를 가져오는 방법

다음 `get-label-detection` 명령은 이전에 `start-label-detection`을 호출하여 시작한 객체 및 장면 감지 작업의 결과를 표시합니다.

```
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 Detecting Labels in a Video를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetLabelDetection을 참조하세요.

get-person-tracking

다음 코드 예시에서는 get-person-tracking 코드를 사용하는 방법을 보여줍니다.

AWS CLI

인력 경로 지정 작업의 결과를 가져오는 방법

다음 get-person-tracking 명령은 이전에 start-person-tracking을 호출하여 시작한 사람 경로 지정 작업의 결과를 표시합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 GetPersonTracking을 참조하세요.

`index-faces`

다음 코드 예시에서는 `index-faces` 코드를 사용하는 방법을 보여줍니다.

자세한 내용은 컬렉션에 얼굴 추가를 참조하세요.

AWS CLI

컬렉션에 얼굴 추가

다음 index-faces 명령은 이미지에서 찾은 얼굴을 지정된 컬렉션에 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 컬렉션에 얼굴 추가를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 IndexFaces를 참조하세요.

`list-collections`

다음 코드 예시에서는 `list-collections` 코드를 사용하는 방법을 보여줍니다.

자세한 내용은 컬렉션 나열을 참조하세요.

AWS CLI

사용 가능한 컬렉션 나열

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 컬렉션 나열을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListCollections를 참조하세요.

list-faces

다음 코드 예시에서는 list-faces 코드를 사용하는 방법을 보여줍니다.

자세한 내용은 컬렉션 내 얼굴 나열을 참조하세요.

AWS CLI

컬렉션 내 얼굴 나열

다음 list-faces 명령은 지정된 컬렉션에 있는 얼굴을 나열합니다.

```
aws rekognition list-faces \
    --collection-id MyCollection
```

출력:

```
{
    "FaceModelVersion": "3.0",
    "Faces": [
        {
            "BoundingBox": {
                "Width": 0.521631002426147 5,
                "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 컬렉션의 얼굴 나열을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListFaces를 참조하세요.

list-stream-processors

다음 코드 예시에서는 `list-stream-processors` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

계정의 스트림 프로세서 나열

다음 `list-stream-processors` 명령은 계정의 스트림 프로세서와 각 스트림 프로세서의 상태를 나열합니다.

```
aws rekognition list-stream-processors
```

출력:

```
{
    "StreamProcessors": [
        {
            "Status": "STOPPED",
            "Name": "my-stream-processor"
        }
    ]
}
```

자세한 내용은 Amazon Rekognition 개발자 안내서의 Working with Streaming Videos를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListStreamProcessors를 참조하세요.

`recognize-celebrities`

다음 코드 예시에서는 `recognize-celebrities` 코드를 사용하는 방법을 보여줍니다.

자세한 내용은 이미지에서 유명인 인식을 참조하세요.

AWS CLI

이미지에서 유명인 인식

다음 `recognize-celebrities` 명령은 Amazon S3 버킷에 저장된 지정된 이미지에서 유명인을 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 이미지에서 유명인 인식을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RecognizeCelebrities를 참조하세요.

search-faces-by-image

다음 코드 예시에서는 search-faces-by-image 코드를 사용하는 방법을 보여줍니다.

자세한 내용은 [얼굴 검색(이미지)](#)을 참조하세요.

AWS CLI

이미지에서 가장 큰 얼굴과 일치하는 얼굴을 컬렉션에서 검색

다음 search-faces-by-image 명령은 지정된 이미지에서 가장 큰 얼굴과 일치하는 얼굴을 컬렉션에서 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": 0.16180199384689331,
                    "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.

- API 세부 정보는 AWS CLI 명령 참조의 SearchFacesByImage를 참조하세요.

search-faces

다음 코드 예시에서는 `search-faces` 코드를 사용하는 방법을 보여줍니다.

자세한 내용은 얼굴 검색(얼굴 ID)을 참조하세요.

AWS CLI

얼굴 ID와 일치하는 얼굴을 컬렉션에서 검색

다음 `search-faces` 명령은 컬렉션에서 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.

- API 세부 정보는 AWS CLI 명령 참조의 SearchFaces를 참조하세요.

`start-celebrity-recognition`

다음 코드 예시에서는 `start-celebrity-recognition` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

저장된 동영상에서 유명인 인식 시작

다음 `start-celebrity-recognition` 명령은 Amazon S3 버킷에 저장된 지정된 동영상 파일에서 유명인을 찾는 작업을 시작합니다.

```
aws rekognition start-celebrity-recognition \
    --video "S3Object={Bucket=MyVideoS3Bucket,Name=MyVideoFile.mpg}"
```

출력:

```
{
    "JobId": "1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef"
}
```

자세한 내용은 Amazon Rekognition 개발자 안내서의 Recognizing Celebrities in a Stored Video을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartCelebrityRecognition을 참조하세요.

`start-content-moderation`

다음 코드 예시에서는 `start-content-moderation` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

저장된 비디오에서 안전하지 않은 콘텐츠 인식을 시작하는 방법

다음 start-content-moderation 명령은 Amazon S3 버킷에 저장된 지정된 동영상 파일에서 안전하지 않은 콘텐츠를 탐지하는 작업을 시작합니다.

```
aws rekognition start-content-moderation \
    --video "S3Object={Bucket=MyVideoS3Bucket,Name=MyVideoFile.mpg}"
```

출력:

```
{
    "JobId": "1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef"
}
```

자세한 내용은 Amazon Rekognition 개발자 안내서의 Detecting Unsafe Stored Videos를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartContentModeration을 참조하세요.

start-face-detection

다음 코드 예시에서는 start-face-detection 코드를 사용하는 방법을 보여줍니다.

AWS CLI

저장된 동영상에서 얼굴 감지

다음 start-face-detection 명령은 Amazon S3 버킷에 저장된 지정된 동영상 파일에서 얼굴을 탐지하는 작업을 시작합니다.

```
aws rekognition start-face-detection
    --video "S3Object={Bucket=MyVideoS3Bucket,Name=MyVideoFile.mpg}"
```

출력:

```
{
    "JobId": "1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef"
}
```

자세한 내용은 Amazon Rekognition 개발자 안내서의 [Detecting Faces in a Stored Video](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [StartFaceDetection](#)를 참조하세요.

`start-face-search`

다음 코드 예시에서는 `start-face-search` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

컬렉션에서 동영상에서 감지된 얼굴과 일치하는 얼굴 검색

다음 `start-face-search` 명령은 Amazon S3 버킷에서 지정된 비디오 파일에서 탐지된 얼굴과 일치하는 컬렉션의 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 [Searching Stored Videos for Faces](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조에서 [StartFaceSearch](#)를 참조하세요.

`start-label-detection`

다음 코드 예시에서는 `start-label-detection` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

비디오에서 객체 및 장면을 감지하는 방법

다음 `start-label-detection` 명령은 Amazon S3 버킷에 저장된 지정된 동영상 파일에서 객체 및 소스를 탐지하는 작업을 시작합니다.

```
aws rekognition start-label-detection \
    --video "S3Object={Bucket=MyVideoS3Bucket,Name=MyVideoFile.mpg}"
```

출력:

```
{
    "JobId": "1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef"
}
```

자세한 내용은 Amazon Rekognition 개발자 안내서의 Detecting Labels in a Video를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartLabelDetection을 참조하세요.

start-person-tracking

다음 코드 예시에서는 `start-person-tracking` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

저장된 비디오에서 사람들의 경로를 시작하는 방법

다음 `start-person-tracking` 명령은 Amazon S3 버킷에 저장된 지정된 동영상 필드에서 사람들이 이동하는 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.

- API 세부 정보는 AWS CLI 명령 참조의 StartPersonTracking을 참조하세요.

start-stream-processor

다음 코드 예시에서는 `start-stream-processor` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

스트림 프로세서 시작

다음 `start-stream-processor` 명령은 지정된 동영상 스트림 프로세서를 시작합니다.

```
aws rekognition start-stream-processor \
    --name my-stream-processor
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Rekognition 개발자 안내서의 Working with Streaming Videos를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartStreamProcessor을 참조하세요.

`stop-stream-processor`

다음 코드 예시에서는 `stop-stream-processor` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

실행 중인 스트림 프로세서를 중지하는 방법

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 Working with Streaming Videos를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StopStreamProcessor를 참조하세요.

AWS CLI를 사용한 AWS RAM 예시

다음 코드 예시에서는 AWS RAM에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

accept-resource-share-invitation

다음 코드 예시에서는 accept-resource-share-invitation의 사용 방법을 보여줍니다.

AWS CLI

리소스 공유 초대 수락

다음 accept-resource-share-invitation 예시에서는 지정된 리소스 공유 초대를 수락합니다. 초대된 계정의 위탁자는 공유에서 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [AcceptResourceShareInvitation](#)을 참조하세요.

associate-resource-share-permission

다음 코드 예시에서는 associate-resource-share-permission의 사용 방법을 보여줍니다.

AWS CLI

리소스 공유에 RAM 관리형 권한 연결

다음 associate-resource-share-permission 예시에서는 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [AssociateResourceSharePermission](#)을 참조하세요.

associate-resource-share

다음 코드 예시에서는 associate-resource-share의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 리소스 공유에 리소스 연결

다음 associate-resource-share 예시에서는 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
        }
    ]
}
```

예시 2: 리소스 공유에 위탁자 연결

다음 `associate-resource-share` 예시에서는 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 AssociateResourceShare를 참조하세요.

`create-resource-share`

다음 코드 예시에서는 `create-resource-share`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 리소스 공유 생성

다음 `create-resource-share` 예시에서는 지정된 이름으로 빈 리소스 공유를 생성합니다. 공유에 리소스, 위탁자 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
    }
}
```

예시 2: AWS 계정을 위탁자로 사용하여 리소스 공유 생성

다음 `create-resource-share` 예시에서는 리소스 공유를 생성하고 지정된 AWS 계정(222222222222)에 대한 액세스 권한을 부여합니다. 지정된 위탁자가 동일한 AWS 조직에 속하지 않은 경우 초대가 전송되며 액세스 권한이 부여되기 전에 수락되어야 합니다.

```
aws ram create-resource-share \
    --name MyNewResourceShare \
    --principals 222222222222
```

예시 3: AWS 조직으로 제한된 리소스 공유 생성

다음 create-resource-share 예시에서는 계정이 멤버로 속한 AWS 조직의 계정으로 제한된 리소스 공유를 생성하고 지정된 OU를 위탁자로 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
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateResourceShare를 참조하세요.

delete-resource-share

다음 코드 예시에서는 delete-resource-share의 사용 방법을 보여줍니다.

AWS CLI

리소스 공유 삭제

다음 delete-resource-share 예시에서는 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteResourceShare를 참조하세요.

disassociate-resource-share-permission

다음 코드 예시에서는 disassociate-resource-share-permission의 사용 방법을 보여줍니다.

AWS CLI

리소스 공유에서 리소스 유형에 대한 RAM 관리형 권한 제거

다음 disassociate-resource-share-permission 예시에서는 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DisassociateResourceSharePermission을 참조하세요.

`disassociate-resource-share`

다음 코드 예시에서는 `disassociate-resource-share`의 사용 방법을 보여줍니다.

AWS CLI

리소스 공유에서 리소스 제거

다음 `disassociate-resource-share` 예시에서는 지정된 리소스 공유에서 지정된 리소스, 이 경우 VPC 서브넷을 제거합니다. 리소스 공유에 액세스할 수 있는 위탁자는 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
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [DisassociateResourceShare](https://docs.aws.amazon.com/goto/aws-cli/ram-2018-01-04/DisassociateResourceShare)를 참조하세요.

`enable-sharing-with-aws-organization`

다음 코드 예시에서는 `enable-sharing-with-aws-organization`의 사용 방법을 보여줍니다.

AWS CLI

AWS 조직에서 리소스 공유 활성화

다음 enable-sharing-with-aws-organization 예시에서는 조직 및 조직 단위에서 리소스 공유를 활성화합니다.

```
aws ram enable-sharing-with-aws-organization
```

다음 출력은 성공을 나타냅니다.

```
{
    "returnValue": true
}
```

- API 세부 정보는 AWS CLI 명령 참조의 EnableSharingWithAwsOrganization을 참조하세요.

get-permission

다음 코드 예시에서는 get-permission의 사용 방법을 보여줍니다.

AWS CLI

RAM 관리형 권한의 세부 정보 가져오기

다음 get-permission 예시에서는 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
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetPermission을 참조하세요.

get-resource-policies

다음 코드 예시에서는 get-resource-policies의 사용 방법을 보여줍니다.

AWS CLI

리소스 정책 가져오기

다음 get-resource-policies 예시에서는 리소스 공유에 연결된 지정된 리소스에 대한 리소스 기반 권한 정책을 표시합니다.

```
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\"}]}"
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetResourcePolicies를 참조하세요.

get-resource-share-associations

다음 코드 예시에서는 get-resource-share-associations의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 모든 리소스 유형에 대한 모든 리소스 연결 나열

다음 `get-resource-share-associations` 예시에서는 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
        }
    ]
}
```

예시 2: 리소스 공유에 대한 위탁자 연결 나열

다음 get-resource-share-associations 예시에서는 지정된 리소스 공유에 대한 위탁자 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
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetResourceShareAssociations를 참조하세요.

get-resource-share-invitations

다음 코드 예시에서는 get-resource-share-invitations의 사용 방법을 보여줍니다.

AWS CLI

리소스 공유 초대 나열

다음 get-resource-share-invitations 예시에서는 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetResourceShareInvitations를 참조하세요.

get-resource-shares

다음 코드 예시에서는 get-resource-shares의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 자신이 소유한 리소스 공유를 나열하고 다른 사용자와 공유

다음 get-resource-shares 예시에서는 자신이 생성하고 다른 사용자와 공유하는 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
        }
    ]
}
```

예시 2: 다른 사용자가 소유하고 자신에게 공유된 리소스 공유 나열

다음 `get-resource-shares` 예시에서는 다른 사용자가 생성하고 자신에게 공유된 리소스 공유를 나열합니다. 이 예시에서는 아무것도 없습니다.

```
aws ram get-resource-shares \
    --resource-owner OTHER-ACCOUNTS
```

출력:

```
{
    "resourceShares": []
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetResourceShares를 참조하세요.

`list-pending-invitation-resources`

다음 코드 예시에서는 `list-pending-invitation-resources`의 사용 방법을 보여줍니다.

AWS CLI

보류 중인 리소스 공유에서 사용할 수 있는 리소스 나열

다음 `list-pending-invitation-resources` 예시에서는 지정된 초대에 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"
      }
   ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [ListPendingInvitationResources](https://docs.aws.amazon.com/goto/aws-cli/ram-2018-01-04/ListPendingInvitationResources)를 참조하세요.

list-permissions

다음 코드 예시에서는 `list-permissions`의 사용 방법을 보여줍니다.

AWS CLI

사용 가능한 RAM 관리형 권한 나열

다음 `list-permissions` 예시에서는 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` 예시에서는 모든 리소스 유형에 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
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListPermissions를 참조하세요.

list-principals

다음 코드 예시에서는 list-principals의 사용 방법을 보여줍니다.

AWS CLI

리소스에 액세스할 수 있는 위탁자 나열

다음 list-principals 예시에서는 모든 리소스 공유를 통해 지정된 유형의 리소스에 액세스할 수 있는 위탁자 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
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListPrincipals를 참조하세요.

`list-resource-share-permissions`

다음 코드 예시에서는 `list-resource-share-permissions`의 사용 방법을 보여줍니다.

AWS CLI

리소스 공유에 현재 연결된 모든 RAM 관리형 권한 나열

다음 `list-resource-share-permissions` 예시에서는 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
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListResourceSharePermissions를 참조하세요.

list-resource-types

다음 코드 예시에서는 `list-resource-types`의 사용 방법을 보여줍니다.

AWS CLI

AWS RAM에서 지원하는 리소스 유형 나열

다음 `list-resource-types` 예시에서는 현재 AWS RAM에서 지원하는 모든 리소스 유형을 나열합니다.

```
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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListResourceTypes를 참조하세요.

`list-resources`

다음 코드 예시에서는 `list-resources`의 사용 방법을 보여줍니다.

AWS CLI

리소스 공유에 연결된 리소스 나열

다음 `list-resources` 예시에서는 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
```

```
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListResources 섹션을 참조하세요.

promote-resource-share-created-from-policy

다음 코드 예시에서는 promote-resource-share-created-from-policy의 사용 방법을 보여줍니다.

AWS CLI

리소스 정책 기반 리소스 공유를 AWS RAM의 전체 기능으로 승격

다음 promote-resource-share-created-from-policy 예시에서는 리소스 기반 정책을 연결하여 묵시적으로 생성한 리소스 공유를 가져와 AWS RAM 콘솔과 CLI 및 API 작업에서 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 PromoteResourceShareCreatedFromPolicy를 참조하세요.

reject-resource-share-invitation

다음 코드 예시에서는 reject-resource-share-invitation의 사용 방법을 보여줍니다.

AWS CLI

리소스 공유 초대 거부

다음 reject-resource-share-invitation 예시에서는 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"
    }
]
```

- API 세부 정보는 AWS CLI 명령 참조의 RejectResourceShareInvitation을 참조하세요.

tag-resource

다음 코드 예시에서는 tag-resource의 사용 방법을 보여줍니다.

AWS CLI

리소스 공유에 태그 추가

다음 tag-resource 예시에서는 태그 키 project와 관련 값 lima를 지정된 리소스 공유에 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.

- API 세부 정보는 AWS CLI 명령 참조의 [TagResource](https://docs.aws.amazon.com/goto/aws-cli/ram-2018-01-04/TagResource)를 참조하세요.

untag-resource

다음 코드 예시에서는 untag-resource의 사용 방법을 보여줍니다.

AWS CLI

리소스 공유에서 태그 제거

다음 untag-resource 예시에서는 지정된 리소스 공유에서 project 태그 키 및 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.

- API 세부 정보는 AWS CLI 명령 참조의 [UntagResource](https://docs.aws.amazon.com/goto/aws-cli/ram-2018-01-04/UntagResource)를 참조하세요.

update-resource-share

다음 코드 예시에서는 update-resource-share의 사용 방법을 보여줍니다.

AWS CLI

리소스 공유 업데이트

다음 update-resource-share 예시에서는 지정된 리소스 공유를 변경하여 AWS 조직에 없는 외부 위탁자를 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
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateResourceShare를 참조하세요.

AWS CLI를 사용한 Resource Explorer 예시

다음 코드 예시에서는 에서 Resource Explorer에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`associate-default-view`

다음 코드 예시에서는 `associate-default-view` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Resource Explorer 뷰를 해당 AWS 리전의 기본값으로 설정하는 방법

다음 `associate-default-view` 예시에서는 ARN으로 지정된 보기를 작업을 호출하는 AWS 리전의 기본 보기로 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 Setting a default view in an AWS Region을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AssociateDefaultView 섹션을 참조하세요.

batch-get-view

다음 코드 예시에서는 `batch-get-view` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

여러 Resource Explorer 뷰에 대한 세부 정보를 검색하는 방법

다음 `batch-get-view` 예시에서는 ARN으로 지정된 두 뷰에 대한 세부 정보를 표시합니다. view-arn 파라미터에서 공백을 사용하여 여러 ARN을 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": []
}
```

보기에 대한 자세한 내용은 AWS Resource Explorer 사용 설명서에서 Resource Explorer 보기에 대한 정보를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 BatchGetView 섹션을 참조하세요.

`create-index`

다음 코드 예시에서는 `create-index` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

색인을 생성하여 AWS 리전에서 Resource Explorer 켜기

다음 create-index 예시에서는 연산이 호출되는 AWS 리전에 로컬 인덱스를 생성합니다. 값을 지정하지 않으면 AWS CLI가 자동으로 임의의 client-token 파라미터 값을 생성하여 AWS 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 만든 후에는 update-index-type 명령을 실행하여 계정에 대한 집계 인덱스로 변환할 수 있습니다.

자세한 내용은 AWS Resource Explorer 사용 설명서의 Turning on Resource Explorer in an AWS Region to index your resources를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateIndex 섹션을 참조하세요.

create-view

다음 코드 예시에서는 create-view 코드를 사용하는 방법을 보여줍니다.

AWS CLI

예시 1: AWS 리전의 인덱스에 대해 필터링되지 않은 뷰를 생성하는 방법

다음 create-view 예시에서는 지정된 AWS 리전에 필터링 없이 리전의 모든 결과를 반환하는 뷰를 만듭니다. 보기에는 반환된 결과에 대한 태그 필드(선택 사항)가 포함되어 있습니다. 이 보기는 집계 인덱스가 포함된 리전에서 만들어지므로 계정의 모든 리전(Resource Explorer 인덱스가 포함된 리전)의 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"
    }
}
```

예시 2: Amazon EC2와 연결된 리소스만 반환하는 보기를 생성하는 방법

다음 `create-view` 명령은 AWS 리전 `us-east-1`에서 리전 내 Amazon EC2 서비스와 연결된 리소스만 반환하는 보기를 만듭니다. 보기에는 반환된 결과에 대한 `Tags` 필드(선택 사항)가 포함되어 있습니다. 이 보기는 집계 인덱스가 포함된 리전에서 만들어지므로 계정의 모든 리전(Resource Explorer 인덱스가 포함된 리전)의 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 Creating views for search를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateView 섹션을 참조하세요.

delete-index

다음 코드 예시에서는 delete-index 코드를 사용하는 방법을 보여줍니다.

AWS CLI

인덱스를 삭제하여 AWS 리전에서 Resource Explorer를 끄려면

다음 delete-index 예시에서는 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 AWS Resource Explorer 끄기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteIndex 섹션을 참조하세요.

delete-view

다음 코드 예시에서는 delete-view 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Resource Explorer 보기를 삭제하는 방법

다음 delete-view 예시에서는 ARN에서 지정한 보기를 삭제합니다.

```
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 Deleting views를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteView 섹션을 참조하세요.

disassociate-default-view

다음 코드 예시에서는 disassociate-default-view 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS 리전에 대한 기본 Resource Explorer 보기를 제거하는 방법

다음 disassociate-default-view 명령은 작업을 호출하는 AWS 리전에 대한 기본 Resource Explorer 보기를 제거합니다. 이 작업을 수행한 후 리전 내 모든 검색 작업은 뷰를 명시적으로 지정해야 하며, 그렇지 않으면 작업이 실패합니다.

```
aws resource-explorer-2 disassociate-default-view
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Resource Explorer 사용 설명서의 Setting a default view in an AWS Region을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisassociateDefaultView 섹션을 참조하세요.

`get-default-view`

다음 코드 예시에서는 `get-default-view` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS 해당 리전의 기본 뷰인 Resource Explorer 뷰를 검색하는 방법

다음 `get-default-view` 예시에서는 작업을 호출하는 AWS 리전의 기본값인 보기의 ARN을 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 Setting a default view in an AWS Region을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetDefaultView 섹션을 참조하세요.

`get-index`

다음 코드 예시에서는 `get-index` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

예시 1: Resource Explorer 애그리게이터 인덱스의 세부 정보를 검색하는 방법

다음 get-index 예시에서는 지정된 AWS 리전 내 Resource Explorer 인덱스에 대한 세부 정보를 표시합니다. 지정된 리전에는 계정에 대한 집계 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"
}
```

예시 2: Resource Explorer 로컬 인덱스에 대한 세부 정보를 검색하는 방법

다음 get-index 예시에서는 지정된 AWS 리전 내 Resource Explorer 인덱스에 대한 세부 정보를 표시합니다. 지정한 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 AWS Resource Explorer 사용 설명서의 Resource Explorer가 켜져 있는 AWS 리전 확인을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetIndex 섹션을 참조하세요.

get-view

다음 코드 예시에서는 `get-view` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Resource Explorer 보기에 대한 세부 정보를 검색하는 방법

다음 `get-view` 예시에서는 ARN으로 지정된 뷰에 대한 세부 정보를 표시합니다.

```
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"
```

```
    }
}
```

보기에 대한 자세한 내용은 AWS Resource Explorer 사용 설명서에서 Resource Explorer 보기에 대한 정보를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetView 섹션을 참조하세요.

list-indexes

다음 코드 예시에서는 `list-indexes` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Resource Explorer에 인덱스가 있는 AWS 리전을 나열하는 방법

다음 `list-indexes` 예시에서는 Resource Explorer에 인덱스가 있는 모든 리전의 인덱스를 나열합니다. 응답은 각 인덱스의 유형, AWS 리전 및 해당 리전의 ARN을 지정합니다.

```
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 AWS Resource Explorer 사용 설명서의 Resource Explorer가 켜져 있는 AWS 리전 확인을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListIndexes 섹션을 참조하세요.

list-supported-resource-types

다음 코드 예시에서는 `list-supported-resource-types` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Resource Explorer에 인덱스가 있는 AWS 리전을 나열하는 방법

다음 `list-supported-resource-types` 예시에서는 &AREXlong; RAM에서 지원하는 모든 리소스 유형을 나열합니다. 예시 응답에는 추가 호출을 통해 검색할 수 있는 출력이 더 있음을 나타내는 `NextToken` 값이 포함되어 있습니다.

```
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 --starting-token의 값으로 전달합니다. NextToken이 응답이 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 AWS Resource Explorer 사용 설명서의 Resource Explorer가 켜져 있는 AWS 리전 확인을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListSupportedResourceTypes 섹션을 참조하세요.

list-tags-for-resource

다음 코드 예시에서는 `list-tags-for-resource` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Resource Explorer 뷰 또는 인덱스에 연결된 태그를 나열하는 방법

다음 `list-tags-for-resource` 예시에서는 지정된 ARN으로 보기에 연결된 태그 키 및 값 페어를 나열합니다. 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"
    }
}
```

보기에 태그를 지정하는 방법에 대한 자세한 내용은 AWS Resource Explorer 사용 설명서의 액세스 제어를 위해 보기에 태그 지정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource를 참조하세요.

list-views

다음 코드 예시에서는 `list-views` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS 리전에서 사용할 수 있는 Resource Explorer 뷰를 나열하는 방법

다음 `list-views` 예시에서는 작업을 호출하는 리전에서 사용 가능한 모든 보기가 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"
    ]
}
```

보기에 대한 자세한 내용은 AWS Resource Explorer 사용 설명서에서 Resource Explorer 보기에 대한 정보를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListViews 섹션을 참조하세요.

search

다음 코드 예시에서는 `search` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

예시 1: 기본 보기를 사용하여 검색하는 방법

다음 `search` 예시에서는 지정된 리소스 중 서비스와 관련된 모든 리소스를 표시합니다. 검색은 리전에 대한 기본 보기를 사용합니다. 예시 응답에는 추가 호출을 통해 검색할 수 있는 출력이 더 있음을 나타내는 `NextToken` 값이 포함되어 있습니다.

```
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"
}
```


예시 2: 지정된 보기를 사용하여 검색하는 방법

다음 search 예시 검색은 지정된 AWS 리전에서 지정된 보기를 통해 볼 수 있는 모든 리소스("*")를 표시합니다. 보기에 연결된 필터로 인해 결과에는 Amazon EC2와 관련된 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 Resource Explorer 사용 설명서AWS의 Using AWS Resource Explorer to search for resources를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 Search 섹션을 참조하세요.

tag-resource

다음 코드 예시에서는 `tag-resource` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Resource Explorer 뷰에 태그를 지정하는 방법

다음 `tag-resource` 예시에서는 지정된 ARN이 있는 보기에 'production' 값과 함께 태그 키 'environment'를 추가합니다.

```
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 Tagging views for access control을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

`untag-resource`

다음 코드 예시에서는 `untag-resource` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Resource Explorer 보기에서 태그를 제거하는 방법

다음 `untag-resource` 예시에서는 지정된 ARN이 있는 보기에서 키 이름이 'environment'인 태그를 모두 제거합니다.

```
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 Tagging views for access control을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

`update-index-type`

다음 코드 예시에서는 `update-index-type` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Resource Explorer 인덱스의 유형을 변경하는 방법

다음 update-index-type 예시에서는 지정된 인덱스를 local 유형에서 aggregator 유형으로 변환하여 계정의 모든 AWS 리전에서 리소스를 검색할 수 있는 기능을 켭니다. 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 애그리게이터 인덱스를 만들어 리전 간 검색 켜기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateIndexType 섹션을 참조하세요.

update-view

다음 코드 예시에서는 update-view 코드를 사용하는 방법을 보여줍니다.

AWS CLI

예시 1: Resource Explorer 보기에 포함된 IncludedProperties 필드를 업데이트하는 방법

다음 update-view 예시에서는 `IncludedProperties` 옵션에 `tags`를 추가하여 지정된 보기를 업데이트합니다. 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"
    }
}
```

예시 2: 뷰에 연결된 필터를 업데이트하는 방법

다음 `update-view` 예시에서는 Amazon EC2 서비스와 연결된 리소스 유형으로만 결과를 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"
    }
}
```

보기에 대한 자세한 내용은 AWS Resource Explorer 사용 설명서에서 Resource Explorer 보기에 대한 정보를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateView 섹션을 참조하세요.

AWS CLI를 사용한 Resource Groups 예시

다음 코드 예시에서는 Resource Groups와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`create-group`

다음 코드 예시에서는 `create-group`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 태그 기반 리소스 그룹 생성

다음 `create-group` 예시에서는 현재 리전에 Amazon EC2 인스턴스의 태그 기반 리소스 그룹을 생성합니다. 키 `Name` 및 값 `WebServers`로 태그가 지정된 리소스에 대한 쿼리를 기반으로 합니다. 그룹 이름은 `tbq-WebServer`입니다. 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\"]}]}"
    }
}
```

예시 2: CloudFormation 스택 기반 리소스 그룹 생성

다음 `create-group` 예시에서는 `sampleCFNstackgroup`이라는 AWS CloudFormation 스택 기반 리소스 그룹을 생성합니다. 쿼리에는 AWS Resource Groups에서 지원하는 지정된 CloudFormation 스택의 모든 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 AWS Resource Groups 사용자 안내서의 그룹 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateGroup을 참조하세요.

delete-group

다음 코드 예시에서는 delete-group의 사용 방법을 보여줍니다.

AWS CLI

리소스 그룹의 설명 업데이트

다음 delete-group 예시에서는 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 AWS Resource Groups 사용자 안내서의 그룹 삭제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteGroup을 참조하세요.

get-group-query

다음 코드 예시에서는 get-group-query의 사용 방법을 보여줍니다.

AWS CLI

리소스 그룹에 연결된 쿼리 가져오기

다음 get-group-query 예시에서는 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\"]}]}"
        }
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetGroupQuery를 참조하세요.

get-group

다음 코드 예시에서는 get-group의 사용 방법을 보여줍니다.

AWS CLI

리소스 그룹의 정보 가져오기

다음 `get-group` 예시에서는 지정된 리소스 그룹의 세부 정보를 표시합니다. 그룹에 연결된 쿼리를 가져오려면 `get-group-query`를 사용합니다.

```
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."
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [GetGroup](#)을 참조하세요.

`get-tags`

다음 코드 예시에서는 `get-tags`의 사용 방법을 보여줍니다.

AWS CLI

리소스 그룹에 연결된 태그 가져오기

다음 `get-tags` 예시에서는 지정된 리소스 그룹(멤버가 아닌 그룹 자체)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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetTags를 참조하세요.

list-group-resources

다음 코드 예시에서는 `list-group-resources`의 사용 방법을 보여줍니다.

AWS CLI

리소스 그룹의 모든 리소스 나열

예시 1: 다음 `list-resource-groups` 예시에서는 지정된 리소스 그룹에 포함된 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"
        }
    ]
}
```

예시 2: 다음 예시에서는 ':AWS:EC2::Instance'의 'resource-type'도 있는 그룹의 모든 리소스를 나열합니다.

aws resource-groups list-group-resources --group-name tbq-WebServer --filters Name=resource-type,Values=AWS::EC2::Instance

- API 세부 정보는 AWS CLI 명령 참조의 ListGroupResources를 참조하세요.

`list-groups`

다음 코드 예시에서는 `list-groups`의 사용 방법을 보여줍니다.

AWS CLI

사용 가능한 리소스 그룹 나열

다음 `list-groups` 예시에서는 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListGroups를 참조하세요.

list-resource-groups

다음 코드 예시에서는 `list-resource-groups`의 사용 방법을 보여줍니다.

AWS CLI

리소스 그룹의 모든 리소스 나열

다음 `list-resource-groups` 예시에서는 지정된 리소스 그룹에 포함된 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListResourceGroups를 참조하세요.

put-group-configuration

다음 코드 예시에서는 `put-group-configuration`의 사용 방법을 보여줍니다.

AWS CLI

리소스 그룹에 서비스 구성 연결

예시 1: 다음 `put-group-configuration` 예시에서는 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 Resource Groups API 참조 안내서의 리소스 그룹의 서비스 구성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutGroupConfiguration을 참조하세요.

search-resources

다음 코드 예시에서는 search-resources의 사용 방법을 보여줍니다.

AWS CLI

쿼리와 일치하는 리소스 찾기

다음 search-resources 예시에서는 지정된 쿼리와 일치하는 모든 AWS 리소스 목록을 가져옵니다.

```
aws resource-groups search-resources \
    --resource-query file://query.json
```

query.json의 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 SearchResources를 참조하세요.

tag

다음 코드 예시에서는 tag의 사용 방법을 보여줍니다.

AWS CLI

리소스 그룹에 태그 연결

다음 tag 예시에서는 지정된 태그 키와 값 페어를 지정된 리소스 그룹(멤버가 아닌 그룹 자체)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 AWS Resource Groups 사용자 안내서의 태그 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 Tag를 참조하세요.

untag

다음 코드 예시에서는 untag의 사용 방법을 보여줍니다.

AWS CLI

리소스에서 태그 제거

다음 untags 예시에서는 멤버가 아닌 리소스 그룹 자체에서 지정된 키가 있는 태그를 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 AWS Resource Groups 사용자 안내서의 태그 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 Untag를 참조하세요.

update-group-query

다음 코드 예시에서는 update-group-query의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 태그 기반 리소스 그룹에 대한 쿼리 업데이트

다음 update-group-query 예시에서는 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 AWS Resource Groups 사용자 안내서의 그룹 업데이트를 참조하세요.

예시 2: CloudFormation 스택 기반 리소스 그룹에 대한 쿼리 업데이트

다음 update-group-query 예시에서는 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 AWS Resource Groups 사용자 안내서의 그룹 업데이트를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateGroupQuery를 참조하세요.

update-group

다음 코드 예시에서는 update-group의 사용 방법을 보여줍니다.

AWS CLI

리소스 그룹의 설명 업데이트

다음 update-group 예시에서는 지정된 리소스 그룹의 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 AWS Resource Groups 사용자 안내서의 그룹 업데이트를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateGroup을 참조하세요.

AWS CLI를 사용한 Resource Groups Tagging API 예시

다음 코드 예시에서는 Resource Groups Tagging API와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

get-resources

다음 코드 예시에서는 `get-resources`의 사용 방법을 보여줍니다.

AWS CLI

태그 지정된 리소스의 목록 가져오기

다음 `get-resources` 예시에서는 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 Resource Groups Tagging API 참조의 GetResources를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetResources를 참조하세요.

get-tag-keys

다음 코드 예시에서는 get-tag-keys의 사용 방법을 보여줍니다.

AWS CLI

모든 태그 키 목록 가져오기

다음 get-tag-keys 예시에서는 계정의 리소스에서 사용하는 모든 태그 키 이름의 목록을 가져옵니다.

```
aws resourcegroupstaggingapi get-tag-keys
```

출력:

```
{
    "TagKeys": [
        "Environment",
```

```
        "CostCenter",
        "Department"
    ]
}
```

자세한 내용은 Resource Groups Tagging API 참조의 GetTagKeys를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetTagKeys를 참조하세요.

get-tag-values

다음 코드 예시에서는 `get-tag-values`의 사용 방법을 보여줍니다.

AWS CLI

모든 태그 값 목록 가져오기

다음 `get-tag-values` 예시에서는 모든 리소스의 지정된 키에 사용된 모든 값을 표시합니다.

```
aws resourcegroupstaggingapi get-tag-values \
    --key=Environment
```

출력:

```
{
    "TagValues": [
        "Alpha",
        "Gamma",
        "Production"
    ]
}
```

자세한 내용은 Resource Groups Tagging API 참조의 GetTagValues를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetTagValues를 참조하세요.

tag-resources

다음 코드 예시에서는 `tag-resources`의 사용 방법을 보여줍니다.

AWS CLI

리소스에 태그 연결

다음 `tag-resources` 예시에서는 지정된 리소스에 키 이름 및 값으로 태그를 지정합니다.

```
aws resourcegroupstaggingapi tag-resources \
    --resource-arn-list arn:aws:s3:::MyProductionBucket \
    --tags Environment=Production,CostCenter=1234
```

출력:

```
{
    "FailedResourcesMap": {}
}
```

자세한 내용은 Resource Groups Tagging API 참조의 TagResources를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TagResources를 참조하세요.

untag-resources

다음 코드 예시에서는 `untag-resources`의 사용 방법을 보여줍니다.

AWS CLI

리소스에서 태그 제거

다음 `untag-resources` 예시에서는 지정된 리소스에서 지정된 태그 키 및 관련된 값을 제거합니다.

```
aws resourcegroupstaggingapi untag-resources \
    --resource-arn-list arn:aws:s3:::amzn-s3-demo-bucket \
    --tag-keys Environment CostCenter
```

출력:

```
{
    "FailedResourcesMap": {}
}
```

자세한 내용은 Resource Groups Tagging API 참조의 UntagResources를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResources를 참조하세요.

AWS CLI를 사용한 AWS RoboMaker 예시

다음 코드 예시에서는 AWS RoboMaker에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

batch-describe-simulation-job

다음 코드 예시에서는 batch-describe-simulation-job의 사용 방법을 보여줍니다.

AWS CLI

시뮬레이션 작업 일괄 설명

다음 batch-describe-simulation-job 예시에서는 지정된 시뮬레이션 작업의 세부 정보를 가져옵니다.

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": []
}
```

- API 세부 정보는 AWS CLI 명령 참조의 BatchDescribeSimulationJob을 참조하세요.

`cancel-simulation-job`

다음 코드 예시에서는 `cancel-simulation-job`의 사용 방법을 보여줍니다.

AWS CLI

시뮬레이션 작업 취소

다음 `cancel-simulation-job` 예시에서는 지정된 시뮬레이션 작업을 취소합니다.

```
aws robomaker cancel-simulation-job \
    --job arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-66bbb3gpxm8x
```

- API 세부 정보는 AWS CLI 명령 참조의 CancelSimulationJob을 참조하세요.

`create-deployment-job`

다음 코드 예시에서는 `create-deployment-job`의 사용 방법을 보여줍니다.

AWS CLI

배포 작업 생성

이 예시에서는 플릿 MyFleet의 배포 작업을 생성합니다. 여기에는 'ENVIRONMENT'라는 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateDeploymentJob을 참조하세요.

create-fleet

다음 코드 예시에서는 `create-fleet`의 사용 방법을 보여줍니다.

AWS CLI

플릿 생성

이 예시에서는 플릿을 생성합니다. Region이라는 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateFleet 섹션을 참조하세요.

create-robot-application-version

다음 코드 예시에서는 `create-robot-application-version`의 사용 방법을 보여줍니다.

AWS CLI

로봇 애플리케이션 버전 생성

이 예시에서는 로봇 애플리케이션의 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": "amzn-s3-demo-bucket",
          "s3Key": "my-robot-application.tar.gz",
          "etag": "f8cf5526f1c6e7b3a72c3ed3f79c5493-70",
          "architecture": "ARMHF"
      }
  ],
  "robotSoftwareSuite": {
      "name": "ROS",
      "version": "Kinetic"
  },
  "lastUpdatedAt": 1551201873.0,
  "revisionId": "9986bb8d-a695-4ab4-8810-9f4a74d1aa00"
  "tags": {}
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateRobotApplicationVersion을 참조하세요.

create-robot-application

다음 코드 예시에서는 create-robot-application의 사용 방법을 보여줍니다.

AWS CLI

로봇 애플리케이션 생성

이 예시에서는 로봇 애플리케이션을 생성합니다.

명령:

```
aws robomaker create-robot-application --name MyRobotApplication
 --sources s3Bucket=amzn-s3-demo-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": "amzn-s3-demo-bucket",
          "s3Key": "my-robot-application.tar.gz",
          "architecture": "ARMHF"
      }
  ],
  "robotSoftwareSuite": {
      "name": "ROS",
      "version": "Kinetic"
  },
  "lastUpdatedAt": 1551201873.0,
  "revisionId": "1f3cb539-9239-4841-a656-d3efcffa07e1",
  "tags": {}
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateRobotApplication을 참조하세요.

`create-robot`

다음 코드 예시에서는 `create-robot`의 사용 방법을 보여줍니다.

AWS CLI

로봇 생성

이 예시에서는 로봇을 생성합니다. ARMHF 아키텍처를 사용합니다. 또한 Region이라는 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"
  }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateRobot을 참조하세요.

`create-simulation-application-version`

다음 코드 예시에서는 `create-simulation-application-version`의 사용 방법을 보여줍니다.

AWS CLI

시뮬레이션 애플리케이션 버전 생성

이 예시에서는 로봇 애플리케이션의 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": "amzn-s3-demo-bucket",
          "s3Key": "my-simulation-application.tar.gz",
          "etag": "00d8a94ff113856688c4fce618ae0f45-94",
          "architecture": "X86_64"
      }
  ],
  "simulationSoftwareSuite": {
      "name": "Gazebo",
      "version": "7"
  },
  "robotSoftwareSuite": {
      "name": "ROS",
      "version": "Kinetic"
  },
  "renderingEngine": {
      "name": "OGRE",
      "version": "1.x"
  },
 "lastUpdatedAt": 1551203853.0,
  "revisionId": "ee753e53-519c-4d37-895d-65e79bcd1914",
  "tags": {}
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateSimulationApplicationVersion을 참조하세요.

create-simulation-application

다음 코드 예시에서는 `create-simulation-application`의 사용 방법을 보여줍니다.

AWS CLI

시뮬레이션 애플리케이션 생성

이 예시에서는 시뮬레이션 애플리케이션을 생성합니다.

명령:

```
aws robomaker create-simulation-application  --name MyRobotApplication
 --sources s3Bucket=amzn-s3-demo-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": "amzn-s3-demo-bucket",
          "s3Key": "my-simulation-application.tar.gz",
          "architecture": "X86_64"
      }
  ],
  "simulationSoftwareSuite": {
      "name": "Gazebo",
      "version": "7"
  },
  "robotSoftwareSuite": {
      "name": "ROS",
      "version": "Kinetic"
  },
  "renderingEngine": {
      "name": "OGRE",
      "version": "1.x"
  },
  "lastUpdatedAt": 1551203301.0,
  "revisionId": "ee753e53-519c-4d37-895d-65e79bcd1914",
  "tags": {}
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateSimulationApplication을 참조하세요.

`create-simulation-job`

다음 코드 예시에서는 `create-simulation-job`의 사용 방법을 보여줍니다.

AWS CLI

시뮬레이션 작업 생성

이 예시에서는 시뮬레이션 작업을 생성합니다. 로봇 애플리케이션 및 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateSimulationJob을 참조하세요.

delete-fleet

다음 코드 예시에서는 delete-fleet의 사용 방법을 보여줍니다.

AWS CLI

플릿 삭제

이 예시에서는 플릿을 삭제합니다.

명령:

```
aws robomaker delete-fleet --fleet arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/MyFleet/1550771394395
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteFleet을 참조하세요.

delete-robot-application

다음 코드 예시에서는 delete-robot-application의 사용 방법을 보여줍니다.

AWS CLI

로봇 애플리케이션 삭제

이 예시에서는 로봇 애플리케이션을 삭제합니다.

명령:

```
aws robomaker delete-robot-application --application arn:aws:robomaker:us-west-2:111111111111:robot-application/MyRobotApplication/1551203485821
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteRobotApplication을 참조하세요.

delete-robot

다음 코드 예시에서는 delete-robot의 사용 방법을 보여줍니다.

AWS CLI

로봇 삭제

이 예시에서는 로봇을 삭제합니다.

명령:

```
aws robomaker delete-robot --robot arn:aws:robomaker:us-west-2:111111111111:robot/MyRobot/1540829698778
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteRobot을 참조하세요.

delete-simulation-application

다음 코드 예시에서는 delete-simulation-application의 사용 방법을 보여줍니다.

AWS CLI

시뮬레이션 애플리케이션 삭제

이 예시에서는 시뮬레이션 애플리케이션을 삭제합니다.

명령:

```
aws robomaker delete-simulation-application --application arn:aws:robomaker:us-west-2:111111111111:simulation-application/MySimulationApplication/1551203427605
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteSimulationApplication을 참조하세요.

deregister-robot

다음 코드 예시에서는 deregister-robot의 사용 방법을 보여줍니다.

AWS CLI

플릿에서 로봇 등록 취소

이 예시에서는 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeregisterRobot을 참조하세요.

describe-deployment-job

다음 코드 예시에서는 describe-deployment-job의 사용 방법을 보여줍니다.

AWS CLI

배포 작업 설명

다음 describe-deployment-job 예시에서는 지정된 배포 작업의 세부 정보를 가져옵니다.

```
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": {}
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDeploymentJob을 참조하세요.

describe-fleet

다음 코드 예시에서는 `describe-fleet`의 사용 방법을 보여줍니다.

AWS CLI

플릿 설명

다음 `describe-fleet` 예시에서는 지정된 플릿의 세부 정보를 가져옵니다.

```
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": {}
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeFleet을 참조하세요.

`describe-robot-application`

다음 코드 예시에서는 `describe-robot-application`의 사용 방법을 보여줍니다.

AWS CLI

로봇 애플리케이션 설명

이 예시에서는 로봇 애플리케이션을 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": "amzn-s3-demo-bucket",
          "s3Key": "my-robot-application.tar.gz",
          "architecture": "X86_64"
      }
  ],
  "robotSoftwareSuite": {
      "name": "ROS",
      "version": "Kinetic"
  },
  "revisionId": "e72efe0d-f44f-4333-b604-f6fa5c6bb50b",
  "lastUpdatedAt": 1551203485.0,
  "tags": {}
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeRobotApplication을 참조하세요.

describe-robot

다음 코드 예시에서는 `describe-robot`의 사용 방법을 보여줍니다.

AWS CLI

로봇 설명

이 예시에서는 로봇을 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"
  }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeRobot을 참조하세요.

describe-simulation-application

다음 코드 예시에서는 describe-simulation-application의 사용 방법을 보여줍니다.

AWS CLI

시뮬레이션 애플리케이션 설명

이 예시에서는 시뮬레이션 애플리케이션을 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": "amzn-s3-demo-bucket",
            "s3Key": "my-simulation-application.tar.gz",
            "architecture": "X86_64"
        }
    ],
    "simulationSoftwareSuite": {
        "name": "Gazebo",
        "version": "7"
    },
    "robotSoftwareSuite": {
        "name": "ROS",
        "version": "Kinetic"
    },
    "renderingEngine": {
        "name": "OGRE",
        "version": "1.x"
    },
    "revisionId": "783674ab-b7b8-42d9-b01f-9373907987e5",
    "lastUpdatedAt": 1551203427.0,
    "tags": {}
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeSimulationApplication을 참조하세요.

describe-simulation-job

다음 코드 예시에서는 describe-simulation-job의 사용 방법을 보여줍니다.

AWS CLI

시뮬레이션 작업 설명

이 예시에서는 시뮬레이션 작업을 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": {}
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeSimulationJob을 참조하세요.

`list-deployment-jobs`

다음 코드 예시에서는 `list-deployment-jobs`의 사용 방법을 보여줍니다.

AWS CLI

배포 작업 나열

다음 `list-deployment-jobs` 예시에서는 배포 작업 목록을 가져옵니다.

```
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
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListDeploymentJobs를 참조하세요.

`list-fleets`

다음 코드 예시에서는 `list-fleets`의 사용 방법을 보여줍니다.

AWS CLI

플릿 나열

이 예시에서는 플릿을 나열합니다. 최대 20개의 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
      }
  ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListFleets 섹션을 참조하세요.

list-robot-applications

다음 코드 예시에서는 `list-robot-applications`의 사용 방법을 보여줍니다.

AWS CLI

로봇 애플리케이션 나열

이 예시에서는 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
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListRobotApplications를 참조하세요.

list-robots

다음 코드 예시에서는 `list-robots`의 사용 방법을 보여줍니다.

AWS CLI

로봇 나열

이 예시에서는 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListRobots를 참조하세요.

`list-simulation-applications`

다음 코드 예시에서는 `list-simulation-applications`의 사용 방법을 보여줍니다.

AWS CLI

시뮬레이션 애플리케이션 나열

이 예시에서는 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
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListSimulationApplications를 참조하세요.

list-simulation-jobs

다음 코드 예시에서는 list-simulation-jobs의 사용 방법을 보여줍니다.

AWS CLI

시뮬레이션 작업 나열

이 예시에서는 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"
            ]
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [ListSimulationJobs](#)를 참조하세요.

`list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`의 사용 방법을 보여줍니다.

AWS CLI

리소스의 태그 나열

이 예시에서는 AWS RoboMaker 리소스의 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource 섹션을 참조하세요.

register-robot

다음 코드 예시에서는 register-robot의 사용 방법을 보여줍니다.

AWS CLI

로봇 등록

이 예시에서는 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 RegisterRobot을 참조하세요.

restart-simulation-job

다음 코드 예시에서는 restart-simulation-job의 사용 방법을 보여줍니다.

AWS CLI

시뮬레이션 다시 시작

이 예시에서는 시뮬레이션을 다시 시작합니다.

명령:

```
aws robomaker restart-simulation-job --job arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-t6rdgt70mftr
```

- API 세부 정보는 AWS CLI 명령 참조의 RestartSimulationJob을 참조하세요.

sync-deployment-job

다음 코드 예시에서는 sync-deployment-job의 사용 방법을 보여줍니다.

AWS CLI

배포 작업 동기화

이 예시에서는 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 SyncDeploymentJob을 참조하세요.

tag-resource

다음 코드 예시에서는 tag-resource의 사용 방법을 보여줍니다.

AWS CLI

리소스에 태그 지정

이 예시에서는 리소스에 태그를 지정합니다. Region과 Stage라는 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
```

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

untag-resource

다음 코드 예시에서는 untag-resource 코드를 사용하는 방법을 보여줍니다.

AWS CLI

리소스에서 태그 제거

이 예시에서는 리소스에서 태그를 제거합니다. Region 태그를 제거합니다.

명령:

```
aws robomaker untag-resource --resource-arn "arn:aws:robomaker:us-west-2:111111111111:robot/MyRobot/1544035373264" --tag-keys Region
```

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

update-robot-application

다음 코드 예시에서는 update-robot-application의 사용 방법을 보여줍니다.

AWS CLI

로봇 애플리케이션 업데이트

이 예시에서는 로봇 애플리케이션을 업데이트합니다.

명령:

```
aws robomaker update-robot-application --application arn:aws:robomaker:us-west-2:111111111111:robot-application/MyRobotApplication/1551203485821
 --sources s3Bucket=amzn-s3-demo-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": "amzn-s3-demo-bucket",
          "s3Key": "my-robot-application.tar.gz",
          "architecture": "X86_64"
      }
  ],
  "robotSoftwareSuite": {
      "name": "ROS",
      "version": "Kinetic"
  },
  "lastUpdatedAt": 1551287993.0,
  "revisionId": "20b5e331-24fd-4504-8b8c-531afe5f4c94"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateRobotApplication을 참조하세요.

update-simulation-application

다음 코드 예시에서는 update-simulation-application의 사용 방법을 보여줍니다.

AWS CLI

시뮬레이션 애플리케이션 업데이트

이 예시에서는 시뮬레이션 애플리케이션을 업데이트합니다.

명령:

```
aws robomaker update-simulation-application --application arn:aws:robomaker:us-west-2:111111111111:simulation-application/MySimulationApplication/1551203427605
 --sources s3Bucket=amzn-s3-demo-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": "amzn-s3-demo-bucket",
          "s3Key": "my-simulation-application.tar.gz",
          "architecture": "X86_64"
      }
  ],
  "simulationSoftwareSuite": {
      "name": "Gazebo",
      "version": "7"
  },
  "robotSoftwareSuite": {
      "name": "ROS",
      "version": "Kinetic"
  },
```

```
  "renderingEngine": {
      "name": "OGRE",
      "version": "1.x"
  },
  "lastUpdatedAt": 1551289361.0,
  "revisionId": "4a22cb5d-93c5-4cef-9311-52bdd119b79e"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateSimulationApplication을 참조하세요.

AWS CLI를 사용한 Route 53 예시

다음 코드 예시에서는 Route 53과 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`change-resource-record-sets`

다음 코드 예시에서는 `change-resource-record-sets`의 사용 방법을 보여줍니다.

AWS CLI

리소스 레코드 세트 생성, 업데이트 또는 삭제

다음 `change-resource-record-sets` 명령은 파일 `C:\awscli\route53\change-resource-record-sets.json`에서 hosted-zone-id `Z1R8UBAEXAMPLE` 및 JSON 형식 구성을 사용하여 리소스 레코드 세트를 생성합니다.

```
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 API 참조의 POST ChangeResourceRecordSets를 참조하세요.

JSON 파일의 구성은 생성하려는 리소스 레코드 세트의 종류에 따라 달라집니다.

BasicWeightedAliasWeighted AliasLatencyLatency AliasFailoverFailover Alias

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"
      }
    },
    {...}
  ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ChangeResourceRecordSets를 참조하세요.

change-tags-for-resource

다음 코드 예시에서는 change-tags-for-resource의 사용 방법을 보여줍니다.

AWS CLI

다음 명령은 ID로 지정된 상태 확인 리소스에 owner라는 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 리소스에서 owner라는 태그를 제거합니다.

```
aws route53 change-tags-for-resource --resource-type hostedzone --resource-
id Z1523434445 --remove-tag-keys owner
```

- API 세부 정보는 AWS CLI 명령 참조의 ChangeTagsForResource를 참조하세요.

create-health-check

다음 코드 예시에서는 create-health-check의 사용 방법을 보여줍니다.

AWS CLI

상태 확인 생성

다음 create-health-check 명령은 파일 C:\awscli\route53\create-health-check.json에서 호출자 참조 2014-04-01-18:47 및 JSON 형식 구성을 사용하여 상태 확인을 생성합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 [CreateHealthCheck](https://docs.aws.amazon.com/goto/aws-cli/route53-2013-04-01/CreateHealthCheck)를 참조하세요.

`create-hosted-zone`

다음 코드 예시에서는 `create-hosted-zone`의 사용 방법을 보여줍니다.

AWS CLI

호스팅 영역 생성

다음 `create-hosted-zone` 명령은 호출자 참조 `2014-04-01-18:47`를 사용하여 `example.com`라는 호스팅 영역을 추가합니다. 선택적 주석에는 공백이 포함되므로 주석을 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.

- API 세부 정보는 AWS CLI 명령 참조의 [CreateHostedZone](https://docs.aws.amazon.com/goto/aws-cli/route53-2013-04-01/CreateHostedZone)을 참조하세요.

`delete-health-check`

다음 코드 예시에서는 `delete-health-check`의 사용 방법을 보여줍니다.

AWS CLI

상태 확인 삭제

다음 delete-health-check 명령은 e75b48d9-547a-4c3d-88a5-ae4002397608이라는 health-check-id를 사용하여 상태 확인을 삭제합니다.

```
aws route53 delete-health-check --health-check-id e75b48d9-547a-4c3d-88a5-
ae4002397608
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteHealthCheck를 참조하세요.

delete-hosted-zone

다음 코드 예시에서는 delete-hosted-zone의 사용 방법을 보여줍니다.

AWS CLI

호스팅 영역 삭제

다음 delete-hosted-zone 명령은 Z36KTIQEXAMPLE이라는 id가 있는 호스팅 영역을 삭제합니다.

```
aws route53 delete-hosted-zone --id Z36KTIQEXAMPLE
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteHostedZone을 참조하세요.

get-change

다음 코드 예시에서는 get-change의 사용 방법을 보여줍니다.

AWS CLI

리소스 레코드 세트의 변경 상태 가져오기

다음 get-change 명령은 /change/CWPIK4URU2I5S라는 Id가 있는 change-resource-record-sets 요청의 상태 및 기타 정보를 가져옵니다.

```
aws route53 get-change --id /change/CWPIK4URU2I5S
```

- API 세부 정보는 AWS CLI명령 참조의 GetChange를 참조하세요.

get-health-check

다음 코드 예시에서는 get-health-check의 사용 방법을 보여줍니다.

AWS CLI

상태 확인 정보 가져오기

다음 `get-health-check` 명령은 `02ec8401-9879-4259-91fa-04e66d094674`라는 `health-check-id`가 있는 상태 확인 정보를 가져옵니다.

```
aws route53 get-health-check --health-check-id 02ec8401-9879-4259-91fa-04e66d094674
```

- API 세부 정보는 AWS CLI 명령 참조의 [GetHealthCheck](https://docs.aws.amazon.com/goto/aws-cli/route53-2013-04-01/GetHealthCheck)를 참조하세요.

`get-hosted-zone`

다음 코드 예시에서는 `get-hosted-zone`의 사용 방법을 보여줍니다.

AWS CLI

호스팅 영역 정보 가져오기

다음 `get-hosted-zone` 명령은 `Z1R8UBAEXAMPLE`이라는 `id`가 있는 호스팅 영역의 정보를 가져옵니다.

```
aws route53 get-hosted-zone --id Z1R8UBAEXAMPLE
```

- API 세부 정보는 AWS CLI 명령 참조의 [GetHostedZone](https://docs.aws.amazon.com/goto/aws-cli/route53-2013-04-01/GetHostedZone)을 참조하세요.

`list-health-checks`

다음 코드 예시에서는 `list-health-checks`의 사용 방법을 보여줍니다.

AWS CLI

AWS 계정에 연결된 상태 확인 나열

다음 AWS 명령은 현재 `list-health-checks` 계정에 연결된 첫 100개의 상태 확인에 대한 요약된 정보를 나열합니다.

```
aws route53 list-health-checks
```

100개를 초과한 상태 확인이 있거나 해당 상태 확인을 100개 미만의 그룹으로 나열하려면 --maxitems 파라미터를 포함합니다. 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 응답에서 NextToken의 값을 가져와 --starting-token 파라미터에 포함합니다. 예를 들면 다음과 같습니다.

```
aws route53 list-health-checks --max-items 1 --starting-token Z3M3LMPEXAMPLE
```

- API 세부 정보는 AWS CLI 명령 참조의 ListHealthChecks를 참조하세요.

list-hosted-zones-by-name

다음 코드 예시에서는 list-hosted-zones-by-name의 사용 방법을 보여줍니다.

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 `www.example.com`으로 시작하는 이름을 기준으로 호스팅 영역을 나열합니다.

```
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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListHostedZonesByName을 참조하세요.

`list-hosted-zones`

다음 코드 예시에서는 `list-hosted-zones`의 사용 방법을 보여줍니다.

AWS CLI

AWS 계정에 연결된 호스팅 영역 나열

다음 AWS 명령은 현재 list-hosted-zones 계정과 연결된 첫 100개의 호스팅 영역에 대한 요약된 정보를 나열합니다.

```
aws route53 list-hosted-zones
```

100개를 초과한 호스팅 영역이 있거나 해당 영역을 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
```

- API 세부 정보는 AWS CLI 명령 참조의 [ListHostedZones](https://docs.aws.amazon.com/goto/aws-cli/route53-2013-04-01/ListHostedZones)를 참조하세요.

list-query-logging-configs

다음 코드 예시에서는 list-query-logging-configs의 사용 방법을 보여줍니다.

AWS CLI

쿼리 로깅 구성 나열

다음 list-query-logging-configs 예시에서는 호스팅 영역 Z1OX3WQEXAMPLE에 대한 AWS 계정 내 첫 100개 쿼리 로깅 구성의 정보를 나열합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 ListQueryLoggingConfigs를 참조하세요.

`list-resource-record-sets`

다음 코드 예시에서는 `list-resource-record-sets`의 사용 방법을 보여줍니다.

AWS CLI

호스팅 영역의 리소스 레코드 세트 업데이트

다음 `list-resource-record-sets` 명령은 지정된 호스팅 영역의 첫 100개 리소스 레코드 세트에 대한 요약 정보를 나열합니다.

```
aws route53 list-resource-record-sets --hosted-zone-id Z2LD58HEXAMPLE
```

호스팅 영역에 100개를 초과한 리소스 레코드 세트가 있거나 해당 세트를 100개 미만의 그룹으로 나열하려면 `--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 영역에서 다음 리소스 레코드 세트의 정보를 보려면 이전 명령에 대한 응답에서 `NextToken`의 값을 가져와 `--starting-token` 파라미터에 포함합니다. 예를 들면 다음과 같습니다.

```
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.']"
```

- API 세부 정보는 AWS CLI 명령 참조의 ListResourceRecordSets를 참조하세요.

AWS CLI를 사용한 Route 53 도메인 등록 예시

다음 코드 예시는 Route 53 도메인 등록과 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

check-domain-availability

다음 코드 예시에서는 `check-domain-availability` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Route 53에 도메인 이름을 등록할 수 있는지 확인

다음 `check-domain-availability` 명령은 Route 53를 사용하여 도메인 이름 `example.com`을 등록할 수 있는지 여부에 대한 정보를 반환합니다.

이 명령은 `us-east-1` 리전에서만 실행됩니다. 기본 리전이 `us-east-1`로 설정된 경우, `region` 파라미터를 생략할 수 있습니다.

```
aws route53domains check-domain-availability \
    --region us-east-1 \
    --domain-name example.com
```

출력:

```
{
    "Availability": "UNAVAILABLE"
}
```

Route 53은 `.com` 및 `.jp`와 같은 최상위 도메인(TLD)을 광범위하게 지원하지만, 사용 가능한 모든 TLD를 지원하지는 않습니다. 도메인의 가용성을 확인하고 Route 53이 TLD를 지원하지 않는 경우, `check-domain-availability`는 다음 메시지를 반환합니다.

```
An error occurred (UnsupportedTLD) when calling the CheckDomainAvailability
 operation: <top-level domain> tld is not supported.
```

Route 53에 도메인을 등록하는 데 사용할 수 있는 TLD 목록은 Amazon Route 53 개발자 안내서의 Amazon Route 53에 등록할 수 있는 도메인을 참조하세요. Amazon Route 53에 도메인을 등록하는 방법에 대한 자세한 내용은 Amazon Route 53 개발자 안내서의 새 도메인 등록을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CheckDomainAvailability 섹션을 참조하세요.

`check-domain-transferability`

다음 코드 예시에서는 `check-domain-transferability` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

도메인을 Route 53으로 이전할 수 있는지 확인

다음 `check-domain-transferability` 명령은 도메인 이름 `example.com`을 Route 53으로 이전할 수 있는지 여부에 대한 정보를 반환합니다.

이 명령은 `us-east-1` 리전에서만 실행됩니다. 기본 리전이 `us-east-1`로 설정된 경우, `region` 파라미터를 생략할 수 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 도메인 등록을 Amazon Route 53으로 이전을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CheckDomainTransferability 섹션을 참조하세요.

`delete-tags-for-domain`

다음 코드 예시에서는 `delete-tags-for-domain` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

도메인의 태그를 삭제하는 방법

다음 `delete-tags-for-domain` 명령은 지정된 도메인에서 세 개의 태그를 삭제합니다. 단, 태그 값이 아닌 태그 키만 지정합니다.

이 명령은 `us-east-1` 리전에서만 실행됩니다. 기본 리전이 `us-east-1`로 설정된 경우, `region` 파라미터를 생략할 수 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 을 실행할 수 있습니다. 자세한 내용은 Amazon Route 53 개발자 안내서의 Tagging Amazon Route 53 Resources를 참조하세요

- API 세부 정보는 AWS CLI 명령 참조의 DeleteTagsForDomain 섹션을 참조하세요.

`disable-domain-auto-renew`

다음 코드 예시에서는 `disable-domain-auto-renew` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

도메인 자동 갱신 비활성화

다음 `disable-domain-auto-renew` 명령은 도메인 등록이 만료되기 전에 도메인 `example.com`을 자동으로 갱신하지 않도록 Route 53을 구성합니다.

이 명령은 us-east-1 리전에서만 실행됩니다. 기본 리전이 us-east-1로 설정된 경우, region 파라미터를 생략할 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 을 실행할 수 있습니다. 자동 갱신이 비활성화된 경우, AutoRenew의 값은 False입니다. 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.

- API 세부 정보는 AWS CLI 명령 참조의 DisableDomainAutoRenew 섹션을 참조하세요.

disable-domain-transfer-lock

다음 코드 예시에서는 disable-domain-transfer-lock 코드를 사용하는 방법을 보여줍니다.

AWS CLI

도메인에서 전송 잠금을 비활성화하는 방법

다음 disable-domain-transfer-lock 명령은 도메인을 다른 등록 기관으로 이전할 수 있도록 도메인 example.com의 이전 잠금을 제거합니다. 이 명령은 clientTransferProhibited 상태를 변경합니다.

이 명령은 us-east-1 리전에서만 실행됩니다. 기본 리전이 us-east-1로 설정된 경우, region 파라미터를 생략할 수 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 을 실행할 수 있습니다. 이전 잠금이 비활성화되면, StatusList의 값에 clientTransferProhibited가 포함되지 않습니다.

이전 프로세스에 대한 자세한 내용은 Amazon Route 53 개발자 안내서의 도메인을 Amazon Route 53에서 다른 등록 기관으로 이전을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisableDomainTransferLock 섹션을 참조하세요.

enable-domain-auto-renew

다음 코드 예시에서는 enable-domain-auto-renew 코드를 사용하는 방법을 보여줍니다.

AWS CLI

도메인 자동 갱신 활성화

다음 enable-domain-auto-renew 명령은 도메인 example.com의 등록이 만료되기 전에 도메인을 자동으로 갱신하도록 Route 53을 구성합니다.

이 명령은 us-east-1 리전에서만 실행됩니다. 기본 리전이 us-east-1로 설정된 경우, region 파라미터를 생략할 수 있습니다.

```
aws route53domains enable-domain-auto-renew \
    --region us-east-1 \
    --domain-name example.com
```

이 명령은 출력을 생성하지 않습니다. 설정이 변경되었는지 확인하기 위해 get-domain-detail 을 실행할 수 있습니다. 자동 갱신이 활성화된 경우, AutoRenew의 값은 True입니다.

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.

- API 세부 정보는 AWS CLI 명령 참조의 EnableDomainAutoRenew 섹션을 참조하세요.

enable-domain-transfer-lock

다음 코드 예시에서는 enable-domain-transfer-lock 코드를 사용하는 방법을 보여줍니다.

AWS CLI

도메인에서 전송 잠금을 활성화하는 방법

다음 enable-domain-transfer-lock 명령은 지정된 도메인을 잠가 다른 등록 기관으로 이전할 수 없도록 합니다. 이 명령은 clientTransferProhibited 상태를 변경합니다.

이 명령은 us-east-1 리전에서만 실행됩니다. 기본 리전이 us-east-1로 설정된 경우, region 파라미터를 생략할 수 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 확인하기 위해 get-domain-detail 을 실행할 수 있습니다. 이전 잠금이 활성화되면 StatusList의 값에 clientTransferProhibited가 포함됩니다.

이전 프로세스에 대한 자세한 내용은 Amazon Route 53 개발자 안내서의 도메인을 Amazon Route 53에서 다른 등록 기관으로 이전을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 EnableDomainTransferLock 섹션을 참조하세요.

get-contact-reachability-status

다음 코드 예시에서는 get-contact-reachability-status 코드를 사용하는 방법을 보여줍니다.

AWS CLI

등록자 연락 담당자가 확인 이메일에 응답했는지 확인하는 방법

다음 get-contact-reachability-status 명령은 지정된 도메인의 등록자 연락처가 확인 이메일에 응답했는지 여부에 대한 정보를 반환합니다.

이 명령은 us-east-1 리전에서만 실행됩니다. 기본 리전이 us-east-1로 설정된 경우, region 파라미터를 생략할 수 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 Resending Authorization and Confirmation Emails를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetContactReachabilityStatus 섹션을 참조하세요.

get-domain-detail

다음 코드 예시에서는 `get-domain-detail` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

지정된 도메인에 대한 자세한 정보를 가져오는 방법

다음 `get-domain-detail` 명령은 지정된 도메인의 자세한 정보를 표시합니다.

이 명령은 `us-east-1` 리전에서만 실행됩니다. 기본 리전이 `us-east-1`로 설정된 경우, `region` 파라미터를 생략할 수 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",
        "RegistrarUrl": "http://registrar.amazon.com",
        "AbuseContactEmail": "abuse@registrar.amazon.com",
        "AbuseContactPhone": "+1.2062661000",
        "CreationDate": 1444934889.601,
        "ExpirationDate": 1602787689.0,
        "StatusList": [
            "clientTransferProhibited"
        ]
    }
```

- API 세부 정보는 AWS CLI 명령 참조의 GetDomainDetail 섹션을 참조하세요.

get-domain-suggestions

다음 코드 예시에서는 get-domain-suggestions 코드를 사용하는 방법을 보여줍니다.

AWS CLI

제안된 도메인 이름 목록을 가져오는 방법

다음 get-domain-suggestions 명령은 도메인 이름 example.com에 따라 제안된 도메인 이름 목록을 표시합니다. 응답에는 사용 가능한 도메인 이름만 포함됩니다. 이 명령은 us-east-1 리전에서만 실행됩니다. 기본 리전이 us-east-1로 설정된 경우, region 파라미터를 생략할 수 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [GetDomainSuggestions](#) 섹션을 참조하세요.

`get-operation-detail`

다음 코드 예시에서는 `get-operation-detail` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

작업의 현재 상태를 가져오는 방법

일부 도메인 등록 작업은 비동기적으로 작동하고 완료되기 전에 응답을 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 `us-east-1`로 설정된 경우, `region` 파라미터를 생략할 수 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [GetOperationDetail](#) 섹션을 참조하세요.

`list-domains`

다음 코드 예시에서는 `list-domains` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

현재 AWS 계정에 등록된 도메인을 나열하는 방법

다음 AWS 명령은 현재 `list-domains` 계정에 등록된 도메인에 대한 요약된 정보를 나열합니다.

이 명령은 us-east-1 리전에서만 실행됩니다. 기본 리전이 us-east-1로 설정된 경우, region 파라미터를 생략할 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
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListDomains 섹션을 참조하세요.

list-operations

다음 코드 예시에서는 list-operations 코드를 사용하는 방법을 보여줍니다.

AWS CLI

작업 ID를 반환하는 작업의 상태를 나열하는 방법

일부 도메인 등록 작업은 비동기적으로 실행되고 완료되기 전에 응답을 반환합니다. 이러한 작업은 현재 상태를 가져오는 데 사용할 수 있는 작업 ID를 반환합니다. 다음 `list-operations` 명령은 상태를 포함한 현재 도메인 등록 작업에 대한 요약 정보를 나열합니다.

이 명령은 `us-east-1` 리전에서만 실행됩니다. 기본 리전이 `us-east-1`로 설정된 경우, `region` 파라미터를 생략할 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 UTC(협정 세계시)로 지정할 수 있습니다. 다음 명령은 2020년 1월 1일 오전 12:00 UTC 이후에 제출된 모든 작업의 상태를 가져옵니다.

```
aws route53domains list-operations \
    --submitted-since 1577836800
```

- API 세부 정보는 AWS CLI 명령 참조의 [ListOperations](https://docs.aws.amazon.com/goto/aws-cli/route53domains-2014-05-15/ListOperations) 섹션을 참조하세요.

`list-tags-for-domain`

다음 코드 예시에서는 `list-tags-for-domain` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

도메인의 태그를 나열하는 방법

다음 `list-tags-for-domain` 명령은 지정된 도메인에 현재 연결된 태그를 나열합니다.

이 명령은 `us-east-1` 리전에서만 실행됩니다. 기본 리전이 `us-east-1`로 설정된 경우, `region` 파라미터를 생략할 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 Tagging Amazon Route 53 Resources를 참조하세요

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForDomain 섹션을 참조하세요.

`register-domain`

다음 코드 예시에서는 `register-domain` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

도메인 등록

다음 `register-domain` 명령은 도메인을 등록하여 JSON 형식 파일에서 모든 파라미터 값을 가져옵니다.

이 명령은 `us-east-1` 리전에서만 실행됩니다. 기본 리전이 `us-east-1`로 설정된 경우, `region` 파라미터를 생략할 수 있습니다.

```
aws route53domains register-domain \
    --region us-east-1 \
    --cli-input-json file://register-domain.json
```

`register-domain.json`의 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-domain-detail을 참조하세요.

자세한 내용은 Amazon Route 53 개발자 안내서의 새 도메인 등록을 참조하세요.

`ExtraParams`의 값이 필요한 최상위 도메인(TLD)과 유효한 값에 대한 자세한 내용은 Amazon Route 53 API 참조의 ExtraParam을 참조하세요.

- API 세부 정보는 AWS CLI API 참조의 RegisterDomain 섹션을 참조하세요.

renew-domain

다음 코드 예시에서는 renew-domain 코드를 사용하는 방법을 보여줍니다.

AWS CLI

도메인을 갱신하는 방법

다음 renew-domain 명령은 지정된 도메인을 5년 동안 갱신합니다. current-expiry-year의 값을 가져오려면 get-domain-detail 명령을 사용하고 Unix 형식에서 ExpirationDate 값을 변환합니다.

이 명령은 us-east-1 리전에서만 실행됩니다. 기본 리전이 us-east-1로 설정된 경우, region 파라미터를 생략할 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 get-operation-detail을 실행할 수 있습니다. 자세한 내용은 get-operation-detail을 참조하세요.

.com 또는 .org와 같은 각 최상위 도메인(TLD)의 레지스트리는 도메인을 갱신할 수 있는 최대 연도를 제어합니다. 도메인의 최대 갱신 기간을 확인하려면 Amazon Route 53 개발자 안내서의 Amazon Route 53에 등록할 수 있는 도메인에서 TLD에 대한 '등록 및 갱신 기간' 섹션을 참조하세요.

자세한 내용은 Amazon Route 53 개발자 안내서의 Renewing Registration for a Domain을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [SendCommand](#) 섹션을 참조하세요.

`resend-contact-reachability-email`

다음 코드 예시에서는 `resend-contact-reachability-email` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

등록자 연락처의 현재 이메일 주소로 확인 이메일 재전송

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 `us-east-1`로 설정된 경우, `region` 파라미터를 생략할 수 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 예시에서와 같이 `true`의 값이 `isAlreadyVerified`인 경우, 등록자 연락처가 지정된 이메일 주소에 연결할 수 있음을 이미 확인한 것입니다.

자세한 내용은 Amazon Route 53 개발자 안내서의 [Resending Authorization and Confirmation Emails](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [AuthorizeSecurityGroupIngress](#) 섹션을 참조하세요.

`retrieve-domain-auth-code`

다음 코드 예시에서는 `retrieve-domain-auth-code` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

도메인을 다른 등록 기관으로 전송할 수 있도록 도메인에 대한 권한 부여 코드를 가져오는 방법

다음 `retrieve-domain-auth-code` 명령은 example.com 도메인의 현재 인증 코드를 가져옵니다. 도메인을 다른 등록 기관으로 이전하려는 경우 해당 도메인 등록 기관에 이 값을 제공합니다.

이 명령은 `us-east-1` 리전에서만 실행됩니다. 기본 리전이 `us-east-1`로 설정된 경우, `region` 파라미터를 생략할 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 Transferring a Domain from Amazon Route 53 to Another Registrar을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RetrieveDomainAuthCode 섹션을 참조하세요.

`transfer-domain`

다음 코드 예시에서는 `transfer-domain` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

도메인을 Amazon Route 53으로 이전

다음 `transfer-domain` 명령은 JSON 형식 파일 `C:\temp\transfer-domain.json`에서 제공하는 파라미터와 함께 도메인을 Route 53으로 이전합니다.

이 명령은 `us-east-1` 리전에서만 실행됩니다. 기본 리전이 `us-east-1`로 설정된 경우, `region` 파라미터를 생략할 수 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 있습니다`get-operation-detail`. 자세한 내용은 get-domain-detail을 참조하세요.

자세한 내용은 Amazon Route 53 개발자 안내서의 도메인 등록을 Amazon Route 53으로 이전을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TransferDomain 섹션을 참조하세요.

`update-domain-contact-privacy`

다음 코드 예시에서는 `update-domain-contact-privacy` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

도메인의 연락처에 대한 개인 정보 보호 설정을 업데이트하는 방법

다음 update-domain-contact-privacy 명령은 example.com 도메인의 관리 연락처에 대한 개인 정보 보호를 해제합니다. 이 명령은 us-east-1 리전에서만 실행됩니다.

기본 리전이 us-east-1로 설정된 경우, region 파라미터를 생략할 수 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-domain-detail을 참조하세요.

자세한 내용은 Amazon Route 53 개발자 안내서의 Enabling or Disabling Privacy Protection for Contact Information for a Domain을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateAccountPasswordPolicy 섹션을 참조하세요.

update-domain-contact

다음 코드 예시에서는 update-domain-contact 코드를 사용하는 방법을 보여줍니다.

AWS CLI

도메인 연락처 정보 업데이트

다음 update-domain-contact 명령은 도메인의 연락처 정보를 업데이트하여 JSON 형식 파일 C:\temp\update-domain-contact.json에서 파라미터를 가져옵니다.

이 명령은 us-east-1 리전에서만 실행됩니다. 기본 리전이 us-east-1로 설정된 경우, region 파라미터를 생략할 수 있습니다.

```
aws route53domains update-domain-contact \
    --region us-east-1 \
    --cli-input-json file://C:\temp\update-domain-contact.json
```

`update-domain-contact.json`의 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-detail을 실행할 수 있습니다. 자세한 내용은 Amazon Route 53 개발자 안내서의 Updating Contact Information for a Domain을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdatePatchBaseline 섹션을 참조하세요.

update-domain-nameservers

다음 코드 예시에서는 update-domain-nameservers 코드를 사용하는 방법을 보여줍니다.

AWS CLI

도메인의 이름 서버 업데이트

다음 update-domain-nameservers 명령은 도메인의 이름 서버를 업데이트합니다.

이 명령은 us-east-1 리전에서만 실행됩니다. 기본 리전이 us-east-1로 설정된 경우, region 파라미터를 생략할 수 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-detail을 실행할 수 있습니다.

자세한 내용은 Amazon Route 53 개발자 안내서의 Adding or Changing Name Servers and Glue Records for a Domain을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateAssociationStatus 섹션을 참조하세요.

update-tags-for-domain

다음 코드 예시에서는 `update-tags-for-domain` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

도메인에 대한 태그를 추가하거나 업데이트하는 방법

다음 `update-tags-for-domain` 명령은 example.com 도메인에 대한 두 개의 키와 그에 해당하는 값을 추가하거나 업데이트합니다. 키 값을 업데이트하는 방법 키와 새 값을 포함하면 됩니다. 한 번에 하나의 도메인에서만 태그를 추가하거나 업데이트할 수 있습니다.

이 명령은 `us-east-1` 리전에서만 실행됩니다. 기본 리전이 `us-east-1`로 설정된 경우, `region` 파라미터를 생략할 수 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 확인하기 위해 list-tags-for-domain 을 실행할 수 있습니다.

자세한 내용은 Amazon Route 53 개발자 안내서의 Tagging Amazon Route 53 Resources를 참조하세요

- API 세부 정보는 AWS CLI 명령 참조의 UpdatePatchBaseline 섹션을 참조하세요.

view-billing

다음 코드 예시에서는 `view-billing` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

현재 AWS 계정의 도메인 등록 요금에 대한 결제 정보를 가져오는 방법

다음 view-billing 명령은 2018년 1월 1일(1514764800 Unix 시간)부터 2019년 12월 31일 자정 (1577836800 Unix 시간)까지의 기간 동안 현재 계정의 모든 도메인 관련 결제 레코드를 반환합니다.

이 명령은 us-east-1 리전에서만 실행됩니다. 기본 리전이 us-east-1로 설정된 경우, region 파라미터를 생략할 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 API 참조의 ViewBilling을 확인하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ViewBilling 섹션을 참조하세요.

AWS CLI를 사용한 Route 53 Profiles 예시

다음 코드 예시에서는 Route 53 Profiles와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

associate-profile

다음 코드 예시에서는 associate-profile의 사용 방법을 보여줍니다.

AWS CLI

프로파일 연결

다음 associate-profile 예시에서는 프로파일을 VPC에 연결합니다.

```
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 프로파일 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AssociateProfile을 참조하세요.

associate-resource-to-profile

다음 코드 예시에서는 associate-resource-to-profile의 사용 방법을 보여줍니다.

AWS CLI

리소스를 프로파일에 연결

다음 associate-resource-to-profile 예시에서는 우선순위가 102인 DNS 방화벽 규칙 그룹을 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 AssociateResourceToProfile을 참조하세요.

`create-profile`

다음 코드 예시에서는 `create-profile`의 사용 방법을 보여줍니다.

AWS CLI

프로파일 생성

다음 `create-profile` 예시에서는 프로파일을 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateProfile을 참조하세요.

`delete-profile`

다음 코드 예시에서는 `delete-profile`의 사용 방법을 보여줍니다.

AWS CLI

프로파일 삭제

다음 `delete-profile` 예시에서는 프로파일을 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteProfile을 참조하세요.

disassociate-profile

다음 코드 예시에서는 disassociate-profile의 사용 방법을 보여줍니다.

AWS CLI

프로파일 연결 해제

다음 disassociate-profile 예시에서는 VPC에서 프로파일의 연결을 해제합니다.

```
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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DisassociateProfile을 참조하세요.

`disassociate-resource-from-profile`

다음 코드 예시에서는 `disassociate-resource-from-profile`의 사용 방법을 보여줍니다.

AWS CLI

프로파일에서 리소스 연결 해제

다음 `disassociate-resource-from-profile` 예시에서는 프로파일에서 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DisassociateResourceFromProfile을 참조하세요.

get-profile-association

다음 코드 예시에서는 `get-profile-association`의 사용 방법을 보여줍니다.

AWS CLI

프로파일 연결 정보 가져오기

다음 `get-profile-association`은 지정된 프로파일 연결의 정보를 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetProfileAssociation을 참조하세요.

`get-profile-resource-association`

다음 코드 예시에서는 `get-profile-resource-association`의 사용 방법을 보여줍니다.

AWS CLI

프로파일에 연결된 리소스 정보 가져오기

다음 `get-profile-resource-association`은 프로파일에 대한 지정된 리소스 연결의 정보를 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [GetProfileResourceAssociation](#)을 참조하세요.

`get-profile`

다음 코드 예시에서는 `get-profile`의 사용 방법을 보여줍니다.

AWS CLI

프로파일 정보 가져오기

다음 get-profile은 지정된 프로파일의 정보를 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetProfile을 참조하세요.

list-profile-associations

다음 코드 예시에서는 list-profile-associations의 사용 방법을 보여줍니다.

AWS CLI

프로파일 연결 나열

다음 list-profile-associations는 AWS 계정의 프로파일 연결을 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListProfileAssociations를 참조하세요.

list-profile-resource-associations

다음 코드 예시에서는 list-profile-resource-associations의 사용 방법을 보여줍니다.

AWS CLI

프로파일 리소스 연결 나열

다음 list-profile-resource-associations는 지정된 프로파일에 대한 프로파일 리소스 연결을 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"
            }
        ]
 }
```

- API 세부 정보는 AWS CLI 명령 참조의 ListProfileResourceAssociations를 참조하세요.

`list-profiles`

다음 코드 예시에서는 `list-profiles`의 사용 방법을 보여줍니다.

AWS CLI

프로파일 나열

다음 `list-profiles`는 AWS 계정의 프로파일을 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"
         }
     ]
 }
```

- API 세부 정보는 AWS CLI 명령 참조의 ListProfiles를 참조하세요.

`list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`의 사용 방법을 보여줍니다.

AWS CLI

리소스의 태그 나열

다음 `list-tags-for-resource`는 지정된 리소스의 태그를 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource 섹션을 참조하세요.

`update-profile-resource-association`

다음 코드 예시에서는 `update-profile-resource-association`의 사용 방법을 보여줍니다.

AWS CLI

프로파일에 연결된 리소스 업데이트

다음 `update-profile-resource-association`은 프로파일에 연결된 DNS 방화벽 규칙 그룹의 우선 순위를 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateProfileResourceAssociation을 참조하세요.

AWS CLI를 사용하여 Route 53 Resolver 예시

다음 코드 예시에서는 Route 53 Resolver에서 AWS Command Line Interface를 사용하여 일반적인 시나리오를 구현하고 개별 서비스 작업을 수행하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`associate-firewall-rule-group`

다음 코드 예시에서는 `associate-firewall-rule-group`의 사용 방법을 보여줍니다.

AWS CLI

방화벽 규칙 그룹 VPC에 연결

다음 `associate-firewall-rule-group` 예시에서는 DNS 방화벽 규칙 그룹을 Amazon VPC와 연결합니다.

```
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 Managing associations between your VPC and Route 53 Resolver DNS Firewall rule groups를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AssociateFirewallRuleGroup 섹션을 참조하세요.

`associate-resolver-endpoint-ip-address`

다음 코드 예시에서는 `associate-resolver-endpoint-ip-address`의 사용 방법을 보여줍니다.

AWS CLI

다른 IP 주소를 Resolver 엔드포인트에 연결하는 방법

다음 `associate-resolver-endpoint-ip-address` 예시에서는 다른 IP 주소를 인바운드 Resolver 엔드포인트에 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 Values That You Specify When You Create or Edit Inbound Endpoints를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AssociateResolverEndpointIpAddress 섹션을 참조하세요.

`associate-resolver-rule`

다음 코드 예시에서는 `associate-resolver-rule`의 사용 방법을 보여줍니다.

AWS CLI

Resolver 규칙을 VPC에 연결

다음 `associate-resolver-rule` 예시에서는 Resolver 규칙을 Amazon VPC에 연결합니다. 명령을 실행한 후 Resolver는 전달된 쿼리의 도메인 이름과 같은 규칙의 설정을 기반으로 DNS 쿼리를 네트워크에 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 [Forwarding Outbound DNS Queries to Your Network](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [AssociateResolverRule](#) 섹션을 참조하세요.

`create-firewall-domain-list`

다음 코드 예시에서는 `create-firewall-domain-list`의 사용 방법을 보여줍니다.

AWS CLI

Route 53 Resolver DNS 방화벽 도메인 목록 생성

다음 `create-firewall-domain-list` 예시에서는 AWS 계정에 test라는 Route 53 Resolver DNS 방화벽 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 Managing your own domain lists를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateFirewallDomainList 섹션을 참조하세요.

`create-firewall-rule-group`

다음 코드 예시에서는 `create-firewall-rule-group`의 사용 방법을 보여줍니다.

AWS CLI

방화벽 규칙 그룹 생성

다음 `create-firewall-rule-group` 예시에서는 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 Managing rule groups and rules in DNS Firewall을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateFirewallRuleGroup 섹션을 참조하세요.

`create-firewall-rule`

다음 코드 예시에서는 `create-firewall-rule`의 사용 방법을 보여줍니다.

AWS CLI

방화벽 규칙 생성

다음 `create-firewall-rule` 예시에서는 DNS 방화벽 도메인 목록에 나열된 도메인에 대한 DNS 방화벽 규칙에 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 Managing rule groups and rules in DNS Firewall을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateFirewallRule 섹션을 참조하세요.

`create-resolver-endpoint`

다음 코드 예시에서는 `create-resolver-endpoint`의 사용 방법을 보여줍니다.

AWS CLI

인바운드 Resolver 엔드포인트 생성

다음 `create-resolver-endpoint` 예시에서는 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 엔드포인트 생성

다음 `create-resolver-endpoint` 예시에서는 JSON 형식 문서 `create-outbound-resolver-endpoint.json`의 값을 사용하여 아웃바운드 Resolver 엔드포인트를 생성합니다.

```
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 VPC와 네트워크 간 DNS 쿼리 해결을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateResolverEndpoint 섹션을 참조하세요.

create-resolver-rule

다음 코드 예시에서는 `create-resolver-rule`의 사용 방법을 보여줍니다.

AWS CLI

Resolver 규칙 생성

다음 `create-resolver-rule` 예시에서는 Resolver 전달 규칙을 생성합니다. 이 규칙은 아웃바운드 엔드포인트 rslvr-out-d5e5920e37example을 사용하여 `example.com`에 대한 DNS 쿼리를 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."
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [CreateResolverRule](https://docs.aws.amazon.com/goto/aws-cli/route53resolver-2018-04-01/CreateResolverRule) 섹션을 참조하세요.

`delete-firewall-domain-list`

다음 코드 예시에서는 `delete-firewall-domain-list`의 사용 방법을 보여줍니다.

AWS CLI

Route 53 Resolver DNS 방화벽 도메인 목록 삭제

다음 `delete-firewall-domain-list` 예시에서는 AWS 계정에서 test라는 Route 53 Resolver DNS 방화벽 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 Managing your own domain lists를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteFirewallDomainList 섹션을 참조하세요.

`delete-firewall-rule-group`

다음 코드 예시에서는 `delete-firewall-rule-group`의 사용 방법을 보여줍니다.

AWS CLI

방화벽 규칙 그룹 삭제

다음 `delete-firewall-rule-group` 예시에서는 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 Managing rule groups and rules in DNS Firewall을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteFirewallRuleGroup 섹션을 참조하세요.

delete-firewall-rule

다음 코드 예시에서는 delete-firewall-rule의 사용 방법을 보여줍니다.

AWS CLI

방화벽 규칙을 삭제하는 방법

다음 delete-firewall-rule 예시에서는 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 Managing rule groups and rules in DNS Firewall을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteFirewallRule](https://docs.aws.amazon.com/goto/aws-cli/route53resolver-2018-04-01/DeleteFirewallRule) 섹션을 참조하세요.

`delete-resolver-endpoint`

다음 코드 예시에서는 `delete-resolver-endpoint`의 사용 방법을 보여줍니다.

AWS CLI

Resolver 엔드포인트를 삭제하는 방법

다음 `delete-resolver-endpoint` 예시에서는 지정된 엔드포인트를 삭제합니다.

중요 인바운드 엔드포인트를 삭제하면 더 이상 네트워크의 DNS 쿼리가 엔드포인트에 지정된 VPC의 Resolver로 전달되지 않습니다. 아웃바운드 엔드포인트를 삭제하면 Resolver는 삭제된 아웃바운드 엔드포인트를 지정하는 규칙의 DNS 쿼리를 더 이상 VPC에서 네트워크로 전달하지 않습니다.

```
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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteResolverEndpoint 섹션을 참조하세요.

delete-resolver-rule

다음 코드 예시에서는 delete-resolver-rule의 사용 방법을 보여줍니다.

AWS CLI

해석기 규칙을 삭제하는 방법

다음 delete-resolver-rule 예시에서는 지정된 규칙을 삭제합니다.

참고 규칙이 VPC에 연결되어 있으면 규칙을 삭제하기 전에 VPC에서 규칙의 연결을 해제해야 합니다.

```
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"
    }
```

```
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteResolverRule 섹션을 참조하세요.

disassociate-firewall-rule-group

다음 코드 예시에서는 `disassociate-firewall-rule-group`의 사용 방법을 보여줍니다.

AWS CLI

방화벽 규칙 그룹을 VPC에서 연결 해제하는 방법

다음 `disassociate-firewall-rule-group` 예시에서는 Amazon VPC에서 DNS 방화벽 규칙 그룹을 연결 해제합니다.

```
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 Managing associations between your VPC and Route 53 Resolver DNS Firewall rule groups를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisassociateFirewallRuleGroup 섹션을 참조하세요.

disassociate-resolver-endpoint-ip-address

다음 코드 예시에서는 disassociate-resolver-endpoint-ip-address의 사용 방법을 보여줍니다.

AWS CLI

Resolver 엔드포인트에서 IP 주소 연결을 해제하는 방법

다음 disassociate-resolver-endpoint-ip-address 예시에서는 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DisassociateResolverEndpointIpAddress 섹션을 참조하세요.

disassociate-resolver-rule

다음 코드 예시에서는 `disassociate-resolver-rule`의 사용 방법을 보여줍니다.

AWS CLI

Amazon VPC에서 Resolver 규칙을 연결 해제하는 방법

다음 `disassociate-resolver-rule` 예시에서는 지정된 Resolver 규칙과 지정된 VPC 간의 연결을 제거합니다. 다음과 같은 경우 VPC에서 규칙의 연결을 해제할 수 있습니다.

이 VPC에서 시작된 DNS 쿼리의 경우 규칙에 지정된 도메인 이름의 쿼리를 네트워크로 전달하는 Resolver의 작업을 중지하려고 합니다. 규칙이 현재 하나 이상의 VPC에 연결된 경우 규칙을 삭제하기 전에 모든 VPC에서 규칙의 연결을 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DisassociateResolverRule 섹션을 참조하세요.

get-firewall-config

다음 코드 예시에서는 `get-firewall-config`의 사용 방법을 보여줍니다.

AWS CLI

VPC에 대한 방화벽 구성을 가져오는 방법

다음 `get-firewall-config` 예시에서는 지정된 VPC에 대한 DNS 방화벽 동작을 가져옵니다.

```
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 Firewall VPC configuration을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetFirewallConfig 섹션을 참조하세요.

`get-firewall-domain-list`

다음 코드 예시에서는 `get-firewall-domain-list`의 사용 방법을 보여줍니다.

AWS CLI

Route 53 Resolver DNS 방화벽 도메인 목록 가져오기

다음 `get-firewall-domain-list` 예시에서는 지정된 ID로 도메인 목록을 가져옵니다.

```
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 Managing your own domain lists를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetFirewallDomainList 섹션을 참조하세요.

get-firewall-rule-group-association

다음 코드 예시에서는 get-firewall-rule-group-association의 사용 방법을 보여줍니다.

AWS CLI

방화벽 규칙 그룹 연결을 가져오는 방법

다음 get-firewall-rule-group-association 예시에서는 방화벽 규칙 그룹 연결을 가져옵니다.

```
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 Managing associations between your VPC and Route 53 Resolver DNS Firewall rule groups를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetFirewallRuleGroupAssociation 섹션을 참조하세요.

get-firewall-rule-group-policy

다음 코드 예시에서는 get-firewall-rule-group-policy의 사용 방법을 보여줍니다.

AWS CLI

AWS IAM 정책 가져오기

다음 get-firewall-rule-group-policy 예시에서는 지정된 규칙 그룹을 공유하기 위한 AWS Identity and Access Management(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 Managing rule groups and rules in DNS Firewall을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetFirewallRuleGroupPolicy 섹션을 참조하세요.

get-firewall-rule-group

다음 코드 예시에서는 get-firewall-rule-group의 사용 방법을 보여줍니다.

AWS CLI

방화벽 규칙 그룹을 가져오는 방법

다음 get-firewall-rule-group 예시에서는 제공된 ID로 DNS 방화벽 규칙 그룹의 정보를 가져옵니다.

```
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 Managing rule groups and rules in DNS Firewall을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetFirewallRuleGroup 섹션을 참조하세요.

get-resolver-endpoint

다음 코드 예시에서는 get-resolver-endpoint의 사용 방법을 보여줍니다.

AWS CLI

Resolver 엔드포인트에 대한 정보를 가져오는 방법

다음 `get-resolver-endpoint` 예시에서는 지정된 아웃바운드 엔드포인트의 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 Values That You Specify When You Create or Edit Inbound Endpoints를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetResolverEndpoint 섹션을 참조하세요.

get-resolver-rule-association

다음 코드 예시에서는 `get-resolver-rule-association`의 사용 방법을 보여줍니다.

AWS CLI

Resolver 규칙과 VPC 간 연결 정보 가져오기

다음 get-resolver-rule-association 예시에서는 지정된 Resolver 규칙과 VPC 간의 연결에 대한 세부 정보를 표시합니다. associate-resolver-rule을 사용하여 Resolver 규칙과 VPC를 연결합니다.

```
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": ""
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetResolverRuleAssociation 섹션을 참조하세요.

get-resolver-rule

다음 코드 예시에서는 get-resolver-rule의 사용 방법을 보여줍니다.

AWS CLI

Resolver 규칙 정보 가져오기

다음 get-resolver-rule 예시에서는 DNS 쿼리를 전달하는 규칙과 관련된 도메인 이름 및 규칙이 연결된 아웃바운드 Resolver 엔드포인트의 ID와 같은 지정된 Resolver 규칙의 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 Values That You Specify When You Create or Edit Rules를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetResolverRule 섹션을 참조하세요.

import-firewall-domains

다음 코드 예시에서는 `import-firewall-domains`의 사용 방법을 보여줍니다.

AWS CLI

도메인을 도메인 목록으로 가져오는 방법

다음 `import-firewall-domains` 예시에서는 도메인 세트를 파일에서 지정된 DNS 방화벽 도메인 목록으로 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 Managing your own domain lists를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ImportFirewallDomains 섹션을 참조하세요.

list-firewall-configs

다음 코드 예시에서는 `list-firewall-configs`의 사용 방법을 보여줍니다.

AWS CLI

방화벽 구성을 나열하는 방법

다음 `list-firewall-configs` 예시에서는 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 Firewall VPC configuration을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListFirewallConfigs 섹션을 참조하세요.

list-firewall-domain-lists

다음 코드 예시에서는 list-firewall-domain-lists의 사용 방법을 보여줍니다.

AWS CLI

Route 53 Resolver DNS 방화벽 도메인 목록 모두 나열

다음 list-firewall-domain-lists 예시에서는 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 domain lists를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListFirewallDomainLists 섹션을 참조하세요.

list-firewall-domains

다음 코드 예시에서는 `list-firewall-domains`의 사용 방법을 보여줍니다.

AWS CLI

도메인 목록에 도메인을 나열하는 방법

다음 `list-firewall-domains` 예시에서는 지정된 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 Managing your own domain lists를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListFirewallDomains 섹션을 참조하세요.

list-firewall-rule-group-associations

다음 코드 예시에서는 `list-firewall-rule-group-associations`의 사용 방법을 보여줍니다.

AWS CLI

DNS 방화벽 규칙 그룹 연결 나열

다음 `list-firewall-rule-group-associations` 예시에서는 Amazon VPC와 DNS 방화벽 규칙 그룹의 연결을 나열합니다.

```
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 Managing associations between your VPC and Route 53 Resolver DNS Firewall rule group을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListFirewallRuleGroupAssociations 섹션을 참조하세요.

list-firewall-rule-groups

다음 코드 예시에서는 `list-firewall-rule-groups`의 사용 방법을 보여줍니다.

AWS CLI

방화벽 규칙 그룹 목록을 가져오는 방법

다음 `list-firewall-rule-groups` 예시에서는 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 Managing rule groups and rules in DNS Firewall을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListFirewallRuleGroups 섹션을 참조하세요.

list-firewall-rules

다음 코드 예시에서는 `list-firewall-rules`의 사용 방법을 보여줍니다.

AWS CLI

방화벽 규칙을 나열하는 방법

다음 `list-firewall-rules` 예시에서는 방화벽 규칙 그룹 내의 모든 DNS 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 Managing rule groups and rules in DNS Firewall을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListFirewallRules 섹션을 참조하세요.

list-resolver-endpoint-ip-addresses

다음 코드 예시에서는 `list-resolver-endpoint-ip-addresses`의 사용 방법을 보여줍니다.

AWS CLI

지정된 인바운드 또는 아웃바운드 엔드포인트의 IP 주소를 나열하는 방법

다음 `list-resolver-endpoint-ip-addresses` 예시에서는 인바운드 엔드포인트 `rslvr-in-f9ab8a03f1example`에 연결된 IP 주소의 정보를 나열합니다. 해당 엔드포인트 ID를 지정하여 아웃바운드 엔드포인트에 `list-resolver-endpoint-ip-addresses`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"
            }
        ]
}
```

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.

- API 세부 정보는 AWS CLI 명령 참조의 ListResolverEndpointIpAddresses 섹션을 참조하세요.

list-resolver-endpoints

다음 코드 예시에서는 `list-resolver-endpoints`의 사용 방법을 보여줍니다.

AWS CLI

AWS 리전에서 Resolver 엔드포인트를 나열하는 방법

다음 `list-resolver-endpoints` 예시에서는 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListResolverEndpoints 섹션을 참조하세요.

`list-resolver-rule-associations`

다음 코드 예시에서는 `list-resolver-rule-associations`의 사용 방법을 보여줍니다.

AWS CLI

Resolver 규칙과 VPC 간 연결 나열

다음 `list-resolver-rule-associations` 예시에서는 현재 AWS 계정의 Resolver 규칙과 VPC 간의 연결을 나열합니다.

```
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 How Route 53 Resolver Forwards DNS Queries from Your VPCs to Your Network를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListResolverRuleAssociations 섹션을 참조하세요.

`list-resolver-rules`

다음 코드 예시에서는 `list-resolver-rules`의 사용 방법을 보여줍니다.

AWS CLI

해석기 규칙을 나열하는 방법

다음 `list-resolver-rules` 예시에서는 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 How Route 53 Resolver Forwards DNS Queries from Your VPCs to Your Network를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListResolverRules 섹션을 참조하세요.

list-tags-for-resource

다음 코드 예시에서는 `list-tags-for-resource`의 사용 방법을 보여줍니다.

AWS CLI

Resolver 리소스의 태그를 나열하는 방법

다음 `list-tags-for-resource` 예시에서는 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 할당 태그를 사용하는 방법에 대한 자세한 내용은 AWS 결제 및 비용 관리 사용자 안내서의 비용 할당 태그 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource를 참조하세요.

put-firewall-rule-group-policy

다음 코드 예시에서는 put-firewall-rule-group-policy의 사용 방법을 보여줍니다.

AWS CLI

방화벽 규칙 그룹 정책을 공유하기 위해 AWS IAM 정책을 연결하는 방법

다음 put-firewall-rule-group-policy 예시에서는 규칙 그룹을 공유하기 위한 AWS Identity and Access Management(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 Managing rule groups and rules in DNS Firewall을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutFirewallRuleGroupPolicy 섹션을 참조하세요.

put-resolver-rule-policy

다음 코드 예시에서는 put-resolver-rule-policy의 사용 방법을 보여줍니다.

AWS CLI

해석기 규칙을 다른 AWS 계정과 공유하는 방법

다음 put-resolver-rule-policy 예시에서는 다른 AWS 계정과 공유하려는 Resolver 규칙, 규칙을 공유하려는 계정, 해당 계정에서 규칙에 따라 수행할 수 있게 하려는 규칙 관련 작업을 지정합니다.

참고 규칙을 생성한 것과 동일한 계정의 자격 증명을 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
}
```

`put-resolver-rule-policy`를 실행한 후 다음 두 개의 Resource Access Manager(RAM) 명령을 실행할 수 있습니다. 규칙을 공유하려는 계정을 사용해야 합니다.

`get-resource-share-invitations`가 값 `resourceShareInvitationArn`을 반환합니다. 공유 규칙을 사용하도록 초대를 수락하려면 이 값이 필요합니다. `accept-resource-share-invitation`은 공유 규칙을 사용하도록 초대를 수락합니다.

자세한 내용은 다음 설명서를 참조하세요.

Amazon Route 53 개발자 안내서의 get-resource-share-invitations accept-resource-share-invitations 다른 AWS 계정과 전달 규칙 공유 및 공유 규칙 사용

- API 세부 정보는 AWS CLI 명령 참조의 PutResolverRulePolicy 섹션을 참조하세요.

tag-resource

다음 코드 예시에서는 `tag-resource`의 사용 방법을 보여줍니다.

AWS CLI

Resolver 리소스에 태그를 연결하는 방법

다음 `tag-resource` 예시에서는 두 태그 키/값 페어를 지정된 Resolver 규칙에 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 할당 태그를 사용하는 방법에 대한 자세한 내용은 AWS 결제 및 비용 관리 사용자 안내서의 비용 할당 태그 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

`untag-resource`

다음 코드 예시에서는 `untag-resource`의 사용 방법을 보여줍니다.

AWS CLI

Resolver 리소스에서 태그 제거

다음 `untag-resource` 예시에서는 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 확인하기 위해 list-tags-for-resource 를 사용할 수 있습니다.

비용 할당 태그를 사용하는 방법에 대한 자세한 내용은 AWS 결제 및 비용 관리 사용자 안내서의 비용 할당 태그 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

update-firewall-config

다음 코드 예시에서는 update-firewall-config의 사용 방법을 보여줍니다.

AWS CLI

방화벽 구성을 업데이트하는 방법

다음 update-firewall-config 예시에서는 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 Firewall VPC configuration을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateFirewallConfig 섹션을 참조하세요.

update-firewall-domains

다음 코드 예시에서는 update-firewall-domains의 사용 방법을 보여줍니다.

AWS CLI

도메인 목록을 업데이트하는 방법

다음 update-firewall-domains 예시에서는 제공된 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 Managing your own domain lists를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateFirewallDomains 섹션을 참조하세요.

`update-firewall-rule-group-association`

다음 코드 예시에서는 `update-firewall-rule-group-association`의 사용 방법을 보여줍니다.

AWS CLI

방화벽 규칙 그룹 연결을 업데이트하는 방법

다음 `update-firewall-rule-group-association` 예시에서는 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 Managing associations between your VPC and Route 53 Resolver DNS Firewall rule group을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateFirewallRuleGroupAssociation 섹션을 참조하세요.

`update-firewall-rule`

다음 코드 예시에서는 `update-firewall-rule`의 사용 방법을 보여줍니다.

AWS CLI

방화벽 규칙을 업데이트하는 방법

다음 `update-firewall-rule` 예시에서는 방화벽 규칙을 지정된 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 Managing rule groups and rules in DNS Firewall을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateFirewallRule 섹션을 참조하세요.

update-resolver-endpoint

다음 코드 예시에서는 update-resolver-endpoint의 사용 방법을 보여줍니다.

AWS CLI

Resolver 엔드포인트의 이름을 업데이트하는 방법

다음 update-resolver-endpoint 예시에서는 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"
    }
```

```
}
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateResolverEndpoint 섹션을 참조하세요.

update-resolver-rule

다음 코드 예시에서는 update-resolver-rule의 사용 방법을 보여줍니다.

AWS CLI

예시 1: Resolver 엔드포인트의 설정 업데이트

다음 update-resolver-rule 예시에서는 규칙의 이름, DNS 쿼리가 전달되는 온프레미스 네트워크의 IP 주소, 네트워크에 쿼리를 전달하는 데 사용하는 아웃바운드 Resolver 엔드포인트의 ID를 업데이트합니다.

참고 TargetIps의 기존 값은 덮어쓰기되므로 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"
    }
}
```

예시 2: `config` 설정에 대한 파일을 사용하여 Resolver 엔드포인트의 설정 업데이트

또는 JSON 파일에 `config` 설정을 포함시킨 다음 `update-resolver-rule`를 직접적으로 호출할 때 해당 파일을 지정할 수 있습니다.

```
aws route53resolver update-resolver-rule \
    --resolver-rule-id rslvr-rr-1247fa64f3example \
    --config file://c:\temp\update-resolver-rule.json
```

`update-resolver-rule.json`의 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 Values That You Specify When You Create or Edit Rules를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateResolverRule 섹션을 참조하세요.

AWS CLI를 사용한 Amazon S3 예시

다음 코드 예시는 Amazon S3와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

작업

abort-multipart-upload

다음 코드 예시에서는 abort-multipart-upload의 사용 방법을 보여줍니다.

AWS CLI

지정된 멀티파트 업로드 중단

다음 abort-multipart-upload 명령은 amzn-s3-demo-bucket 버킷의 multipart/01 키에 대한 멀티파트 업로드를 중단합니다.

```
aws s3api abort-multipart-upload \
    --bucket amzn-s3-demo-bucket \
    --key multipart/01 \
    --upload-
id dfRtDYU0WWCCcH43C3WFbkRONycyCpTJJvxu2i5GYkZljF.Yxwh6XG7WfS2vC4to6HiV6Yjlx.cph0gtNBtJ8P3UR
```

이 명령에 필요한 업로드 ID는 create-multipart-upload로 출력되며 list-multipart-uploads를 사용하여 가져올 수도 있습니다.

- API 세부 정보는 AWS CLI 명령 참조의 AbortMultipartUpload를 참조하세요.

complete-multipart-upload

다음 코드 예시에서는 complete-multipart-upload의 사용 방법을 보여줍니다.

AWS CLI

다음 명령은 버킷 amzn-s3-demo-bucket의 키 multipart/01에 대한 멀티파트 업로드를 완료합니다.

```
aws s3api complete-multipart-upload --multipart-upload file://
mpustruct --bucket amzn-s3-demo-bucket --key 'multipart/01' --upload-
id dfRtDYU0WWCCcH43C3WFbkRONycyCpTJJvxu2i5GYkZljF.Yxwh6XG7WfS2vC4to6HiV6Yjlx.cph0gtNBtJ8P3UR
```

이 명령에 필요한 업로드 ID는 create-multipart-upload로 출력되며 list-multipart-uploads를 사용하여 가져올 수도 있습니다.

위 명령의 멀티파트 업로드 옵션은 전체 파일로 리어셈블해야 하는 멀티파트 업로드 부분을 설명하는 JSON 구조를 사용합니다. 이 예시에서는 file:// 접두사를 사용하여 mpustruct라는 로컬 폴더에 있는 파일에서 JSON 구조를 로드합니다.

mpustruct:

```
{
  "Parts": [
    {
      "ETag": "e868e0f4719e394144ef36531ee6824c",
      "PartNumber": 1
    },
    {
      "ETag": "6bb2b12753d66fe86da4998aa33fffb0",
      "PartNumber": 2
    },
    {
      "ETag": "d0a0112e841abec9c9ec83406f0159c8",
      "PartNumber": 3
    }
  ]
}
```

업로드되는 각 파트의 ETag 값은 upload-part 명령을 사용하여 파트를 업로드할 때마다 출력되며, list-parts를 직접 호출하거나 각 파트의 MD5 체크섬으로 계산할 수도 있습니다.

출력:

```
{
    "ETag": "\"3944a9f7a4faab7f78788ff6210f63f0-3\"",
```

```
    "Bucket": "amzn-s3-demo-bucket",
    "Location": "https://amzn-s3-demo-bucket.s3.amazonaws.com/multipart%2F01",
    "Key": "multipart/01"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CompleteMultipartUpload를 참조하세요.

copy-object

다음 코드 예시에서는 copy-object의 사용 방법을 보여줍니다.

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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CopyObject를 참조하세요.

cp

다음 코드 예시에서는 cp의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 로컬 파일 S3에 복사

다음 cp 명령은 단일 파일을 지정된 버킷 및 키에 복사합니다.

```
aws s3 cp test.txt s3://amzn-s3-demo-bucket/test2.txt
```

출력:

```
upload: test.txt to s3://amzn-s3-demo-bucket/test2.txt
```

예시 2: 만료 날짜가 있는 로컬 파일을 S3에 복사

다음 cp 명령은 지정된 ISO 8601 타임스탬프에서 만료되는 지정된 버킷 및 키에 단일 파일을 복사합니다.

```
aws s3 cp test.txt s3://amzn-s3-demo-bucket/test2.txt \
    --expires 2014-10-01T20:30:00Z
```

출력:

```
upload: test.txt to s3://amzn-s3-demo-bucket/test2.txt
```

예시 3: S3에서 S3로 파일 복사

다음 cp 명령은 단일 s3 객체를 지정된 버킷 및 키에 복사합니다.

```
aws s3 cp s3://amzn-s3-demo-bucket/test.txt s3://amzn-s3-demo-bucket/test2.txt
```

출력:

```
copy: s3://amzn-s3-demo-bucket/test.txt to s3://amzn-s3-demo-bucket/test2.txt
```

예시 4: Amazon S3 객체를 로컬 파일에 복사

다음 cp 명령은 단일 객체를 지정된 파일에 로컬로 복사합니다.

```
aws s3 cp s3://amzn-s3-demo-bucket/test.txt test2.txt
```

출력:

```
download: s3://amzn-s3-demo-bucket/test.txt to test2.txt
```

예시 5: 한 버킷에서 다른 버킷으로 객체 복사

다음 cp 명령은 원래 이름을 유지하면서 단일 객체를 지정된 버킷에 복사합니다.

```
aws s3 cp s3://amzn-s3-demo-bucket/test.txt s3://amzn-s3-demo-bucket2/
```

출력:

```
copy: s3://amzn-s3-demo-bucket/test.txt to s3://amzn-s3-demo-bucket2/test.txt
```

예시 6: S3 객체를 로컬 디렉터리에 반복적으로 복사

`--recursive` 파라미터와 함께 전달되면 다음 `cp` 명령은 지정된 접두사 및 버킷에 있는 모든 객체를 지정된 디렉터리에 반복적으로 복사합니다. 이 예시에서는 `amzn-s3-demo-bucket` 버킷에 `test1.txt` 및 `test2.txt` 객체가 있습니다.

```
aws s3 cp s3://amzn-s3-demo-bucket . \
    --recursive
```

출력:

```
download: s3://amzn-s3-demo-bucket/test1.txt to test1.txt
download: s3://amzn-s3-demo-bucket/test2.txt to test2.txt
```

예시 7: 로컬 파일을 S3에 반복적으로 복사

`--recursive` 파라미터와 함께 전달되면 다음 `cp` 명령은 `--exclude` 파라미터를 사용하여 일부 파일을 제외하면서 지정된 디렉터리에 있는 모든 파일을 지정된 버킷 및 접두사에 반복적으로 복사합니다. 이 예시에서는 `myDir` 디렉터리에 `test1.txt` 및 `test2.jpg` 파일이 있습니다.

```
aws s3 cp myDir s3://amzn-s3-demo-bucket/ \
    --recursive \
    --exclude "*.jpg"
```

출력:

```
upload: myDir/test1.txt to s3://amzn-s3-demo-bucket/test1.txt
```

예시 8: S3 객체를 다른 버킷에 반복적으로 복사

`--recursive` 파라미터와 함께 전달되면 다음 `cp` 명령은 `--exclude` 파라미터를 사용하여 일부 객체를 제외하면서 지정된 버킷에 있는 모든 객체를 다른 버킷에 반복적으로 복사합니다. 이 예시에서는 `amzn-s3-demo-bucket` 버킷에 `test1.txt` 및 `another/test1.txt` 객체가 있습니다.

```
aws s3 cp s3://amzn-s3-demo-bucket/ s3://amzn-s3-demo-bucket2/ \
    --recursive \
    --exclude "another/*"
```

출력:

```
copy: s3://amzn-s3-demo-bucket/test1.txt to s3://amzn-s3-demo-bucket2/test1.txt
```

`--exclude` 및 `--include` 옵션을 결합하여 다른 모든 객체를 제외하고 패턴과 일치하는 객체만 복사할 수 있습니다.

```
aws s3 cp s3://amzn-s3-demo-bucket/logs/ s3://amzn-s3-demo-bucket2/logs/ \
    --recursive \
    --exclude "*" \
    --include "*.log"
```

출력:

```
copy: s3://amzn-s3-demo-bucket/logs/test/test.log to s3://amzn-s3-demo-bucket2/logs/
test/test.log
copy: s3://amzn-s3-demo-bucket/logs/test3.log to s3://amzn-s3-demo-bucket2/logs/
test3.log
```

예시 9: S3 객체를 복사하는 동안 액세스 제어 목록(ACL) 설정

다음 `cp` 명령은 ACL을 `public-read-write`로 설정하는 동안 단일 객체를 지정된 버킷 및 키에 복사합니다.

```
aws s3 cp s3://amzn-s3-demo-bucket/test.txt s3://amzn-s3-demo-bucket/test2.txt \
    --acl public-read-write
```

출력:

```
copy: s3://amzn-s3-demo-bucket/test.txt to s3://amzn-s3-demo-bucket/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:::amzn-s3-demo-bucket/*"
                ],
                "Effect": "Allow",
                "Sid": "Stmt1234567891234"
            }
        ]
    }
}
```

예시 10: S3 객체에 대한 권한 부여

다음 cp 명령은 --grants 옵션을 사용하여 URI로 식별된 모든 사용자에게 읽기 액세스 권한을 부여하고 표준 ID로 식별된 특정 사용자에게 전체 제어를 부여하는 방법을 보여줍니다.

```
aws s3 cp file.txt s3://amzn-s3-demo-bucket/ --grants read=uri=http://
acs.amazonaws.com/groups/global/
AllUsers full=id=79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
```

출력:

```
upload: file.txt to s3://amzn-s3-demo-bucket/file.txt
```

예시 11: S3에 로컬 파일 스트림 업로드

PowerShell은 의 인코딩을 변경하거나 파이프 입력에 CRLF를 추가할 수 있습니다.

다음 cp 명령은 표준 입력에서 지정된 버킷 및 키로 로컬 파일 스트림을 업로드합니다.

```
aws s3 cp - s3://amzn-s3-demo-bucket/stream.txt
```

예시 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://amzn-s3-demo-bucket/stream.txt --expected-size 54760833024
```

예시 13: 로컬 파일 스트림으로 S3 객체 다운로드

PowerShell은 의 인코딩을 변경하거나 파이프 또는 리디렉션된 출력에 CRLF를 추가할 수 있습니다.

다음 cp 명령은 S3 객체를 스트림으로 로컬에서 표준 출력으로 다운로드합니다. 스트림으로 다운로드는 현재 --recursive 파라미터와 호환되지 않습니다.

```
aws s3 cp s3://amzn-s3-demo-bucket/stream.txt -
```

예시 14: S3 액세스 포인트에 업로드

다음 cp 명령은 키(mykey)의 액세스 포인트(myaccesspoint)에 단일 파일(mydoc.txt)을 업로드합니다.

```
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
```

예시 15: S3 액세스 포인트에서 다운로드

다음 cp 명령은 액세스 포인트(myaccesspoint)에서 로컬 파일(mydoc.txt)로 단일 객체(mykey)를 다운로드합니다.

```
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 세부 정보는 AWS CLI 명령 참조의 Cp 섹션을 참조하세요.

`create-bucket`

다음 코드 예시에서는 `create-bucket`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 버킷 생성

다음 `create-bucket` 예시에서는 `amzn-s3-demo-bucket`이라는 버킷을 생성합니다.

```
aws s3api create-bucket \
    --bucket amzn-s3-demo-bucket \
    --region us-east-1
```

출력:

```
{
    "Location": "/amzn-s3-demo-bucket"
}
```

자세한 내용은 Amazon S3 사용자 안내서의 버킷 생성을 참조하세요.

예시 2: 소유자가 적용된 버킷 생성

다음 `create-bucket` 예시에서는 S3 객체 소유권에 대해 버킷 소유자 적용 설정을 사용하는 `amzn-s3-demo-bucket`이라는 버킷을 생성합니다.

```
aws s3api create-bucket \
    --bucket amzn-s3-demo-bucket \
    --region us-east-1 \
```

```
    --object-ownership BucketOwnerEnforced
```

출력:

```
{
    "Location": "/amzn-s3-demo-bucket"
}
```

자세한 내용은 Amazon S3 사용자 안내서의 객체 소유권 제어 및 ACL 비활성화를 참조하세요.

예시 3: 'us-east-1' 리전 외부에서 버킷 생성

다음 `create-bucket` 예시에서는 `eu-west-1` 리전에서 `amzn-s3-demo-bucket`이라는 버킷을 생성합니다. `us-east-1` 외부 리전의 경우 원하는 리전에 버킷을 생성하려면 적절한 `LocationConstraint`를 지정해야 합니다.

```
aws s3api create-bucket \
    --bucket amzn-s3-demo-bucket \
    --region eu-west-1 \
    --create-bucket-configuration LocationConstraint=eu-west-1
```

출력:

```
{
    "Location": "http://amzn-s3-demo-bucket.s3.amazonaws.com/"
}
```

자세한 내용은 Amazon S3 사용자 안내서의 버킷 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateBucket을 참조하세요.

`create-multipart-upload`

다음 코드 예시에서는 `create-multipart-upload`의 사용 방법을 보여줍니다.

AWS CLI

다음 명령은 키 `multipart/01`를 사용하여 버킷 `amzn-s3-demo-bucket`에 멀티파트 업로드를 생성합니다.

```
aws s3api create-multipart-upload --bucket amzn-s3-demo-bucket --key 'multipart/01'
```

출력:

```
{
    "Bucket": "amzn-s3-demo-bucket",
    "UploadId":
 "dfRtDYU0WWCCcH43C3WFbkRONycyCpTJJvxu2i5GYkZljF.Yxwh6XG7WfS2vC4to6HiV6Yjlx.cph0gtNBtJ8P3URC
    "Key": "multipart/01"
}
```

완성된 파일은 이름이 01이며 amzn-s3-demo-bucket 버킷의 multipart 폴더에 있습니다. upload-part 명령과 함께 사용할 업로드 ID, 키, 버킷 이름을 저장합니다.

- API 세부 정보는 AWS CLI 명령 참조의 CreateMultipartUpload를 참조하세요.

delete-bucket-analytics-configuration

다음 코드 예시에서는 delete-bucket-analytics-configuration의 사용 방법을 보여줍니다.

AWS CLI

버킷의 분석 구성 삭제

다음 delete-bucket-analytics-configuration 예시에서는 지정된 버킷 및 ID에 대한 분석 구성을 제거합니다.

```
aws s3api delete-bucket-analytics-configuration \
    --bucket amzn-s3-demo-bucket \
    --id 1
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteBucketAnalyticsConfiguration을 참조하세요.

delete-bucket-cors

다음 코드 예시에서는 delete-bucket-cors의 사용 방법을 보여줍니다.

AWS CLI

다음 명령은 amzn-s3-demo-bucket이라는 버킷에서 Cross-Origin Resource Sharing 구성을 삭제합니다.

```
aws s3api delete-bucket-cors --bucket amzn-s3-demo-bucket
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteBucketCors를 참조하세요.

delete-bucket-encryption

다음 코드 예시에서는 delete-bucket-encryption의 사용 방법을 보여줍니다.

AWS CLI

버킷의 서버 측 암호화 구성 삭제

다음 delete-bucket-encryption 예시에서는 지정된 버킷의 서버 측 암호화 구성을 삭제합니다.

```
aws s3api delete-bucket-encryption \
    --bucket amzn-s3-demo-bucket
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteBucketEncryption을 참조하세요.

delete-bucket-intelligent-tiering-configuration

다음 코드 예시에서는 delete-bucket-intelligent-tiering-configuration의 사용 방법을 보여줍니다.

AWS CLI

버킷에서 S3 Intelligent-Tiering 구성을 제거하는 방법

다음 delete-bucket-intelligent-tiering-configuration 예시에서는 버킷에서 ExampleConfig라는 S3 Intelligent-Tiering 구성을 제거합니다.

```
aws s3api delete-bucket-intelligent-tiering-configuration \
    --bucket amzn-s3-demo-bucket \
    --id ExampleConfig
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon S3 사용 설명서의 S3 Intelligent-Tiering 사용 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteBucketIntelligentTieringConfiguration 섹션을 참조하세요.

delete-bucket-inventory-configuration

다음 코드 예시에서는 delete-bucket-inventory-configuration의 사용 방법을 보여줍니다.

AWS CLI

버킷의 인벤토리 구성 삭제

다음 delete-bucket-inventory-configuration 예시에서는 지정된 버킷에 대해 ID가 1인 인벤토리 구성을 삭제합니다.

```
aws s3api delete-bucket-inventory-configuration \
    --bucket amzn-s3-demo-bucket \
    --id 1
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteBucketInventoryConfiguration을 참조하세요.

delete-bucket-lifecycle

다음 코드 예시에서는 delete-bucket-lifecycle의 사용 방법을 보여줍니다.

AWS CLI

다음 명령은 amzn-s3-demo-bucket이라는 버킷에서 수명 주기 구성을 삭제합니다.

```
aws s3api delete-bucket-lifecycle --bucket amzn-s3-demo-bucket
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteBucketLifecycle을 참조하세요.

delete-bucket-metrics-configuration

다음 코드 예시에서는 delete-bucket-metrics-configuration의 사용 방법을 보여줍니다.

AWS CLI

버킷의 지표 구성 삭제

다음 delete-bucket-metrics-configuration 예시에서는 지정된 버킷 및 ID에 대한 지표 구성을 제거합니다.

```
aws s3api delete-bucket-metrics-configuration \
    --bucket amzn-s3-demo-bucket \
    --id 123
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteBucketMetricsConfiguration을 참조하세요.

delete-bucket-ownership-controls

다음 코드 예시에서는 delete-bucket-ownership-controls의 사용 방법을 보여줍니다.

AWS CLI

버킷의 버킷 소유권 설정을 제거하는 방법

다음 delete-bucket-ownership-controls 예시에서는 버킷의 버킷 소유권 설정을 제거합니다.

```
aws s3api delete-bucket-ownership-controls \
    --bucket amzn-s3-demo-bucket
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon S3 사용 설명서의 기존 버킷에 대한 객체 소유권 설정 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteBucketOwnershipControls 섹션을 참조하세요.

delete-bucket-policy

다음 코드 예시에서는 delete-bucket-policy의 사용 방법을 보여줍니다.

AWS CLI

다음 명령은 amzn-s3-demo-bucket이라는 버킷에서 버킷 정책을 삭제합니다.

```
aws s3api delete-bucket-policy --bucket amzn-s3-demo-bucket
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteBucketPolicy를 참조하세요.

delete-bucket-replication

다음 코드 예시에서는 delete-bucket-replication의 사용 방법을 보여줍니다.

AWS CLI

다음 명령은 amzn-s3-demo-bucket이라는 버킷에서 복제 구성을 삭제합니다.

```
aws s3api delete-bucket-replication --bucket amzn-s3-demo-bucket
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteBucketReplication을 참조하세요.

delete-bucket-tagging

다음 코드 예시에서는 delete-bucket-tagging의 사용 방법을 보여줍니다.

AWS CLI

다음 명령은 amzn-s3-demo-bucket이라는 버킷에서 태그 지정 구성을 삭제합니다.

```
aws s3api delete-bucket-tagging --bucket amzn-s3-demo-bucket
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteBucketTagging을 참조하세요.

delete-bucket-website

다음 코드 예시에서는 delete-bucket-website의 사용 방법을 보여줍니다.

AWS CLI

다음 명령은 amzn-s3-demo-bucket이라는 버킷에서 웹 사이트 구성을 삭제합니다.

```
aws s3api delete-bucket-website --bucket amzn-s3-demo-bucket
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteBucketWebsite를 참조하세요.

delete-bucket

다음 코드 예시에서는 delete-bucket의 사용 방법을 보여줍니다.

AWS CLI

다음 명령은 amzn-s3-demo-bucket이라는 버킷을 삭제합니다.

```
aws s3api delete-bucket --bucket amzn-s3-demo-bucket --region us-east-1
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteBucket을 참조하세요.

delete-object-tagging

다음 코드 예시에서는 delete-object-tagging의 사용 방법을 보여줍니다.

AWS CLI

객체의 태그 세트 삭제

다음 delete-object-tagging 예시에서는 지정된 키가 있는 태그를 객체 doc1.rtf에서 삭제합니다.

```
aws s3api delete-object-tagging \
    --bucket amzn-s3-demo-bucket \
    --key doc1.rtf
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteObjectTagging을 참조하세요.

delete-object

다음 코드 예시에서는 delete-object의 사용 방법을 보여줍니다.

AWS CLI

다음 명령은 amzn-s3-demo-bucket이라는 버킷에서 test.txt라는 객체를 삭제합니다.

```
aws s3api delete-object --bucket amzn-s3-demo-bucket --key test.txt
```

버킷 버전 관리가 활성화된 경우 출력에는 삭제 마커의 버전 ID가 포함됩니다.

```
{
```

```
  "VersionId": "9_gKg5vG56F.TTEUdwkxGpJ3tNDlWlGq",
  "DeleteMarker": true
}
```

객체를 삭제하는 방법에 대한 자세한 내용은 Amazon S3 개발자 안내서의 객체 삭제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteBucket을 참조하세요.

delete-objects

다음 코드 예시에서는 `delete-objects`의 사용 방법을 보여줍니다.

AWS CLI

다음 명령은 `amzn-s3-demo-bucket`이라는 버킷에서 객체를 삭제합니다.

```
aws s3api delete-objects --bucket amzn-s3-demo-bucket --delete file://delete.json
```

`delete.json`은 삭제할 객체를 지정하는 현재 디렉터리의 JSON 문서입니다.

```
{
  "Objects": [
    {
      "Key": "test1.txt"
    }
  ],
  "Quiet": false
}
```

출력:

```
{
    "Deleted": [
        {
            "DeleteMarkerVersionId": "mYAT5Mc6F7aeUL8SS7FAAqUPO1koHwzU",
            "Key": "test1.txt",
            "DeleteMarker": true
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteObjects](https://docs.aws.amazon.com/goto/aws-cli/s3-2006-03-01/DeleteObjects)를 참조하세요.

`delete-public-access-block`

다음 코드 예시에서는 `delete-public-access-block`의 사용 방법을 보여줍니다.

AWS CLI

버킷의 퍼블릭 액세스 차단 구성 삭제

다음 `delete-public-access-block` 예시에서는 지정된 버킷에서 퍼블릭 액세스 차단 구성을 제거합니다.

```
aws s3api delete-public-access-block \
    --bucket amzn-s3-demo-bucket
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 [DeletePublicAccessBlock](https://docs.aws.amazon.com/goto/aws-cli/s3-2006-03-01/DeletePublicAccessBlock)을 참조하세요.

`get-bucket-accelerate-configuration`

다음 코드 예시에서는 `get-bucket-accelerate-configuration`의 사용 방법을 보여줍니다.

AWS CLI

버킷의 가속화 구성 가져오기

다음 `get-bucket-accelerate-configuration` 예시에서는 지정된 버킷에 대한 가속화 구성을 가져옵니다.

```
aws s3api get-bucket-accelerate-configuration \
    --bucket amzn-s3-demo-bucket
```

출력:

```
{
    "Status": "Enabled"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [GetBucketAccelerateConfiguration](https://docs.aws.amazon.com/goto/aws-cli/s3-2006-03-01/GetBucketAccelerateConfiguration)을 참조하세요.

get-bucket-acl

다음 코드 예시에서는 get-bucket-acl의 사용 방법을 보여줍니다.

AWS CLI

다음 명령은 amzn-s3-demo-bucket이라는 버킷의 액세스 제어 목록을 가져옵니다.

```
aws s3api get-bucket-acl --bucket amzn-s3-demo-bucket
```

출력:

```
{
    "Owner": {
        "DisplayName": "my-username",
        "ID": "7009a8971cd538e11f6b6606438875e7c86c5b672f46db45460ddcd087d36c32"
    },
    "Grants": [
        {
            "Grantee": {
                "DisplayName": "my-username",
                "ID":
 "7009a8971cd538e11f6b6606438875e7c86c5b672f46db45460ddcd087d36c32"
            },
            "Permission": "FULL_CONTROL"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetBucketAcl을 참조하세요.

get-bucket-analytics-configuration

다음 코드 예시에서는 get-bucket-analytics-configuration의 사용 방법을 보여줍니다.

AWS CLI

특정 ID를 가진 버킷의 분석 구성 가져오기

다음 get-bucket-analytics-configuration 예시에서는 지정된 버킷 및 ID에 대한 분석 구성을 표시합니다.

```
aws s3api get-bucket-analytics-configuration \
    --bucket amzn-s3-demo-bucket \
    --id 1
```

출력:

```
{
    "AnalyticsConfiguration": {
        "StorageClassAnalysis": {},
        "Id": "1"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetBucketAnalyticsConfiguration을 참조하세요.

get-bucket-cors

다음 코드 예시에서는 get-bucket-cors의 사용 방법을 보여줍니다.

AWS CLI

다음 명령은 amzn-s3-demo-bucket이라는 버킷에 대한 Cross-Origin Resource Sharing 구성을 가져옵니다.

```
aws s3api get-bucket-cors --bucket amzn-s3-demo-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": [
                "*"
            ]
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetBucketCors를 참조하세요.

get-bucket-encryption

다음 코드 예시에서는 get-bucket-encryption의 사용 방법을 보여줍니다.

AWS CLI

버킷의 서버 측 암호화 구성 가져오기

다음 get-bucket-encryption 예시에서는 amzn-s3-demo-bucket 버킷의 서버 측 암호화 구성을 가져옵니다.

```
aws s3api get-bucket-encryption \
    --bucket amzn-s3-demo-bucket
```

출력:

```
{
    "ServerSideEncryptionConfiguration": {
```

```
        "Rules": [
            {
                "ApplyServerSideEncryptionByDefault": {
                    "SSEAlgorithm": "AES256"
                }
            }
        ]
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetBucketEncryption을 참조하세요.

get-bucket-intelligent-tiering-configuration

다음 코드 예시에서는 get-bucket-intelligent-tiering-configuration의 사용 방법을 보여줍니다.

AWS CLI

버킷에서 S3 Intelligent-Tiering 구성을 검색하는 방법

다음 get-bucket-intelligent-tiering-configuration 예시는 버킷에서 ExampleConfig라는 S3 Intelligent-Tiering 구성을 검색합니다.

```
aws s3api get-bucket-intelligent-tiering-configuration \
    --bucket amzn-s3-demo-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 사용 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetBucketIntelligentTieringConfiguration 섹션을 참조하세요.

get-bucket-inventory-configuration

다음 코드 예시에서는 get-bucket-inventory-configuration의 사용 방법을 보여줍니다.

AWS CLI

버킷의 인벤토리 구성 가져오기

다음 get-bucket-inventory-configuration 예시에서는 지정된 버킷에 대해 ID가 1인 인벤토리 구성을 가져옵니다.

```
aws s3api get-bucket-inventory-configuration \
    --bucket amzn-s3-demo-bucket \
    --id 1
```

출력:

```
{
    "InventoryConfiguration": {
        "IsEnabled": true,
        "Destination": {
            "S3BucketDestination": {
                "Format": "ORC",
                "Bucket": "arn:aws:s3:::amzn-s3-demo-bucket",
                "AccountId": "123456789012"
            }
        },
        "IncludedObjectVersions": "Current",
```

```
            "Id": "1",
            "Schedule": {
                "Frequency": "Weekly"
            }
        }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetBucketInventoryConfiguration을 참조하세요.

get-bucket-lifecycle-configuration

다음 코드 예시에서는 get-bucket-lifecycle-configuration의 사용 방법을 보여줍니다.

AWS CLI

다음 명령은 amzn-s3-demo-bucket이라는 버킷의 수명 주기 구성을 가져옵니다.

```
aws s3api get-bucket-lifecycle-configuration --bucket amzn-s3-demo-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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetBucketLifecycleConfiguration을 참조하세요.

get-bucket-lifecycle

다음 코드 예시에서는 get-bucket-lifecycle의 사용 방법을 보여줍니다.

AWS CLI

다음 명령은 amzn-s3-demo-bucket이라는 버킷의 수명 주기 구성을 가져옵니다.

```
aws s3api get-bucket-lifecycle --bucket amzn-s3-demo-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"
    }
  ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetBucketLifecycle 섹션을 참조하세요.

get-bucket-location

다음 코드 예시에서는 get-bucket-location의 사용 방법을 보여줍니다.

AWS CLI

다음 명령은 amzn-s3-demo-bucket이라는 버킷의 위치 제약 조건을 가져옵니다(제약 조건이 있는 경우).

```
aws s3api get-bucket-location --bucket amzn-s3-demo-bucket
```

출력:

```
{
    "LocationConstraint": "us-west-2"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetBucketLocation을 참조하세요.

get-bucket-logging

다음 코드 예시에서는 get-bucket-logging의 사용 방법을 보여줍니다.

AWS CLI

버킷의 로깅 상태 가져오기

다음 get-bucket-logging 예시에서는 지정된 버킷의 로깅 상태를 가져옵니다.

```
aws s3api get-bucket-logging \
    --bucket amzn-s3-demo-bucket
```

출력:

```
{
    "LoggingEnabled": {
        "TargetPrefix": "",
        "TargetBucket": "amzn-s3-demo-bucket-logs"
```

```
            }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetBucketLogging을 참조하세요.

get-bucket-metrics-configuration

다음 코드 예시에서는 get-bucket-metrics-configuration의 사용 방법을 보여줍니다.

AWS CLI

특정 ID를 가진 버킷의 지표 구성 가져오기

다음 get-bucket-metrics-configuration 예시에서는 지정된 버킷 및 ID의 지표 구성을 표시합니다.

```
aws s3api get-bucket-metrics-configuration \
    --bucket amzn-s3-demo-bucket \
    --id 123
```

출력:

```
{
    "MetricsConfiguration": {
        "Filter": {
            "Prefix": "logs"
        },
        "Id": "123"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetBucketMetricsConfiguration을 참조하세요.

get-bucket-notification-configuration

다음 코드 예시에서는 get-bucket-notification-configuration의 사용 방법을 보여줍니다.

AWS CLI

다음 명령은 amzn-s3-demo-bucket이라는 버킷의 알림 구성을 가져옵니다.

```
aws s3api get-bucket-notification-configuration --bucket amzn-s3-demo-bucket
```

출력:

```
{
    "TopicConfigurations": [
        {
            "Id": "YmQzMmEwM2EjZWVlI0NGItNzVtZjI1MC00ZjgyLWZDBiZWNl",
            "TopicArn": "arn:aws:sns:us-west-2:123456789012:my-notification-topic",
            "Events": [
                "s3:ObjectCreated:*"
            ]
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetBucketInventoryConfiguration 섹션을 참조하세요.

`get-bucket-notification`

다음 코드 예시에서는 `get-bucket-notification`의 사용 방법을 보여줍니다.

AWS CLI

다음 명령은 `amzn-s3-demo-bucket`이라는 버킷의 알림 구성을 가져옵니다.

```
aws s3api get-bucket-notification --bucket amzn-s3-demo-bucket
```

출력:

```
{
    "TopicConfiguration": {
        "Topic": "arn:aws:sns:us-west-2:123456789012:my-notification-topic",
        "Id": "YmQzMmEwM2EjZWVlI0NGItNzVtZjI1MC00ZjgyLWZDBiZWNl",
        "Event": "s3:ObjectCreated:*",
        "Events": [
            "s3:ObjectCreated:*"
        ]
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetBucketNotification을 참조하세요.

get-bucket-ownership-controls

다음 코드 예시에서는 get-bucket-ownership-controls의 사용 방법을 보여줍니다.

AWS CLI

버킷의 버킷 소유권 설정을 검색하는 방법

다음 get-bucket-ownership-controls 예시에서는 버킷의 버킷 소유권 설정을 검색합니다.

```
aws s3api get-bucket-ownership-controls \
    --bucket amzn-s3-demo-bucket
```

출력:

```
{
    "OwnershipControls": {
        "Rules": [
            {
                "ObjectOwnership": "BucketOwnerEnforced"
            }
        ]
    }
}
```

자세한 내용은 Amazon S3 사용 설명서의 S3 버킷에 대한 객체 소유권 설정 보기 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetBucketOwnershipControls 섹션을 참조하세요.

get-bucket-policy-status

다음 코드 예시에서는 get-bucket-policy-status의 사용 방법을 보여줍니다.

AWS CLI

특정 버킷이 퍼블릭인지 나타내는 버킷 정책 상태 가져오기

다음 get-bucket-policy-status 예시에서는 버킷 amzn-s3-demo-bucket의 정책 상태를 가져옵니다.

```
aws s3api get-bucket-policy-status \
    --bucket amzn-s3-demo-bucket
```

출력:

```
{
    "PolicyStatus": {
        "IsPublic": false
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetBucketPolicyStatus를 참조하세요.

get-bucket-policy

다음 코드 예시에서는 get-bucket-policy의 사용 방법을 보여줍니다.

AWS CLI

다음 명령은 amzn-s3-demo-bucket이라는 버킷의 버킷 정책을 가져옵니다.

```
aws s3api get-bucket-policy --bucket amzn-s3-demo-bucket
```

출력:

```
{
    "Policy": "{\"Version\":\"2008-10-17\",\"Statement\":[{\"Sid\":\"\",\"Effect
\":\"Allow\",\"Principal\":\"*\",\"Action\":\"s3:GetObject\",\"Resource\":
\"arn:aws:s3:::amzn-s3-demo-bucket/*\"},{\"Sid\":\"\",\"Effect\":\"Deny\",
\"Principal\":\"*\",\"Action\":\"s3:GetObject\",\"Resource\":\"arn:aws:s3:::amzn-s3-
demo-bucket/secret/*\"}]}"
}
```

버킷 정책 가져오기 및 넣기 다음 예시에서는 Amazon S3 버킷 정책을 다운로드하고 파일을 수정한 다음 put-bucket-policy를 사용하여 수정된 버킷 정책을 적용하는 방법을 보여줍니다. 버킷 정책을 파일로 다운로드하려면 다음을 실행할 수 있습니다.

```
aws s3api get-bucket-policy --bucket amzn-s3-demo-bucket --query Policy --output
 text > policy.json
```

그런 다음 필요에 따라 policy.json 파일을 수정할 수 있습니다. 마지막으로 다음을 실행하여 수정된 정책을 S3 버킷에 다시 적용할 수 있습니다.

policy.json 파일을 실행하여 수정된 정책을 S3 버킷에 다시 적용할 수 있습니다. 마지막으로 다음을 실행하여 수정된 정책을 S3 버킷에 다시 적용할 수 있습니다.

파일을 실행하여 수정된 정책을 S3 버킷에 다시 적용할 수 있습니다. 마지막으로 다음을 실행하여 수정된 정책을 S3 버킷에 다시 적용할 수 있습니다.

```
aws s3api put-bucket-policy --bucket amzn-s3-demo-bucket --policy file://policy.json
```

- API 세부 정보는 AWS CLI 명령 참조의 GetBucketPolicy를 참조하세요.

get-bucket-replication

다음 코드 예시에서는 get-bucket-replication의 사용 방법을 보여줍니다.

AWS CLI

다음 명령은 amzn-s3-demo-bucket이라는 버킷의 복제 구성을 가져옵니다.

```
aws s3api get-bucket-replication --bucket amzn-s3-demo-bucket
```

출력:

```
{
    "ReplicationConfiguration": {
        "Rules": [
            {
                "Status": "Enabled",
                "Prefix": "",
                "Destination": {
                    "Bucket": "arn:aws:s3:::amzn-s3-demo-bucket-backup",
                    "StorageClass": "STANDARD"
                },
                "ID": "ZmUwNzE4ZmQ4tMjVhOS00MTlkLOGI4NDkzZTIWJjNTUtYTA1"
            }
        ],
        "Role": "arn:aws:iam::123456789012:role/s3-replication-role"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetBucketReplication을 참조하세요.

get-bucket-request-payment

다음 코드 예시에서는 get-bucket-request-payment의 사용 방법을 보여줍니다.

AWS CLI

버킷의 구매 요청 구성 가져오기

다음 get-bucket-request-payment 예시에서는 지정된 버킷의 requester pays 구성을 가져옵니다.

```
aws s3api get-bucket-request-payment \
    --bucket amzn-s3-demo-bucket
```

출력:

```
{
    "Payer": "BucketOwner"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetBucketRequestPayment를 참조하세요.

get-bucket-tagging

다음 코드 예시에서는 get-bucket-tagging의 사용 방법을 보여줍니다.

AWS CLI

다음 명령은 amzn-s3-demo-bucket이라는 버킷의 태그 지정 구성을 가져옵니다.

```
aws s3api get-bucket-tagging --bucket amzn-s3-demo-bucket
```

출력:

```
{
    "TagSet": [
        {
            "Value": "marketing",
```

```
            "Key": "organization"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetBucketTagging을 참조하세요.

get-bucket-versioning

다음 코드 예시에서는 get-bucket-versioning의 사용 방법을 보여줍니다.

AWS CLI

다음 명령은 amzn-s3-demo-bucket이라는 버킷의 버전 관리 구성을 가져옵니다.

```
aws s3api get-bucket-versioning --bucket amzn-s3-demo-bucket
```

출력:

```
{
    "Status": "Enabled"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetBucketVersioning을 참조하세요.

get-bucket-website

다음 코드 예시에서는 get-bucket-website의 사용 방법을 보여줍니다.

AWS CLI

다음 명령은 amzn-s3-demo-bucket이라는 버킷의 정적 웹 사이트 구성을 가져옵니다.

```
aws s3api get-bucket-website --bucket amzn-s3-demo-bucket
```

출력:

```
{
    "IndexDocument": {
```

```
        "Suffix": "index.html"
    },
    "ErrorDocument": {
        "Key": "error.html"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetBucketWebsite를 참조하세요.

get-object-acl

다음 코드 예시에서는 `get-object-acl`의 사용 방법을 보여줍니다.

AWS CLI

다음 명령은 `amzn-s3-demo-bucket`이라는 버킷의 객체에 대한 액세스 제어 목록을 가져옵니다.

```
aws s3api get-object-acl --bucket amzn-s3-demo-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"
        }
    ]
```

```
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetObjectAcl을 참조하세요.

get-object-attributes

다음 코드 예시에서는 get-object-attributes의 사용 방법을 보여줍니다.

AWS CLI

객체 자체를 반환하지 않고 객체에서 메타데이터 가져오기

다음 get-object-attributes 예시에서는 객체 doc1.rtf에서 메타데이터를 가져옵니다.

```
aws s3api get-object-attributes \
    --bucket amzn-s3-demo-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 참조의 GetObjectAttributes를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetObjectAttributes를 참조하세요.

get-object-legal-hold

다음 코드 예시에서는 get-object-legal-hold의 사용 방법을 보여줍니다.

AWS CLI

객체의 법적 보류 상태 가져오기

다음 get-object-legal-hold 예시에서는 지정된 객체의 법적 보류 상태를 가져옵니다.

```
aws s3api get-object-legal-hold \
    --bucket amzn-s3-demo-bucket-with-object-lock \
    --key doc1.rtf
```

출력:

```
{
    "LegalHold": {
        "Status": "ON"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetObjectLegalHold를 참조하세요.

get-object-lock-configuration

다음 코드 예시에서는 get-object-lock-configuration의 사용 방법을 보여줍니다.

AWS CLI

버킷의 객체 잠금 구성 가져오기

다음 get-object-lock-configuration 예시에서는 지정된 버킷의 객체 잠금 구성을 가져옵니다.

```
aws s3api get-object-lock-configuration \
    --bucket amzn-s3-demo-bucket-with-object-lock
```

출력:

```
{
    "ObjectLockConfiguration": {
        "ObjectLockEnabled": "Enabled",
        "Rule": {
            "DefaultRetention": {
                "Mode": "COMPLIANCE",
                "Days": 50
            }
        }
    }
```

```
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetObjectLockConfiguration을 참조하세요.

get-object-retention

다음 코드 예시에서는 get-object-retention의 사용 방법을 보여줍니다.

AWS CLI

객체에 대한 객체 보존 구성 가져오기

다음 get-object-retention 예시에서는 지정된 객체에 대한 객체 보존 구성을 가져옵니다.

```
aws s3api get-object-retention \
    --bucket amzn-s3-demo-bucket-with-object-lock \
    --key doc1.rtf
```

출력:

```
{
    "Retention": {
        "Mode": "GOVERNANCE",
        "RetainUntilDate": "2025-01-01T00:00:00.000Z"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetObjectRetention을 참조하세요.

get-object-tagging

다음 코드 예시에서는 get-object-tagging의 사용 방법을 보여줍니다.

AWS CLI

객체에 연결된 태그 가져오기

다음 get-object-tagging 예시에서는 지정된 객체에서 지정된 키의 값을 가져옵니다.

```
aws s3api get-object-tagging \
```

```
    --bucket amzn-s3-demo-bucket \
    --key doc1.rtf
```

출력:

```
{
    "TagSet": [
        {
            "Value": "confidential",
            "Key": "designation"
        }
    ]
}
```

다음 `get-object-tagging` 예시에서는 태그가 없는 객체 `doc2.rtf`의 태그 세트를 가져오려고 시도합니다.

```
aws s3api get-object-tagging \
    --bucket amzn-s3-demo-bucket \
    --key doc2.rtf
```

출력:

```
{
    "TagSet": []
}
```

다음 `get-object-tagging` 예시에서는 태그가 여러 개 있는 객체 `doc3.rtf`의 태그 세트를 가져옵니다.

```
aws s3api get-object-tagging \
    --bucket amzn-s3-demo-bucket \
    --key doc3.rtf
```

출력:

```
{
    "TagSet": [
        {
            "Value": "confidential",
```

```
                "Key": "designation"
            },
            {
                "Value": "finance",
                "Key": "department"
            },
            {
                "Value": "payroll",
                "Key": "team"
            }
        ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetObjectTagging을 참조하세요.

get-object-torrent

다음 코드 예시에서는 get-object-torrent의 사용 방법을 보여줍니다.

AWS CLI

다음 명령은 amzn-s3-demo-bucket 버킷의 객체에 대한 메타데이터를 검색합니다.

```
aws s3api get-object-torrent --bucket amzn-s3-demo-bucket --key large-video-file.mp4 large-video-file.torrent
```

torrent 파일은 현재 폴더에 로컬로 저장됩니다. 출력 파일 이름(large-video-file.torrent)은 옵션 이름 없이 지정되며 명령의 마지막 인수여야 합니다.

- API 세부 정보는 AWS CLI 명령 참조의 GetObjectTorrent 섹션을 참조하세요.

get-object

다음 코드 예시에서는 get-object의 사용 방법을 보여줍니다.

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
```

객체를 가져오는 방법에 대한 자세한 내용은 Amazon S3 개발자 안내서의 객체 가져오기를 참조하세요.

- API 세부 정보는 AWS CLI명령 참조의 GetObject를 참조하세요.

get-public-access-block

다음 코드 예시에서는 get-public-access-block의 사용 방법을 보여줍니다.

AWS CLI

버킷의 퍼블릭 액세스 차단 구성을 설정하거나 수정

다음 get-public-access-block 예시에서는 지정된 버킷의 퍼블릭 액세스 차단 구성을 표시합니다.

```
aws s3api get-public-access-block \
    --bucket amzn-s3-demo-bucket
```

출력:

```
{
    "PublicAccessBlockConfiguration": {
        "IgnorePublicAcls": true,
        "BlockPublicPolicy": true,
        "BlockPublicAcls": true,
        "RestrictPublicBuckets": true
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetPublicAccessBlock을 참조하세요.

head-bucket

다음 코드 예시에서는 head-bucket의 사용 방법을 보여줍니다.

AWS CLI

다음 명령은 amzn-s3-demo-bucket이라는 버킷에 대한 액세스를 확인합니다.

```
aws s3api head-bucket --bucket amzn-s3-demo-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
```

- API 세부 정보는 AWS CLI 명령 참조의 HeadBucket을 참조하세요.

head-object

다음 코드 예시에서는 head-object의 사용 방법을 보여줍니다.

AWS CLI

다음 명령은 amzn-s3-demo-bucket이라는 버킷의 객체에 대한 메타데이터를 가져옵니다.

```
aws s3api head-object --bucket amzn-s3-demo-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": {}
}
```

- API 세부 정보는 AWS CLI 명령 참조의 HeadObject를 참조하세요.

`list-bucket-analytics-configurations`

다음 코드 예시에서는 `list-bucket-analytics-configurations`의 사용 방법을 보여줍니다.

AWS CLI

버킷의 분석 구성 목록 가져오기

다음 `list-bucket-analytics-configurations`는 지정된 버킷의 분석 구성 목록을 가져옵니다.

```
aws s3api list-bucket-analytics-configurations \
    --bucket amzn-s3-demo-bucket
```

출력:

```
{
    "AnalyticsConfigurationList": [
        {
            "StorageClassAnalysis": {},
            "Id": "1"
        }
    ],
    "IsTruncated": false
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListBucketAnalyticsConfigurations를 참조하세요.

`list-bucket-intelligent-tiering-configurations`

다음 코드 예시에서는 `list-bucket-intelligent-tiering-configurations`의 사용 방법을 보여줍니다.

AWS CLI

버킷에서 모든 S3 Intelligent-Tiering 구성을 검색하는 방법

다음 `list-bucket-intelligent-tiering-configurations` 예시에서는 버킷의 모든 S3 Intelligent-Tiering 구성을 검색합니다.

```
aws s3api list-bucket-intelligent-tiering-configurations \
    --bucket amzn-s3-demo-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 사용 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListBucketIntelligentTieringConfigurations 섹션을 참조하세요.

`list-bucket-inventory-configurations`

다음 코드 예시에서는 `list-bucket-inventory-configurations`의 사용 방법을 보여줍니다.

AWS CLI

버킷의 인벤토리 구성 가져오기

다음 `list-bucket-inventory-configurations` 예시에서는 지정된 버킷의 인벤토리 구성을 나열합니다.

```
aws s3api list-bucket-inventory-configurations \
    --bucket amzn-s3-demo-bucket
```

출력:

```
{
    "InventoryConfigurationList": [
        {
            "IsEnabled": true,
            "Destination": {
                "S3BucketDestination": {
                    "Format": "ORC",
                    "Bucket": "arn:aws:s3:::amzn-s3-demo-bucket",
                    "AccountId": "123456789012"
```

```
                }
            },
            "IncludedObjectVersions": "Current",
            "Id": "1",
            "Schedule": {
                "Frequency": "Weekly"
            }
        },
        {
            "IsEnabled": true,
            "Destination": {
                "S3BucketDestination": {
                    "Format": "CSV",
                    "Bucket": "arn:aws:s3:::amzn-s3-demo-bucket",
                    "AccountId": "123456789012"
                }
            },
            "IncludedObjectVersions": "Current",
            "Id": "2",
            "Schedule": {
                "Frequency": "Daily"
            }
        }
    ],
    "IsTruncated": false
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListBucketInventoryConfigurations를 참조하세요.

list-bucket-metrics-configurations

다음 코드 예시에서는 `list-bucket-metrics-configurations`의 사용 방법을 보여줍니다.

AWS CLI

버킷의 지표 구성 목록 검색

다음 `list-bucket-metrics-configurations` 예시에서는 지정된 버킷에 대한 지표 구성 목록을 검색합니다.

```
aws s3api list-bucket-metrics-configurations \
    --bucket amzn-s3-demo-bucket
```

출력:

```
{
    "IsTruncated": false,
    "MetricsConfigurationList": [
        {
            "Filter": {
                "Prefix": "logs"
            },
            "Id": "123"
        },
        {
            "Filter": {
                "Prefix": "tmp"
            },
            "Id": "234"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [ListBucketInventoryConfigurations](#) 섹션을 참조하세요.

list-buckets

다음 코드 예시에서는 `list-buckets`의 사용 방법을 보여줍니다.

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.

- API 세부 정보는 AWS CLI 명령 참조의 [ListBuckets](#)를 참조하세요.

list-multipart-uploads

다음 코드 예시에서는 `list-multipart-uploads`의 사용 방법을 보여줍니다.

AWS CLI

다음 명령은 `amzn-s3-demo-bucket`이라는 버킷의 활성 멀티파트 업로드를 모두 나열합니다.

```
aws s3api list-multipart-uploads --bucket amzn-s3-demo-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.

- API 세부 정보는 AWS CLI 명령 참조의 ListMultipartUploads를 참조하세요.

`list-object-versions`

다음 코드 예시에서는 `list-object-versions`의 사용 방법을 보여줍니다.

AWS CLI

다음 명령은 `amzn-s3-demo-bucket`이라는 버킷에서 객체의 버전 정보를 가져옵니다.

```
aws s3api list-object-versions --bucket amzn-s3-demo-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
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListObjectVersions를 참조하세요.

`list-objects-v2`

다음 코드 예시에서는 `list-objects-v2`의 사용 방법을 보여줍니다.

AWS CLI

버킷의 객체 목록 가져오기

다음 `list-objects-v2` 예시에서는 지정된 버킷의 객체를 나열합니다.

```
aws s3api list-objects-v2 \
    --bucket amzn-s3-demo-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 세부 정보는 AWS CLI 명령 참조의 ListObjectsV2를 참조하세요.

list-objects

다음 코드 예시에서는 `list-objects`의 사용 방법을 보여줍니다.

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.

- API 세부 정보는 AWS CLI 명령 참조의 ListObjects를 참조하세요.

list-parts

다음 코드 예시에서는 `list-parts`의 사용 방법을 보여줍니다.

AWS CLI

다음 명령은 `amzn-s3-demo-bucket` 버킷에 `multipart/01` 키를 사용하여 멀티파트 업로드를 위해 업로드된 모든 부분을 나열합니다.

```
aws s3api list-parts --bucket amzn-s3-demo-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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListParts 섹션을 참조하세요.

ls

다음 코드 예시에서는 `ls`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 사용자 소유 버킷 모두 나열

다음 `ls` 명령은 사용자가 소유한 모든 버킷을 나열합니다. 이 예시에서는 사용자가 `amzn-s3-demo-bucket` 및 `amzn-s3-demo-bucket2` 버킷을 소유합니다. 타임스탬프는 버킷이 생성된 날짜로, 기계의 시간대에 표시됩니다. 버킷 정책 편집과 같이 버킷을 변경할 때 이 날짜가 변경될 수 있습니다. `s3://`가 경로 인수 `<S3Uri>`에 사용되는 경우 모든 버킷도 나열됩니다.

```
aws s3 ls
```

출력:

```
2013-07-11 17:08:50 amzn-s3-demo-bucket
2013-07-24 14:55:44 amzn-s3-demo-bucket2
```

예시 2: 버킷의 모든 접두사 및 객체 나열

다음 `ls` 명령은 지정된 버킷 및 접두사 아래에 객체와 공통 접두사를 나열합니다. 이 예시에서는 사용자가 `test.txt` 및 `somePrefix/test.txt` 객체를 사용하여 `amzn-s3-demo-bucket` 버킷을 소유합니다. `LastWriteTime` 및 `Length`는 임의입니다. `ls` 명령은 로컬 파일 시스템과 상호 작용하지 않으므로 `s3://` URI 체계는 모호성을 해결하는 데 필요하지 않으며 생략될 수 있습니다.

```
aws s3 ls s3://amzn-s3-demo-bucket
```

출력:

```
                           PRE somePrefix/
2013-07-25 17:06:27         88 test.txt
```

예시 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://amzn-s3-demo-bucket/noExistPrefix
```

출력:

```
None
```

예시 4: 버킷의 모든 접두사 및 객체를 반복적으로 나열

다음 `ls` 명령은 버킷의 객체를 반복적으로 나열합니다. 출력에 `PRE dirname/`을 표시하는 대신 버킷의 모든 콘텐츠가 순서대로 나열됩니다.

```
aws s3 ls s3://amzn-s3-demo-bucket \
    --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
```

예시 5: 버킷의 모든 접두사 및 객체 요약

다음 `ls` 명령은 --human-readable 및 --summarize 옵션을 사용하여 동일한 명령을 보여줍니다. --human-readable은 파일 크기를 Bytes/MiB/KiB/GiB/TiB/PiB/EiB 단위로 표시합니다. --summarize는 결과 목록 끝에 총 객체 수와 총 크기를 표시합니다.

```
aws s3 ls s3://amzn-s3-demo-bucket \
    --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
```

예시 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 세부 정보는 AWS CLI 명령 참조의 Ls 섹션을 참조하세요.

mb

다음 코드 예시에서는 mb의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 버킷 생성

다음 mb 명령은 버킷을 생성합니다. 이 예시에서는 사용자가 버킷 amzn-s3-demo-bucket을 생성합니다. 버킷은 사용자의 구성 파일에 지정된 리전에 생성됩니다.

```
aws s3 mb s3://amzn-s3-demo-bucket
```

출력:

```
make_bucket: s3://amzn-s3-demo-bucket
```

예시 2: 지정된 리전에서 버킷 생성

다음 mb 명령은 --region 파라미터에 의해 지정된 리전에 버킷을 생성합니다. 이 예시에서는 사용자가 us-west-1 리전에서 amzn-s3-demo-bucket 버킷을 만듭니다.

```
aws s3 mb s3://amzn-s3-demo-bucket \
    --region us-west-1
```

출력:

```
make_bucket: s3://amzn-s3-demo-bucket
```

- API 세부 정보는 AWS CLI 명령 참조의 Mb 섹션을 참조하세요.

mv

다음 코드 예시에서는 mv의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 로컬 파일을 지정된 버킷으로 이동

다음 mv 명령은 단일 파일을 지정된 버킷 및 키로 이동합니다.

```
aws s3 mv test.txt s3://amzn-s3-demo-bucket/test2.txt
```

출력:

```
move: test.txt to s3://amzn-s3-demo-bucket/test2.txt
```

예시 2: 객체를 지정된 버킷 및 키로 이동

다음 mv 명령은 단일 s3 객체를 지정된 버킷 및 키로 이동합니다.

```
aws s3 mv s3://amzn-s3-demo-bucket/test.txt s3://amzn-s3-demo-bucket/test2.txt
```

출력:

```
move: s3://amzn-s3-demo-bucket/test.txt to s3://amzn-s3-demo-bucket/test2.txt
```

예시 3: S3 객체를 로컬 디렉터리로 이동

다음 mv 명령은 단일 객체를 로컬로 지정된 파일로 이동합니다.

```
aws s3 mv s3://amzn-s3-demo-bucket/test.txt test2.txt
```

출력:

```
move: s3://amzn-s3-demo-bucket/test.txt to test2.txt
```

예시 4: 원래 이름을 가진 객체를 지정된 버킷으로 이동

다음 mv 명령은 원래 이름을 유지하면서 단일 객체를 지정된 버킷으로 이동합니다.

```
aws s3 mv s3://amzn-s3-demo-bucket/test.txt s3://amzn-s3-demo-bucket2/
```

출력:

```
move: s3://amzn-s3-demo-bucket/test.txt to s3://amzn-s3-demo-bucket2/test.txt
```

예시 5: 버킷의 모든 객체 및 접두사를 로컬 디렉터리로 이동

`--recursive` 파라미터와 함께 전달되면 다음 mv 명령은 지정된 접두사 및 버킷에 있는 모든 객체를 지정된 디렉터리에 반복적으로 이전합니다. 이 예시에서는 `amzn-s3-demo-bucket` 버킷에 `test1.txt` 및 `test2.txt` 객체가 있습니다.

```
aws s3 mv s3://amzn-s3-demo-bucket . \
    --recursive
```

출력:

```
move: s3://amzn-s3-demo-bucket/test1.txt to test1.txt
move: s3://amzn-s3-demo-bucket/test2.txt to test2.txt
```

예시 6: ``.jpg`` 파일을 제외한 버킷의 모든 객체 및 접두사를 로컬 디렉터리로 이동

`--recursive` 파라미터와 함께 전달되면 다음 mv 명령은 `--exclude` 파라미터를 사용하여 일부 파일을 제외하면서 지정된 디렉터리에 있는 모든 파일을 지정된 버킷 및 접두사에 반복적으로 이전합니다. 이 예시에서는 `myDir` 디렉터리에 `test1.txt` 및 `test2.jpg` 파일이 있습니다.

```
aws s3 mv myDir s3://amzn-s3-demo-bucket/ \
    --recursive \
    --exclude "*.jpg"
```

출력:

```
move: myDir/test1.txt to s3://amzn-s3-demo-bucket2/test1.txt
```

예시 7: 지정된 접두사를 제외한 버킷의 모든 객체 및 접두사를 로컬 디렉터리로 이동

`--recursive` 파라미터와 함께 전달되면 다음 mv 명령은 `--exclude` 파라미터를 사용하여 일부 객체를 제외하면서 지정된 버킷에 있는 모든 객체를 다른 버킷에 반복적으로 이전합니다. 이 예시에서는 `amzn-s3-demo-bucket` 버킷에 `test1.txt` 및 `another/test1.txt` 객체가 있습니다.

```
aws s3 mv s3://amzn-s3-demo-bucket/ s3://amzn-s3-demo-bucket2/ \
```

```
    --recursive \
    --exclude "amzn-s3-demo-bucket/another/*"
```

출력:

```
move: s3://amzn-s3-demo-bucket/test1.txt to s3://amzn-s3-demo-bucket2/test1.txt
```

예시 8: 객체를 지정된 버킷으로 이동하고 ACL 설정

다음 mv 명령은 ACL을 `public-read-write`로 설정하는 동안 단일 객체를 지정된 버킷 및 키에 이전합니다.

```
aws s3 mv s3://amzn-s3-demo-bucket/test.txt s3://amzn-s3-demo-bucket/test2.txt \
    --acl public-read-write
```

출력:

```
move: s3://amzn-s3-demo-bucket/test.txt to s3://amzn-s3-demo-bucket/test2.txt
```

예시 9: 로컬 파일을 지정된 버킷으로 이동하고 권한 부여

다음 mv 명령은 모든 사용자에게 읽기 권한을 부여하고 이메일 주소로 식별되는 특정 사용자에게 모든 권한을 부여하는 `--grants` 옵션을 사용하는 방법을 보여줍니다.

```
aws s3 mv file.txt s3://amzn-s3-demo-bucket/ \
    --grants read=uri=http://acs.amazonaws.com/groups/global/
AllUsers full=emailaddress=user@example.com
```

출력:

```
move: file.txt to s3://amzn-s3-demo-bucket/file.txt
```

예시 10: 파일을 S3 액세스 포인트로 이동

다음 mv 명령은 단일 `mydoc.txt` 파일을 `mykey` 키의 `myaccesspoint` 액세스 포인트로 이동합니다.

```
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 세부 정보는 AWS CLI 명령 참조의 Mv 섹션을 참조하세요.

presign

다음 코드 예시에서는 presign의 사용 방법을 보여줍니다.

AWS CLI

예시 1: S3 버킷의 객체에 연결하는 기본 1시간 수명으로 미리 서명된 URL 생성

다음 presign 명령은 1시간 동안 유효한 지정된 버킷 및 키에 대해 미리 서명된 URL을 생성합니다.

```
aws s3 presign s3://amzn-s3-demo-bucket/test2.txt
```

출력:

```
https://amzn-s3-demo-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
```

예시 2: S3 버킷의 객체에 연결하는 사용자 지정 수명 주기로 미리 서명된 URL 생성

다음 presign 명령은 1주일 동안 유효한 지정된 버킷 및 키에 대해 미리 서명된 URL을 생성합니다.

```
aws s3 presign s3://amzn-s3-demo-bucket/test2.txt \
    --expires-in 604800
```

출력:

```
https://amzn-s3-demo-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 공유 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 Presign 섹션을 참조하세요.

put-bucket-accelerate-configuration

다음 코드 예시에서는 put-bucket-accelerate-configuration의 사용 방법을 보여줍니다.

AWS CLI

버킷의 가속화 구성 설정

다음 put-bucket-accelerate-configuration 예시에서는 지정된 버킷의 가속화 구성을 활성화합니다.

```
aws s3api put-bucket-accelerate-configuration \
    --bucket amzn-s3-demo-bucket \
    --accelerate-configuration Status=Enabled
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 PutBucketAccelerateConfiguration을 참조하세요.

put-bucket-acl

다음 코드 예시에서는 put-bucket-acl의 사용 방법을 보여줍니다.

AWS CLI

이 예시에서는 두 명의 AWS 사용자(user1@example.com 및 user2@example.com)에게 full control 권한을 부여하고 모든 사용자에게 read 권한을 부여합니다.

```
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
```

사용자 지정 ACL에 대한 자세한 내용은 http://docs.aws.amazon.com/AmazonS3/latest/API/RESTBucketPUTacl.html을 참조하세요(put-bucket-acl과 같은 s3api ACL 명령은 동일한 간편 인수 표기법을 사용함).

- API 세부 정보는 AWS CLI 명령 참조의 PutBucketAcl을 참조하세요.

put-bucket-analytics-configuration

다음 코드 예시에서는 put-bucket-analytics-configuration의 사용 방법을 보여줍니다.

AWS CLI

버킷의 분석 구성 설정

다음 put-bucket-analytics-configuration 예시는 지정된 버킷에 대한 분석을 구성합니다.

```
aws s3api put-bucket-analytics-configuration \
    --bucket amzn-s3-demo-bucket --id 1 \
    --analytics-configuration '{"Id": "1","StorageClassAnalysis": {}}'
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 PutBucketAnalyticsConfiguration 섹션을 참조하세요.

put-bucket-cors

다음 코드 예시에서는 put-bucket-cors의 사용 방법을 보여줍니다.

AWS CLI

다음 예시에서는 www.example.com의 PUT, POST 및 DELETE 요청을 활성화하고 모든 도메인의 GET 요청을 활성화합니다.

```
aws s3api put-bucket-cors --bucket amzn-s3-demo-bucket --cors-configuration file://
cors.json

cors.json:
{
  "CORSRules": [
    {
      "AllowedOrigins": ["http://www.example.com"],
      "AllowedHeaders": ["*"],
      "AllowedMethods": ["PUT", "POST", "DELETE"],
      "MaxAgeSeconds": 3000,
```

```
        "ExposeHeaders": ["x-amz-server-side-encryption"]
      },
      {
        "AllowedOrigins": ["*"],
        "AllowedHeaders": ["Authorization"],
        "AllowedMethods": ["GET"],
        "MaxAgeSeconds": 3000
      }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 PutBucketCors를 참조하세요.

put-bucket-encryption

다음 코드 예시에서는 `put-bucket-encryption`의 사용 방법을 보여줍니다.

AWS CLI

버킷의 서버 측 암호화 구성

다음 `put-bucket-encryption` 예시에서는 AES256 암호화를 지정된 버킷의 기본값으로 설정합니다.

```
aws s3api put-bucket-encryption \
    --bucket amzn-s3-demo-bucket \
    --server-side-encryption-configuration '{"Rules":
 [{"ApplyServerSideEncryptionByDefault": {"SSEAlgorithm": "AES256"}}]}'
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 PutBucketEncryption을 참조하세요.

put-bucket-intelligent-tiering-configuration

다음 코드 예시에서는 `put-bucket-intelligent-tiering-configuration`의 사용 방법을 보여줍니다.

AWS CLI

버킷에서 S3 Intelligent-Tiering 구성을 업데이트하는 방법

다음 put-bucket-intelligent-tiering-configuration 예시에서는 버킷에서 ExampleConfig라는 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 amzn-s3-demo-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 버킷에 대한 객체 소유권 설정 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutBucketAccelerateConfiguration 섹션을 참조하세요.

put-bucket-inventory-configuration

다음 코드 예시에서는 put-bucket-inventory-configuration의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 버킷에 대한 인벤토리 구성 설정

다음 put-bucket-inventory-configuration 예시에서는 amzn-s3-demo-bucket 버킷 에 대한 주간 ORC 형식 인벤토리 보고서를 설정합니다.

```
aws s3api put-bucket-inventory-configuration \
    --bucket amzn-s3-demo-bucket \
    --id 1 \
    --inventory-configuration '{"Destination": { "S3BucketDestination":
 { "AccountId": "123456789012", "Bucket": "arn:aws:s3:::amzn-s3-demo-bucket",
 "Format": "ORC" }}, "IsEnabled": true, "Id": "1", "IncludedObjectVersions":
 "Current", "Schedule": { "Frequency": "Weekly" }}'
```

이 명령은 출력을 생성하지 않습니다.

예시 2: 버킷에 대한 인벤토리 구성 설정

다음 put-bucket-inventory-configuration 예시에서는 amzn-s3-demo-bucket 버킷 에 대한 일일 CSV 형식 인벤토리 보고서를 설정합니다.

```
aws s3api put-bucket-inventory-configuration \
    --bucket amzn-s3-demo-bucket \
    --id 2 \
    --inventory-configuration '{"Destination": { "S3BucketDestination":
 { "AccountId": "123456789012", "Bucket": "arn:aws:s3:::amzn-s3-demo-bucket",
 "Format": "CSV" }}, "IsEnabled": true, "Id": "2", "IncludedObjectVersions":
 "Current", "Schedule": { "Frequency": "Daily" }}'
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 PutBucketInventoryConfiguration 섹션을 참조하세요.

put-bucket-lifecycle-configuration

다음 코드 예시에서는 put-bucket-lifecycle-configuration의 사용 방법을 보여줍니다.

AWS CLI

다음 명령은 amzn-s3-demo-bucket이라는 버킷에 수명 주기 구성을 적용합니다.

```
aws s3api put-bucket-lifecycle-configuration --bucket amzn-s3-demo-bucket --
lifecycle-configuration  file://lifecycle.json
```

lifecycle.json 파일은 다음 두 규칙을 지정하는 현재 폴더의 JSON 문서입니다.

```
{
    "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 사용자 안내서의 파라미터 값 지정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutBucketLifecycleConfiguration을 참조하세요.

put-bucket-lifecycle

다음 코드 예시에서는 put-bucket-lifecycle의 사용 방법을 보여줍니다.

AWS CLI

다음 명령은 amzn-s3-demo-bucket 버킷에 수명 주기 구성을 적용합니다.

```
aws s3api put-bucket-lifecycle --bucket amzn-s3-demo-bucket --lifecycle-configuration file://lifecycle.json
```

`lifecycle.json` 파일은 다음 두 규칙을 지정하는 현재 폴더의 JSON 문서입니다.

```
{
  "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 사용자 안내서의 파라미터 값 지정을 참조하세요.

위 예시의 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"
          }
        }
      ]
    }
```

- API 세부 정보는 AWS CLI 명령 참조의 PutBucketLifecycle 섹션을 참조하세요.

put-bucket-logging

다음 코드 예시에서는 put-bucket-logging의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 버킷 정책 로깅 설정

다음 put-bucket-logging 예제에서는 amzn-s3-demo-bucket에 대한 로깅 정책을 설정합니다. 먼저 put-bucket-policy 명령을 사용하여 버킷 정책에서 로깅 서비스 보안 주체 권한을 부여합니다.

```
aws s3api put-bucket-policy \
    --bucket amzn-s3-demo-bucket \
    --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:::amzn-s3-demo-bucket/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 amzn-s3-demo-bucket \
    --bucket-logging-status file://logging.json
```

logging.json의 콘텐츠:

```
{
     "LoggingEnabled": {
         "TargetBucket": "amzn-s3-demo-bucket",
         "TargetPrefix": "Logs/"
     }
 }
```

로깅 서비스 위탁자에 s3:PutObject 권한을 부여하려면 put-bucket-policy 명령이 필요합니다.

자세한 내용은 Amazon S3 사용자 안내서의 Amazon S3 서버 액세스 로깅을 참조하세요.

예시 2: 단일 사용자에게만 액세스 로깅에 대한 버킷 정책 설정

다음 put-bucket-logging 예제에서는 amzn-s3-demo-bucket에 대한 로깅 정책을 설정합니다. AWS 사용자 bob@example.com은 로그 파일을 완전히 제어할 수 있으며 다른 사용자는 액세스 권한을 갖지 않습니다. 먼저 put-bucket-acl을 사용하여 S3 권한을 부여합니다.

```
aws s3api put-bucket-acl \
    --bucket amzn-s3-demo-bucket \
    --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-logging을 사용하여 로깅 정책을 적용합니다.

```
aws s3api put-bucket-logging \
    --bucket amzn-s3-demo-bucket \
    --bucket-logging-status file://logging.json
```

logging.json의 콘텐츠:

```
{
    "LoggingEnabled": {
        "TargetBucket": "amzn-s3-demo-bucket",
        "TargetPrefix": "amzn-s3-demo-bucket-logs/",
        "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.

- API 세부 정보는 AWS CLI 명령 참조의 PutBucketLogging을 참조하세요.

put-bucket-metrics-configuration

다음 코드 예시에서는 put-bucket-metrics-configuration의 사용 방법을 보여줍니다.

AWS CLI

버킷의 지표 구성 설정

다음 put-bucket-metrics-configuration 예시에서는 지정된 버킷의 ID 123에 대한 지표 구성을 설정합니다.

```
aws s3api put-bucket-metrics-configuration \
    --bucket amzn-s3-demo-bucket \
    --id 123 \
    --metrics-configuration '{"Id": "123", "Filter": {"Prefix": "logs"}}'
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 PutBucketLifecycleConfiguration 섹션을 참조하세요.

put-bucket-notification-configuration

다음 코드 예시에서는 put-bucket-notification-configuration의 사용 방법을 보여줍니다.

AWS CLI

버킷에 지정된 알림 활성화

다음 put-bucket-notification-configuration 예시에서는 amzn-s3-demo-bucket이라는 버킷에 알림 구성을 적용합니다. notification.json 파일은 모니터링할 SNS 주제와 이벤트 유형을 지정하는 현재 폴더의 JSON 문서입니다.

```
aws s3api put-bucket-notification-configuration \
    --bucket amzn-s3-demo-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 주제에 IAM 정책이 연결되어 있어야 Amazon S3이 해당 주제에 게시할 수 있습니다.

```
{
    "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:*:*:amzn-s3-demo-bucket"
                    }
                }
            }
        ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 PutBucketNotificationConfiguration을 참조하세요.

put-bucket-notification

다음 코드 예시에서는 put-bucket-notification의 사용 방법을 보여줍니다.

AWS CLI

amzn-s3-demo-bucket이라는 버킷에 알림 구성을 적용합니다.

```
aws s3api put-bucket-notification --bucket amzn-s3-demo-bucket --notification-configuration file://notification.json
```

notification.json 파일은 모니터링할 SNS 주제와 이벤트 유형을 지정하는 현재 폴더의 JSON 문서입니다.

```
{
  "TopicConfiguration": {
    "Event": "s3:ObjectCreated:*",
    "Topic": "arn:aws:sns:us-west-2:123456789012:s3-notification-topic"
  }
}
```

SNS 주제에 IAM 정책이 연결되어 있어야 Amazon S3이 해당 주제에 게시할 수 있습니다.

```
{
 "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:amzn-s3-demo-bucket",
    "Condition": {
       "ArnLike": {
       "aws:SourceArn": "arn:aws:s3:*:*:amzn-s3-demo-bucket"
     }
    }
   }
  ]
 }
```

- API 세부 정보는 AWS CLI 명령 참조의 PutBucketNotification을 참조하세요.

put-bucket-ownership-controls

다음 코드 예시에서는 put-bucket-ownership-controls의 사용 방법을 보여줍니다.

AWS CLI

버킷의 버킷 소유권 설정을 업데이트하는 방법

다음 put-bucket-ownership-controls 예시에서는 버킷의 버킷 소유권 설정을 업데이트합니다.

```
aws s3api put-bucket-ownership-controls \
    --bucket amzn-s3-demo-bucket \
    --ownership-controls="Rules=[{ObjectOwnership=BucketOwnerEnforced}]"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon S3 사용 설명서의 기존 버킷에 대한 객체 소유권 설정 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutBucketOwnershipControls 섹션을 참조하세요.

put-bucket-policy

다음 코드 예시는 put-bucket-policy의 사용 방법을 보여 줍니다.

AWS CLI

이 예제에서는 모든 사용자가 MySecretFolder에 있는 객체를 제외하고 amzn-s3-demo-bucket에 있는 모든 객체를 가져올 수 있습니다. 또한 1234-5678-9012라는 AWS 계정의 루트 사용자에게 put 및 delete 권한을 부여합니다.

```
aws s3api put-bucket-policy --bucket amzn-s3-demo-bucket --policy file://policy.json

policy.json:
{
   "Statement": [
      {
         "Effect": "Allow",
         "Principal": "*",
         "Action": "s3:GetObject",
         "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
      },
      {
         "Effect": "Deny",
         "Principal": "*",
         "Action": "s3:GetObject",
         "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/MySecretFolder/*"
      },
      {
         "Effect": "Allow",
         "Principal": {
            "AWS": "arn:aws:iam::123456789012:root"
         },
         "Action": [
            "s3:DeleteObject",
            "s3:PutObject"
         ],
         "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
      }

   ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 PutBucketPolicy를 참조하세요.

put-bucket-replication

다음 코드 예시에서는 put-bucket-replication의 사용 방법을 보여줍니다.

AWS CLI

S3 버킷의 복제 구성

다음 put-bucket-replication 예시에서는 지정된 S3 버킷에 복제 구성을 적용합니다.

```
aws s3api put-bucket-replication \
    --bucket amzn-s3-demo-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:::amzn-s3-demo-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:::amzn-s3-demo-bucket1"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObjectVersion",
                "s3:GetObjectVersionAcl",
                "s3:GetObjectVersionTagging"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket1/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ReplicateObject",
                "s3:ReplicateDelete",
                "s3:ReplicateTags"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-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 사용자 안내서의 주제 제목을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutBucketReplication을 참조하세요.

put-bucket-request-payment

다음 코드 예시에서는 put-bucket-request-payment의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 버킷의 `requester pays` 구성 활성화

다음 put-bucket-request-payment 예시에서는 지정된 버킷의 requester pays를 활성화합니다.

```
aws s3api put-bucket-request-payment \
    --bucket amzn-s3-demo-bucket \
    --request-payment-configuration '{"Payer":"Requester"}'
```

이 명령은 출력을 생성하지 않습니다.

예시 2: 버킷의 `requester pays` 구성 비활성화

다음 put-bucket-request-payment 예시에서는 지정된 버킷의 requester pays를 비활성화합니다.

```
aws s3api put-bucket-request-payment \
    --bucket amzn-s3-demo-bucket \
    --request-payment-configuration '{"Payer":"BucketOwner"}'
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 PutBucketRequestPayment를 참조하세요.

put-bucket-tagging

다음 코드 예시에서는 put-bucket-tagging의 사용 방법을 보여줍니다.

AWS CLI

다음 명령은 amzn-s3-demo-bucket이라는 버킷에 태그 지정 구성을 적용합니다.

```
aws s3api put-bucket-tagging --bucket amzn-s3-demo-bucket --tagging file://
tagging.json
```

tagging.json 파일은 태그를 지정하는 현재 폴더의 JSON 문서입니다.

```
{
   "TagSet": [
     {
       "Key": "organization",
       "Value": "marketing"
     }
   ]
}
```

또는 명령줄에서 태그 지정 구성을 amzn-s3-demo-bucket에 직접 적용할 수도 있습니다.

```
aws s3api put-bucket-tagging --bucket amzn-s3-demo-bucket --tagging
 'TagSet=[{Key=organization,Value=marketing}]'
```

- API 세부 정보는 AWS CLI 명령 참조의 PutBucketTagging을 참조하세요.

put-bucket-versioning

다음 코드 예시에서는 put-bucket-versioning의 사용 방법을 보여줍니다.

AWS CLI

다음 명령은 amzn-s3-demo-bucket이라는 버킷의 버전 관리를 활성화합니다.

```
aws s3api put-bucket-versioning --bucket amzn-s3-demo-bucket --versioning-
configuration Status=Enabled
```

다음 명령은 버전 관리를 활성화하고 mfa 코드를 사용합니다.

```
aws s3api put-bucket-versioning --bucket amzn-s3-demo-bucket --versioning-
configuration Status=Enabled --mfa "SERIAL 123456"
```

- API 세부 정보는 AWS CLI 명령 참조의 PutBucketVersioning을 참조하세요.

put-bucket-website

다음 코드 예시에서는 put-bucket-website의 사용 방법을 보여줍니다.

AWS CLI

amzn-s3-demo-bucket이라는 버킷에 정적 웹 사이트 구성을 적용합니다.

```
aws s3api put-bucket-website --bucket amzn-s3-demo-bucket --website-
configuration file://website.json
```

website.json 파일은 웹 사이트의 색인 및 오류 페이지를 지정하는 현재 폴더의 JSON 문서입니다.

```
{
    "IndexDocument": {
        "Suffix": "index.html"
    },
    "ErrorDocument": {
        "Key": "error.html"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 PutBucketWebsite를 참조하세요.

put-object-acl

다음 코드 예시에서는 put-object-acl의 사용 방법을 보여줍니다.

AWS CLI

다음 명령은 두 명의 AWS 사용자(user1@example.com 및 user2@example.com)에게 full control 권한을 부여하고 모든 사용자에게 read 권한을 부여합니다.

```
aws s3api put-object-acl --bucket amzn-s3-demo-bucket --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 지정 ACL에 대한 자세한 내용은 http://docs.aws.amazon.com/AmazonS3/latest/API/RESTBucketPUTacl.html을 참조하세요(put-object-acl과 같은 s3api ACL 명령은 동일한 간편 인수 표기법을 사용함).

- API 세부 정보는 AWS CLI 명령 참조의 PutObjectAcl을 참조하세요.

put-object-legal-hold

다음 코드 예시에서는 put-object-legal-hold의 사용 방법을 보여줍니다.

AWS CLI

객체에 법적 보류 적용

다음 put-object-legal-hold 예시에서는 doc1.rtf 객체에 법적 보존을 설정합니다.

```
aws s3api put-object-legal-hold \
    --bucket amzn-s3-demo-bucket-with-object-lock \
    --key doc1.rtf \
    --legal-hold Status=ON
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 PutObjectLegalHold를 참조하세요.

put-object-lock-configuration

다음 코드 예시에서는 put-object-lock-configuration의 사용 방법을 보여줍니다.

AWS CLI

버킷에 객체 잠금 구성 설정

다음 put-object-lock-configuration 예시에서는 지정된 버킷에 50일 객체 잠금을 설정합니다.

```
aws s3api put-object-lock-configuration \
    --bucket amzn-s3-demo-bucket-with-object-lock \
    --object-lock-configuration '{ "ObjectLockEnabled": "Enabled", "Rule":
 { "DefaultRetention": { "Mode": "COMPLIANCE", "Days": 50 }}}'
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 PutObjectLockConfiguration을 참조하세요.

put-object-retention

다음 코드 예시에서는 put-object-retention의 사용 방법을 보여줍니다.

AWS CLI

객체에 대한 객체 보존 구성 설정

다음 put-object-retention 예시에서는 지정된 객체에 대한 객체 보존 구성을 2025년 1월 1일 전까지로 설정합니다.

```
aws s3api put-object-retention \
    --bucket amzn-s3-demo-bucket-with-object-lock \
    --key doc1.rtf \
    --retention '{ "Mode": "GOVERNANCE", "RetainUntilDate": "2025-01-01T00:00:00" }'
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 PutObjectRetention을 참조하세요.

put-object-tagging

다음 코드 예시에서는 put-object-tagging의 사용 방법을 보여줍니다.

AWS CLI

객체에 태그를 설정하는 방법

다음 put-object-tagging 예시에서는 키 이름 designation 및 값 confidential이 있는 태그를 지정된 객체에 추가합니다.

```
aws s3api put-object-tagging \
    --bucket amzn-s3-demo-bucket \
    --key doc1.rtf \
    --tagging '{"TagSet": [{ "Key": "designation", "Value": "confidential" }]}'
```

이 명령은 출력을 생성하지 않습니다.

다음 put-object-tagging 예시에서는 지정된 객체에 여러 태그 세트를 설정합니다.

```
aws s3api put-object-tagging \
```

```
    --bucket amzn-s3-demo-bucket-example \
    --key doc3.rtf \
    --tagging '{"TagSet": [{ "Key": "designation", "Value": "confidential" },
 { "Key": "department", "Value": "finance" }, { "Key": "team", "Value":
 "payroll" } ]}'
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 PutObjectTagging 섹션을 참조하세요.

put-object

다음 코드 예시는 put-object의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: Amazon S3에 객체 업로드

다음 put-object 예제에서는 Amazon S3에 객체를 업로드합니다.

```
aws s3api put-object \
    --bucket amzn-s3-demo-bucket \
    --key my-dir/MySampleImage.png \
    --body MySampleImage.png
```

객체 업로드에 대한 자세한 내용은 Amazon S3 개발자 안내서의 객체 업로드 < http://docs.aws.amazon.com/AmazonS3/latest/dev/UploadingObjects.html>를 참조하세요.

예제 2: Amazon S3에 비디오 파일 업로드

다음 put-object 예제 명령은 비디오 파일을 업로드합니다.

```
aws s3api put-object \
    --bucket amzn-s3-demo-bucket \
    --key my-dir/big-video-file.mp4 \
    --body /media/videos/f-sharp-3-data-services.mp4
```

객체 업로드에 대한 자세한 내용은 Amazon S3 개발자 안내서의 객체 업로드 < http://docs.aws.amazon.com/AmazonS3/latest/dev/UploadingObjects.html>를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutObject를 참조하세요.

put-public-access-block

다음 코드 예시에서는 put-public-access-block의 사용 방법을 보여줍니다.

AWS CLI

버킷의 퍼블릭 액세스 차단 구성 설정

다음 put-public-access-block 예시는 지정된 버킷에 대한 퍼블릭 액세스 차단 구성을 설정합니다.

```
aws s3api put-public-access-block \
    --bucket amzn-s3-demo-bucket \
    --public-access-block-
configuration "BlockPublicAcls=true,IgnorePublicAcls=true,BlockPublicPolicy=true,RestrictPub
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 PutPublicAccessBlock 섹션을 참조하세요.

rb

다음 코드 예시에서는 rb의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 버킷 삭제

다음 rb 명령은 버킷을 제거합니다. 이 예시에서 사용자의 버킷은 amzn-s3-demo-bucket입니다. 버킷을 제거하려면 비어 있어야 합니다.

```
aws s3 rb s3://amzn-s3-demo-bucket
```

출력:

```
remove_bucket: amzn-s3-demo-bucket
```

예시 2: 버킷 강제 삭제

다음 rb 명령은 --force 파라미터를 사용하여 먼저 버킷의 모든 객체를 제거한 다음 버킷 자체를 제거합니다. 이 예시에서 사용자의 버킷은 amzn-s3-demo-bucket이고 amzn-s3-demo-bucket의 객체는 test1.txt 및 test2.txt입니다.

```
aws s3 rb s3://amzn-s3-demo-bucket \
    --force
```

출력:

```
delete: s3://amzn-s3-demo-bucket/test1.txt
delete: s3://amzn-s3-demo-bucket/test2.txt
remove_bucket: amzn-s3-demo-bucket
```

- API 세부 정보는 AWS CLI 명령 참조의 Rb 섹션을 참조하세요.

`restore-object`

다음 코드 예시에서는 `restore-object`의 사용 방법을 보여줍니다.

AWS CLI

객체에 대한 복원 요청 생성

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.

- API 세부 정보는 AWS CLI 명령 참조의 RestoreObject를 참조하세요.

`rm`

다음 코드 예시에서는 `rm`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: S3 객체 삭제

다음 `rm` 명령은 단일 s3 객체를 삭제합니다.

```
aws s3 rm s3://amzn-s3-demo-bucket/test2.txt
```

출력:

```
delete: s3://amzn-s3-demo-bucket/test2.txt
```

예시 2: 버킷의 모든 콘텐츠 삭제

다음 rm 명령은 --recursive 파라미터와 함께 전달될 때 지정된 버킷 및 접두사 아래의 모든 객체를 반복적으로 삭제합니다. 이 예시에서는 amzn-s3-demo-bucket 버킷에 test1.txt 및 test2.txt 객체가 포함되어 있습니다.

```
aws s3 rm s3://amzn-s3-demo-bucket \
    --recursive
```

출력:

```
delete: s3://amzn-s3-demo-bucket/test1.txt
delete: s3://amzn-s3-demo-bucket/test2.txt
```

예시 3: ``.jpg`` 파일을 제외한 버킷의 모든 콘텐츠 삭제

다음 rm 명령은 --recursive 파라미터를 사용하여 일부 객체를 제외하면서 --exclude 파라미터와 함께 전달될 때 지정된 버킷 및 접두사 아래의 모든 객체를 반복적으로 삭제합니다. 이 예시에서는 amzn-s3-demo-bucket 버킷에 test1.txt 및 test2.jpg 객체가 있습니다.

```
aws s3 rm s3://amzn-s3-demo-bucket/ \
    --recursive \
    --exclude "*.jpg"
```

출력:

```
delete: s3://amzn-s3-demo-bucket/test1.txt
```

예시 4: 지정된 접두사 아래의 객체를 제외한 버킷의 모든 콘텐츠 삭제

다음 rm 명령은 --recursive 파라미터를 사용하여 특정 접두사 아래의 모든 객체를 제외하면서 --exclude 파라미터와 함께 전달될 때 지정된 버킷 및 접두사 아래의 모든 객체를 반복적으로 삭

제합니다. 이 예시에서는 amzn-s3-demo-bucket 버킷에 test1.txt 및 another/test.txt 객체가 있습니다.

```
aws s3 rm s3://amzn-s3-demo-bucket/ \
    --recursive \
    --exclude "another/*"
```

출력:

```
delete: s3://amzn-s3-demo-bucket/test1.txt
```

예시 5: S3 액세스 포인트에서 객체 삭제

다음 rm 명령은 액세스 포인트(myaccesspoint)에서 단일 객체(mykey)를 삭제합니다. :: 다음 rm 명령은 액세스 포인트(myaccesspoint)에서 단일 객체(mykey)를 삭제합니다.

```
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 세부 정보는 AWS CLI 명령 참조의 Rm 섹션을 참조하세요.

select-object-content

다음 코드 예시에서는 select-object-content의 사용 방법을 보여줍니다.

AWS CLI

SQL 문을 기반으로 Amazon S3 객체의 콘텐츠 필터링

다음 select-object-content 예시에서는 지정된 SQL 문으로 객체 my-data-file.csv를 필터링하고 출력을 파일로 보냅니다.

```
aws s3api select-object-content \
    --bucket amzn-s3-demo-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.

- API 세부 정보는 AWS CLI 명령 참조의 SelectObjectContent를 참조하세요.

sync

다음 코드 예시에서는 sync의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 모든 로컬 객체를 지정된 버킷에 동기화

다음 sync 명령은 로컬 파일을 S3에 업로드하여 로컬 디렉터리의 객체를 지정된 접두사 및 버킷으로 동기화합니다. 로컬 파일의 크기가 S3 객체의 크기와 다르거나 로컬 파일의 마지막 수정 시간이 S3 객체의 마지막 수정 시간보다 최신이거나 로컬 파일이 지정된 버킷 및 접두사에 존재하지 않는 경우 로컬 파일을 업로드해야 합니다. 이 예시에서는 사용자가 amzn-s3-demo-bucket 버킷을 로컬 현재 디렉터리에 동기화합니다. 현재 로컬 디렉터리에는 test.txt 및 test2.txt 파일이 들어 있습니다. amzn-s3-demo-bucket 버킷에 객체가 없습니다.

```
aws s3 sync . s3://amzn-s3-demo-bucket
```

출력:

```
upload: test.txt to s3://amzn-s3-demo-bucket/test.txt
upload: test2.txt to s3://amzn-s3-demo-bucket/test2.txt
```

예시 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 예시에서는 사용자가 amzn-s3-demo-bucket 버킷을 amzn-s3-demo-bucket2 버킷에 동기화합니다. amzn-s3-demo-bucket 버킷에는 test.txt 및 test2.txt 객체가 포함되어 있습니다. amzn-s3-demo-bucket2 버킷에 객체가 없습니다.

```
aws s3 sync s3://amzn-s3-demo-bucket s3://amzn-s3-demo-bucket2
```

출력:

```
copy: s3://amzn-s3-demo-bucket/test.txt to s3://amzn-s3-demo-bucket2/test.txt
copy: s3://amzn-s3-demo-bucket/test2.txt to s3://amzn-s3-demo-bucket2/test2.txt
```

예시 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 객체의 마지막 수정 시간이 로컬 파일의 마지막 수정 시간보다 최신인 경우, 또는 S3 객체가 로컬 디렉터리에 없는 경우에는 다운로드해야 합니다. S3에서 객체를 다운로드하면 로컬 파일의 마지막 수정 시간이 S3 객체의 마지막 수정 시간으로 변경됩니다. 이 예시에서는 사용자가 `amzn-s3-demo-bucket` 버킷을 현재 로컬 디렉터리와 동기화합니다. `amzn-s3-demo-bucket` 버킷에는 `test.txt` 및 `test2.txt` 객체가 포함되어 있습니다. 현재 로컬 디렉터리에는 파일이 없습니다.

```
aws s3 sync s3://amzn-s3-demo-bucket .
```

출력:

```
download: s3://amzn-s3-demo-bucket/test.txt to test.txt
download: s3://amzn-s3-demo-bucket/test2.txt to test2.txt
```

예시 4: 모든 로컬 객체를 지정된 버킷에 동기화하고 일치하지 않는 모든 파일 삭제

다음 sync 명령은 로컬 파일을 S3에 업로드하여 지정된 접두사 및 버킷 아래의 객체를 로컬 디렉터리의 파일에 동기화합니다. `--delete` 파라미터를 사용하면 지정된 접두사와 버킷 아래에 존재하지만 로컬 디렉토리에 존재하지 않는 모든 파일이 삭제됩니다. 이 예시에서는 사용자가 `amzn-s3-demo-bucket` 버킷을 로컬 현재 디렉터리에 동기화합니다. 현재 로컬 디렉터리에는 `test.txt` 및 `test2.txt` 파일이 들어 있습니다. `amzn-s3-demo-bucket` 버킷에는 `test3.txt` 객체가 포함되어 있습니다.

```
aws s3 sync . s3://amzn-s3-demo-bucket \
    --delete
```

출력:

```
upload: test.txt to s3://amzn-s3-demo-bucket/test.txt
upload: test2.txt to s3://amzn-s3-demo-bucket/test2.txt
delete: s3://amzn-s3-demo-bucket/test3.txt
```

예시 5: ``.jpg`` 파일을 제외한 모든 로컬 객체를 지정된 버킷에 동기화

다음 sync 명령은 로컬 파일을 S3에 업로드하여 지정된 접두사 및 버킷 아래의 객체를 로컬 디렉터리의 파일에 동기화합니다. `--exclude` 파라미터로 인해 S3 및 로컬 모두에 존재하는 패턴과 일치하는 모든 파일은 동기화에서 제외됩니다. 이 예시에서는 사용자가 `amzn-s3-demo-bucket` 버킷을 로컬 현재 디렉터리에 동기화합니다. 현재 로컬 디렉터리에는 `test.jpg` 및 `test2.txt` 파일이 들어 있습니다. `amzn-s3-demo-bucket` 버킷에는 로컬 `test.jpg`와 다른 크기의 `test.jpg` 객체가 포함되어 있습니다.

```
aws s3 sync . s3://amzn-s3-demo-bucket \
    --exclude "*.jpg"
```

출력:

```
upload: test2.txt to s3://amzn-s3-demo-bucket/test2.txt
```

예제 6: 지정된 디렉터리 파일을 제외한 모든 로컬 객체를 지정된 버킷에 동기화

다음 sync 명령은 S3 객체를 다운로드하여 로컬 디렉터리 아래의 파일을 지정된 접두사 및 버킷 아래의 객체에 동기화합니다. 이 예시에서는 `--exclude` 파라미터 플래그를 사용하여 지정된 디렉터리와 S3 접두사를 sync 명령에서 제외합니다. 이 예시에서는 사용자가 로컬 현재 디렉터리를 `amzn-s3-demo-bucket` 버킷에 동기화합니다. 현재 로컬 디렉터리에는 `test.txt` 및 `another/test2.txt` 파일이 들어 있습니다. `amzn-s3-demo-bucket` 버킷에는 `another/test5.txt` 및 `test1.txt` 객체가 포함되어 있습니다.

```
aws s3 sync s3://amzn-s3-demo-bucket/ . \
    --exclude "*another/*"
```

출력:

```
download: s3://amzn-s3-demo-bucket/test1.txt to test1.txt
```

예시 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
```

예시 8: S3 액세스 포인트에 동기화

다음 sync 명령은 현재 디렉터리를 액세스 포인트(myaccesspoint)와 동기화합니다.

```
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 세부 정보는 AWS CLI 명령 참조의 Sync 섹션을 참조하세요.

`upload-part-copy`

다음 코드 예시에서는 `upload-part-copy`의 사용 방법을 보여줍니다.

AWS CLI

기존 객체의 데이터를 데이터 소스로 복사하여 객체의 일부를 업로드하는 방법

다음 `upload-part-copy` 예시에서는 기존 객체의 데이터를 데이터 소스로 복사하여 파트를 업로드합니다.

```
aws s3api upload-part-copy \
    --bucket amzn-s3-demo-bucket \
    --key "Map_Data_June.mp4" \
```

```
    --copy-source "amzn-s3-demo-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\""
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 UploadPartCopy 섹션을 참조하세요.

upload-part

다음 코드 예시에서는 upload-part의 사용 방법을 보여줍니다.

AWS CLI

다음 명령은 create-multipart-upload 명령으로 시작된 멀티파트 업로드의 첫 번째 부분을 업로드합니다.

```
aws s3api upload-part --bucket amzn-s3-demo-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. 업로드 ID는 create-multipart-upload에 의해 반환되며 list-multipart-uploads를 사용하여 가져올 수도 있습니다. 멀티파트 업로드를 생성할 때 버킷과 키가 지정됩니다.

출력:

```
{
    "ETag": "\"e868e0f4719e394144ef36531ee6824c\""
}
```

나중을 위해 각 파트의 ETag 값을 저장하세요. 멀티파트 업로드를 완료하는 데 필요합니다.

- API 세부 정보는 AWS CLI 명령 참조의 UploadPart를 참조하세요.

`website`

다음 코드 예시는 `website`의 사용 방법을 보여 줍니다.

AWS CLI

정적 웹 사이트로 Amazon S3 버킷 구성

다음 명령은 `amzn-s3-demo-bucket` 버킷의 정적 웹 사이트 구성을 검색합니다. 인덱스 문서 옵션은 방문자가 웹사이트 URL로 이동할 때 `amzn-s3-demo-bucket`으로 연결되는 파일을 지정합니다. 이 경우 버킷은 us-west-2 리전에 있으므로 사이트는 `http://amzn-s3-demo-bucket.s3-website-us-west-2.amazonaws.com`에 표시됩니다.

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://amzn-s3-demo-bucket/ \
    --index-document index.html \
    --error-document error.html
```

Amazon S3에서 정적 웹사이트를 호스팅하는 방법에 대한 자세한 내용은 Amazon Simple Storage Service 개발자 안내서의 정적 웹사이트 호스팅을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 Website 섹션을 참조하세요.

AWS CLI를 사용한 Amazon S3 Control 예시

다음 코드 예시에서는 Amazon S3 Control과 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

create-access-point

다음 코드 예시에서는 `create-access-point`의 사용 방법을 보여줍니다.

AWS CLI

액세스 포인트 생성

다음 `create-access-point` 예시에서는 계정 123456789012의 버킷 `business-records`에 `finance-ap`라는 액세스 포인트를 생성합니다. 이 예시를 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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateAccessPoint를 참조하세요.

create-job

다음 코드 예시에서는 `create-job`의 사용 방법을 보여줍니다.

AWS CLI

Amazon S3 Batch Operations 작업 생성

다음 `create-job` 예시에서는 Amazon S3 Batch Operations 작업을 생성하고 객체를 `confidential` in the bucket ``employee-records`로 태그 지정합니다.

```
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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateJob을 참조하세요.

delete-access-point-policy

다음 코드 예시에서는 delete-access-point-policy의 사용 방법을 보여줍니다.

AWS CLI

액세스 포인트 정책 삭제

다음 delete-access-point-policy 예시에서는 계정 123456789012의 finance-ap라는 액세스 포인트에서 액세스 포인트 정책을 삭제합니다. 이 예시를 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 Access Points를 사용한 데이터 액세스 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteAccessPointPolicy를 참조하세요.

delete-access-point

다음 코드 예시에서는 delete-access-point의 사용 방법을 보여줍니다.

AWS CLI

액세스 포인트 삭제

다음 delete-access-point 예시에서는 계정 123456789012에서 finance-ap라는 액세스 포인트를 삭제합니다. 이 예시를 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 Access Points를 사용한 데이터 액세스 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteAccessPoint를 참조하세요.

delete-public-access-block

다음 코드 예시에서는 delete-public-access-block의 사용 방법을 보여줍니다.

AWS CLI

계정의 퍼블릭 액세스 차단 설정 삭제

다음 delete-public-access-block 예시에서는 지정된 계정의 퍼블릭 액세스 차단 설정을 삭제합니다.

```
aws s3control delete-public-access-block \
    --account-id 123456789012
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 DeletePublicAccessBlock을 참조하세요.

describe-job

다음 코드 예시에서는 describe-job의 사용 방법을 보여줍니다.

AWS CLI

Amazon S3 Batch Operations 작업 설명

다음 describe-job 예시에서는 지정된 Batch Operations 작업의 구성 파라미터와 상태를 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeJob을 참조하세요.

get-access-point-policy-status

다음 코드 예시에서는 get-access-point-policy-status의 사용 방법을 보여줍니다.

AWS CLI

액세스 포인트 정책 상태 가져오기

다음 get-access-point-policy-status 예시에서는 계정 123456789012에서 finance-ap라는 액세스 포인트의 액세스 포인트 정책 상태를 가져옵니다. 액세스 포인트 정책 상태는 액세스 포인트의 정책이 퍼블릭 액세스를 허용하는지 여부를 나타냅니다. 이 예시를 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 정책을 퍼블릭으로 간주하는 경우에 대한 자세한 내용은 Amazon Simple Storage Service 개발자 안내서의 '퍼블릭'의 의미를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetAccessPointPolicyStatus를 참조하세요.

get-access-point-policy

다음 코드 예시에서는 get-access-point-policy의 사용 방법을 보여줍니다.

AWS CLI

액세스 포인트 정책 가져오기

다음 get-access-point-policy 예시에서는 계정 123456789012의 finance-ap라는 액세스 포인트에서 액세스 포인트 정책을 가져옵니다. 이 예시를 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 Access Points를 사용한 데이터 액세스 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetAccessPointPolicy를 참조하세요.

get-access-point

다음 코드 예시에서는 get-access-point의 사용 방법을 보여줍니다.

AWS CLI

액세스 포인트의 구성 세부 정보 가져오기

다음 get-access-point 예시에서는 계정 123456789012에서 finance-ap라는 액세스 포인트의 구성 세부 정보를 반환합니다. 이 예시를 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 Access Points를 사용한 데이터 액세스 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetAccessPoint를 참조하세요.

get-multi-region-access-point-routes

다음 코드 예시에서는 get-multi-region-access-point-routes의 사용 방법을 보여줍니다.

AWS CLI

현재 다중 리전 액세스 포인트 라우팅 구성에 대한 쿼리

다음 get-multi-region-access-point-routes 예시에서는 지정된 다중 리전 액세스 포인트의 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": "amzn-s3-demo-bucket1",
            "Region": "ap-southeast-2",
            "TrafficDialPercentage": 100
        },
        {
            "Bucket": "amzn-s3-demo-bucket2",
            "Region": "us-west-1",
            "TrafficDialPercentage": 0
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetMultiRegionAccessPointRoutes를 참조하세요.

get-public-access-block

다음 코드 예시에서는 get-public-access-block의 사용 방법을 보여줍니다.

AWS CLI

계정의 퍼블릭 액세스 차단 설정 나열

다음 get-public-access-block 예시에서는 지정된 계정의 퍼블릭 액세스 차단 설정을 표시합니다.

```
aws s3control get-public-access-block \
    --account-id 123456789012
```

출력:

```
{
   "PublicAccessBlockConfiguration": {
      "BlockPublicPolicy": true,
      "RestrictPublicBuckets": true,
      "IgnorePublicAcls": true,
      "BlockPublicAcls": true
   }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetPublicAccessBlock을 참조하세요.

list-access-points

다음 코드 예시에서는 list-access-points의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 계정의 모든 액세스 포인트 목록 가져오기

다음 list-access-points 예시에서는 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"
        }
    ]
}
```

예시 2: 버킷의 모든 액세스 포인트 목록 가져오기

다음 `list-access-points` 예시에서는 계정 123456789012에서 소유한 버킷 `external-docs`에 연결된 모든 액세스 포인트의 목록을 표시합니다.

```
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 Access Points를 사용한 데이터 액세스 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListAccessPoints를 참조하세요.

list-jobs

다음 코드 예시에서는 `list-jobs`의 사용 방법을 보여줍니다.

AWS CLI

계정 Amazon S3 Batch Operations 작업 나열

다음 `list-jobs` 예시에서는 지정된 계정의 모든 최근 Batch Operations 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
            },
        ]
    }
```

- API 세부 정보는 AWS CLI 명령 참조의 ListJobs를 참조하세요.

put-access-point-policy

다음 코드 예시에서는 put-access-point-policy의 사용 방법을 보여줍니다.

AWS CLI

액세스 포인트 정책 설정

다음 put-access-point-policy 예시에서는 계정 123456789012의 액세스 포인트 finance-ap에 지정된 액세스 포인트 정책을 배치합니다. 액세스 포인트 finance-ap에 이미 정책이 있는 경우 이 명령은 기존 정책을 해당 명령에 지정된 정책으로 바꿉니다. 이 예시를 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 Access Points를 사용한 데이터 액세스 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutAccessPointPolicy를 참조하세요.

put-public-access-block

다음 코드 예시에서는 put-public-access-block의 사용 방법을 보여줍니다.

AWS CLI

계정의 퍼블릭 액세스 차단 설정 편집

다음 put-public-access-block 예시에서는 지정된 계정의 모든 퍼블릭 액세스 차단 설정을 true로 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.

- API 세부 정보는 AWS CLI 명령 참조의 PutPublicAccessBlock 섹션을 참조하세요.

submit-multi-region-access-point-routes

다음 코드 예시에서는 submit-multi-region-access-point-routes의 사용 방법을 보여줍니다.

AWS CLI

다중 리전 액세스 포인트 라우팅 구성 업데이트

다음 submit-multi-region-access-point-routes 예시에서는 다중 리전 액세스 포인트의 ap-southeast-2 리전에서 amzn-s3-demo-bucket1 및 amzn-s3-demo-bucket2의 라우팅 상태를 업데이트합니다.

```
aws s3control submit-multi-region-access-point-routes \
    --region ap-southeast-2 \
    --account-id 111122223333 \
    --mrap MultiRegionAccessPoint_ARN \
    --route-updates Bucket=amzn-s3-demo-
bucket1,TrafficDialPercentage=100 Bucket=amzn-s3-demo-
bucket2,TrafficDialPercentage=0
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 SubmitMultiRegionAccessPointRoutes를 참조하세요.

update-job-priority

다음 코드 예시에서는 update-job-priority의 사용 방법을 보여줍니다.

AWS CLI

Amazon S3 Batch Operations의 작업 우선 순위 업데이트

다음 update-job-priority 예시에서는 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateJobPriority를 참조하세요.

`update-job-status`

다음 코드 예시에서는 `update-job-status`의 사용 방법을 보여줍니다.

AWS CLI

Amazon S3 Batch Operations의 작업 상태 업데이트

다음 `update-job-status` 예시에서는 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` 예시에서는 승인을 기다리는 지정된 작업을 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` 예시에서는 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateJobStatus를 참조하세요.

AWS CLI를 사용한 S3 Glacier 예제

다음 코드 예제는 S3 Glacier와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

abort-multipart-upload

다음 코드 예시에서는 `abort-multipart-upload` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

다음 명령은 `my-vault`라는 이름의 저장소에 진행 중인 멀티파트 업로드를 삭제합니다.

```
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-uploads`를 사용하여 가져올 수도 있습니다.

AWS CLI를 사용하여 Amazon Glacier에 멀티파트 업로드를 수행하는 방법에 대한 자세한 내용은 AWS CLI 사용 설명서의 Amazon Glacier 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AbortMultipartUpload를 참조하세요.

`abort-vault-lock`

다음 코드 예시에서는 `abort-vault-lock` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

진행 중인 저장소 잠금 프로세스 중단

다음 `abort-vault-lock` 예시에서는 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 Abort Vault Lock (DELETE lock-policy)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AbortVaultLock 섹션을 참조하세요.

`add-tags-to-vault`

다음 코드 예시에서는 `add-tags-to-vault` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

다음 명령은 이름이 지정된 `my-vault`에 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.

- API 세부 정보는 AWS CLI 명령 참조의 AddTagsToVault 섹션을 참조하세요.

`complete-multipart-upload`

다음 코드 예시에서는 `complete-multipart-upload` 코드를 사용하는 방법을 보여줍니다.

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-uploads`를 사용하여 가져올 수도 있습니다. 체크섬 파라미터는 아카이브의 SHA-256 트리 해시를 16진수로 사용합니다.

트리 해시를 계산하는 방법에 대한 지침을 비롯해 AWS CLI를 사용하여 Amazon Glacier에 멀티파트 업로드를 수행하는 방법에 대한 자세한 내용은 AWS CLI 사용 설명서의 Amazon Glacier 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CompleteMultipartUpload를 참조하세요.

`complete-vault-lock`

다음 코드 예시에서는 `complete-vault-lock` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

진행 중인 저장소 잠금 프로세스 완료

다음 `complete-vault-lock` 예시에서는 지정된 저장소의 진행 중인 잠금 진행 상황을 완료하고 저장소 잠금의 잠금 상태를 `Locked`로 설정합니다. `initiate-lock-process`를 실행할 때 `lock-id` 파라미터 값을 가져옵니다.

```
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 Complete Vault Lock (POST lockId)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CompleteVaultLock 섹션을 참조하세요.

create-vault

다음 코드 예시에서는 `create-vault` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

다음 명령은 `my-vault`라는 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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateVault를 참조하세요.

delete-archive

다음 코드 예시에서는 `delete-archive` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

볼트에서 아카이브를 삭제하는 방법

다음 `delete-archive` 예시에서는 `example_vault`에서 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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteArchive를 참조하세요.

delete-vault-access-policy

다음 코드 예시에서는 delete-vault-access-policy 코드를 사용하는 방법을 보여줍니다.

AWS CLI

저장소의 액세스 정책 제거

다음 delete-vault-access-policy 예시에서는 지정된 저장소에 대한 액세스 정책을 제거합니다.

```
aws glacier delete-vault-access-policy \
    --account-id 111122223333 \
    --vault-name example_vault
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteVaultAccessPolicy 섹션을 참조하세요.

delete-vault-notifications

다음 코드 예시에서는 delete-vault-notifications 코드를 사용하는 방법을 보여줍니다.

AWS CLI

저장소에 대한 SNS 알림 제거

다음 delete-vault-notifications 예시에서는 지정된 볼트에 대해 Amazon Simple Notification Service(SNS)에서 전송한 알림을 제거합니다.

```
aws glacier delete-vault-notifications \
    --account-id 111122223333 \
    --vault-name example_vault
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteVaultNotifications를 참조하세요.

delete-vault

다음 코드 예시에서는 delete-vault 코드를 사용하는 방법을 보여줍니다.

AWS CLI

다음 명령은 my-vault라는 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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteVault를 참조하세요.

describe-job

다음 코드 예시에서는 describe-job 코드를 사용하는 방법을 보여줍니다.

AWS CLI

다음 명령은 my-vault라는 저장소의 인벤토리 검색 작업에 대한 정보를 검색합니다.

```
aws glacier describe-job --account-id - --vault-name my-
vault --job-id zbxcm3Z_3z5UkoroF7SuZKrxgGoDc3RloGduS7Eg-
RO47Yc6FxsdGBgf_Q2DK5Ejh18CnTS5XW4_XqlNHS61dsO4CnMW
```

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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeJob을 참조하세요.

`describe-vault`

다음 코드 예시에서는 `describe-vault` 코드를 사용하는 방법을 보여줍니다.

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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeVault 섹션을 참조하세요.

`get-data-retrieval-policy`

다음 코드 예시에서는 `get-data-retrieval-policy` 코드를 사용하는 방법을 보여줍니다.

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.

- API 세부 정보는 AWS CLI 명령 참조의 GetDataRetrievalPolicy 섹션을 참조하세요.

get-job-output

다음 코드 예시에서는 `get-job-output` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

다음 명령은 볼트 인벤토리 작업의 출력을 `output.json`라는 현재 디렉터리의 파일에 저장합니다.

```
aws glacier get-job-output --account-id - --vault-name my-vault --job-id zbxcm3Z_3z5UkoroF7SuZKrxgGoDc3RloGduS7Eg-RO47Yc6FxsdGBgf_Q2DK5Ejh18CnTS5XW4_XqlNHS61dsO4CnMW output.json
```

`job-id`는 `aws glacier list-jobs`의 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
```

- API 세부 정보는 AWS CLI 명령 참조의 [GetJobOutput](https://docs.aws.amazon.com/goto/aws-cli/glacier-2012-06-01/GetJobOutput)을 참조하세요.

`get-vault-access-policy`

다음 코드 예시에서는 `get-vault-access-policy` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

저장소의 액세스 정책 검색

다음 `get-vault-access-policy` 예시에서는 지정된 저장소에 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\"}]}"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [GetVaultAccessPolicy](https://docs.aws.amazon.com/goto/aws-cli/glacier-2012-06-01/GetVaultAccessPolicy) 섹션을 참조하세요.

`get-vault-lock`

다음 코드 예시에서는 `get-vault-lock` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

저장소 잠금의 세부 정보 가져오기

다음 get-vault-lock 예시에서는 지정된 저장소에 대한 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.

- API 세부 정보는 AWS CLI 명령 참조의 GetVaultLock 섹션을 참조하세요.

get-vault-notifications

다음 코드 예시에서는 get-vault-notifications 코드를 사용하는 방법을 보여줍니다.

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.

- API 세부 정보는 AWS CLI 명령 참조의 GetVaultNotifications를 참조하세요.

initiate-job

다음 코드 예시에서는 `initiate-job` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

다음 명령은 `my-vault` 저장소 인벤토리를 가져오는 작업을 시작합니다.

```
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 `my-vault` 저장소에서 아카이브를 가져오는 작업을 시작합니다.

```
aws glacier initiate-job --account-id - --vault-name my-vault --job-
parameters file://job-archive-retrieval.json
```

job-archive-retrieval.json은 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"
}
```

아카이브 ID는 aws glacier upload-archive 및 aws glacier get-job-output 출력에 표시됩니다.

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 시작 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 InitiateJob 섹션을 참조하세요.

initiate-multipart-upload

다음 코드 예시에서는 initiate-multipart-upload 코드를 사용하는 방법을 보여줍니다.

AWS CLI

다음 명령은 파일당 파트 크기가 1MB(1024 x 1024바이트)인 my-vault 저장소에 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. `aws glacier upload-multipart-part`를 사용하여 아카이브의 각 부분을 업로드할 때 업로드 ID를 사용합니다. AWS CLI를 사용하여 Amazon Glacier에 멀티파트 업로드를 수행하는 방법에 대한 자세한 내용은 AWS CLI 사용 설명서의 Amazon Glacier 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 InitiateMultipartUpload 섹션을 참조하세요.

`initiate-vault-lock`

다음 코드 예시에서는 `initiate-vault-lock` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

저장소 잠금 프로세스 시작

다음 `initiate-vault-lock` 예시에서는 지정된 저장소에 저장소 잠금 정책을 설치하고 저장소 잠금의 잠금 상태를 `InProgress`로 설정합니다. `complete-vault-lock`을 호출하여 저장소 잠금 프로세스를 마치면 저장소 잠금 상태가 `Locked`로 설정됩니다.

```
aws glacier initiate-vault-lock \
    --account-id - \
    --vault-name MyVaultName \
    --policy file://vault_lock_policy.json
```

`vault_lock_policy.json`의 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 Initiate Vault Lock (POST lock-policy)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 InitiateVaultLock 섹션을 참조하세요.

list-jobs

다음 코드 예시에서는 list-jobs 코드를 사용하는 방법을 보여줍니다.

AWS CLI

다음 명령은 my-vault라는 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListJobs를 참조하세요.

`list-multipart-uploads`

다음 코드 예시에서는 `list-multipart-uploads` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

다음 명령은 `my-vault` 이름의 저장소에 대해 진행 중인 모든 멀티파트 업로드를 보여줍니다.

```
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.

AWS CLI를 사용하여 Amazon Glacier에 멀티파트 업로드를 수행하는 방법에 대한 자세한 내용은 AWS CLI 사용 설명서의 Amazon Glacier 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListMultipartUploads를 참조하세요.

`list-parts`

다음 코드 예시에서는 `list-parts` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

다음 명령은 `my-vault` 저장소에 대한 멀티파트 업로드에 대해 업로드된 부분을 나열합니다.

```
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.

AWS CLI를 사용하여 Amazon Glacier에 멀티파트 업로드를 수행하는 방법에 대한 자세한 내용은 AWS CLI 사용 설명서의 Amazon Glacier 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListParts 섹션을 참조하세요.

`list-provisioned-capacity`

다음 코드 예시에서는 `list-provisioned-capacity` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

프로비저닝된 용량 단위를 검색하는 방법

다음 `list-provisioned-capacity` 예시에서는 지정된 계정에 프로비저닝된 용량 단위의 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListProvisionedCapacity 섹션을 참조하세요.

`list-tags-for-vault`

다음 코드 예시에서는 `list-tags-for-vault` 코드를 사용하는 방법을 보여줍니다.

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.

- API 세부 정보는 AWS CLI 명령 참조의 [ListTagsForVault](#) 섹션을 참조하세요.

list-vaults

다음 코드 예시에서는 list-vaults 코드를 사용하는 방법을 보여줍니다.

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.

- API 세부 정보는 AWS CLI 명령 참조의 [ListVaults](#)를 참조하세요.

purchase-provisioned-capacity

다음 코드 예시에서는 purchase-provisioned-capacity 코드를 사용하는 방법을 보여줍니다.

AWS CLI

프로비저닝된 용량 단위를 구매하는 방법

다음 purchase-provisioned-capacity 예시에서는 프로비저닝된 용량 단위를 구매합니다.

```
aws glacier purchase-provisioned-capacity \
    --account-id 111122223333
```

출력:

```
{
    "capacityId": "HpASAuvfRFiVDbOjMfEIcr8K"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 PurchaseProvisionedCapacity 섹션을 참조하세요.

`remove-tags-from-vault`

다음 코드 예시에서는 `remove-tags-from-vault` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

다음 명령은 `my-vault`라는 저장소에서 `date` 키가 있는 태그를 제거합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 RemoveTagsFromVault 섹션을 참조하세요.

`set-data-retrieval-policy`

다음 코드 예시에서는 `set-data-retrieval-policy` 코드를 사용하는 방법을 보여줍니다.

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`은 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 인라인 JSON을 사용하여 데이터 검색 정책을 `FreeTier`로 설정합니다

```
aws glacier set-data-retrieval-policy --account-id - --policy '{"Rules":
[{"Strategy":"FreeTier"}]}'
```

정책 형식 지정에 대한 자세한 내용은 Amazon Glacier API 참조에서 데이터 검색 정책 설정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 SetDataRetrievalPolicy 섹션을 참조하세요.

`set-vault-access-policy`

다음 코드 예시에서는 `set-vault-access-policy` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

저장소의 액세스 정책 설정

다음 `set-vault-access-policy` 예시에서는 권한 정책을 지정된 저장소에 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.

- API 세부 정보는 AWS CLI 명령 참조의 SetVaultNotifications 섹션을 참조하세요.

set-vault-notifications

다음 코드 예시에서는 set-vault-notifications 코드를 사용하는 방법을 보여줍니다.

AWS CLI

다음 명령은 my-vault라는 볼트에 대한 SNS 알림을 구성합니다.

```
aws glacier set-vault-notifications --account-id - --vault-name my-vault --vault-notification-config file://notificationconfig.json
```

notificationconfig.json은 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.

- API 세부 정보는 AWS CLI 명령 참조의 SetVaultNotifications를 참조하세요.

upload-archive

다음 코드 예시에서는 upload-archive 코드를 사용하는 방법을 보여줍니다.

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.

- API 세부 정보는 AWS CLI 명령 참조의 UploadArchive를 참조하세요.

`upload-multipart-part`

다음 코드 예시에서는 `upload-multipart-part` 코드를 사용하는 방법을 보여줍니다.

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 완성된 아카이브에서 부분이 차지하는 바이트를 나타내는 HTTP 콘텐츠 범위를 사용합니다. 업로드 ID는 `aws glacier initiate-multipart-upload` 명령으로 반환되며 `aws glacier list-multipart-uploads`를 사용하여 가져올 수도 있습니다.

AWS CLI를 사용하여 Amazon Glacier에 멀티파트 업로드를 수행하는 방법에 대한 자세한 내용은 AWS CLI 사용 설명서의 Amazon Glacier 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UploadMultipartPart를 참조하세요.

AWS CLI를 사용한 Secrets Manager 예시

다음 코드 예시는 Secrets Manager와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

작업

batch-get-secret-value

다음 코드 예시에서는 batch-get-secret-value 코드를 사용하는 방법을 보여줍니다.

AWS CLI

예시 1: 이름별로 나열된 보안 암호 그룹의 보안 암호 값 검색

다음 batch-get-secret-value 예시에서는 현재 보안 암호 값을 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 AWS Secrets Manager 사용 설명서의 Retrieve a group of secrets in a batch를 참조하세요.

예시 2: 필터로 선택한 보안 암호 그룹에 대한 보안 암호 값 검색

다음 batch-get-secret-value 예시에서는 계정의 이름에 있는 보안 암호 값을 가져옵니다. 이름별 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 AWS Secrets Manager 사용 설명서의 Retrieve a group of secrets in a batch를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 BatchGetSecretValue 섹션을 참조하세요.

cancel-rotate-secret

다음 코드 예시에서는 `cancel-rotate-secret` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

보안 암호에 대한 자동 교체 끄기

다음 `cancel-rotate-secret` 예시에서는 보안 암호에 대한 자동 교체를 끕니다. 교체를 재개하려면 `rotate-secret`을 호출합니다.

```
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 Rotate a secret를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CancelRotateSecret 섹션을 참조하세요.

create-secret

다음 코드 예시에서는 `create-secret` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

예시 1: JSON 파일의 보안 인증 정보로 보안 암호 생성

다음 `create-secret` 예시에서는 파일의 자격 증명을 사용하여 시크릿을 생성합니다. 자세한 내용은 AWS CLI 사용자 안내서의 파일에서 AWS CLI 파라미터 로드를 참조하세요.

```
aws secretsmanager create-secret \
    --name MyTestSecret \
    --secret-string file://mycreds.json
```

`mycreds.json`의 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 사용자 안내서의 시크릿 생성을 참조하세요.

예시 2: 보안 암호 생성

다음 `create-secret` 예시에서는 2개의 키와 값 페어로 시크릿을 생성합니다. 명령 쉘에 명령을 입력하면 명령 기록이 액세스되거나 유틸리티가 명령 파라미터에 액세스할 위험이 있습니다. 명령에 보안 암호 값이 포함된 경우 이 문제가 발생할 수 있습니다. 자세한 내용은 Secrets Manager 사용 설명서의 Mitigate the risks of using command-line tools to store Secrets를 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 사용자 안내서의 시크릿 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateSecret을 참조하세요.

`delete-resource-policy`

다음 코드 예시에서는 `delete-resource-policy` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

보안 암호에 연결된 리소스 기반 정책 삭제

다음 `delete-resource-policy` 예시에서는 시크릿에 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 정보는 Secrets Manager 사용 설명서의 Authentication and access control를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteResourcePolicy 섹션을 참조하세요.

`delete-secret`

다음 코드 예시에서는 `delete-secret` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

예시 1: 시크릿 삭제

다음 delete-secret 예시에서는 시크릿을 삭제합니다. DeletionDate 응답 필드의 날짜 및 시간 전까지 restore-secret으로 시크릿을 복구할 수 있습니다. 다른 리전에 복제된 시크릿을 삭제하려면 먼저 remove-regions-from-replication으로 해당 복제본을 삭제한 다음 delete-secret을 직접적으로 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 사용자 안내서의 시크릿 삭제를 참조하세요.

예시 2: 시크릿 즉시 삭제

다음 delete-secret 예시에서는 복구 기간 없이 즉시 시크릿을 삭제합니다. 이러한 시크릿은 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 사용자 안내서의 시크릿 삭제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteSecret](https://docs.aws.amazon.com/goto/aws-cli/secretsmanager-2017-10-17/DeleteSecret)을 참조하세요.

describe-secret

다음 코드 예시에서는 describe-secret 코드를 사용하는 방법을 보여줍니다.

AWS CLI

시크릿의 세부 정보 가져오기

다음 describe-secret 예시에서는 시크릿의 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 사용자 안내서의 시크릿을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeSecret을 참조하세요.

get-random-password

다음 코드 예시에서는 get-random-password 코드를 사용하는 방법을 보여줍니다.

AWS CLI

임의 암호 생성

다음 get-random-password 예시에서는 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 Create and manage Secrets Manager를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetRandomPassword 섹션을 참조하세요.

get-resource-policy

다음 코드 예시에서는 `get-resource-policy` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

보안 암호에 연결된 리소스 기반 정책 검색

다음 `get-resource-policy` 예시에서는 시크릿에 연결된 리소스 기반 정책을 가져옵니다.

```
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 정보는 Secrets Manager 사용 설명서의 Authentication and access control를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetResourcePolicy 섹션을 참조하세요.

get-secret-value

다음 코드 예시에서는 get-secret-value 코드를 사용하는 방법을 보여줍니다.

AWS CLI

예시 1: 시크릿의 암호화된 시크릿 값 가져오기

다음 get-secret-value 예시에서는 현재 시크릿 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 사용자 안내서의 [시크릿 가져오기]를 참조하세요.

예시 2: 이전 시크릿 값 가져오기

다음 get-secret-value 예시에서는 이전 시크릿 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 사용자 안내서의 시크릿 가져오기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetSecretValue를 참조하세요.

list-secret-version-ids

다음 코드 예시에서는 `list-secret-version-ids` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

보안 암호와 연결된 모든 보안 암호 버전을 나열하는 방법

다음 `list-secret-version-ids` 예시에서는 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 Version을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListSecretVersionIds 섹션을 참조하세요.

list-secrets

다음 코드 예시에서는 `list-secrets` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

예시 1: 계정의 시크릿 나열

다음 `list-secrets` 예시에서는 계정에 있는 시크릿 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 사용자 안내서의 시크릿 찾기를 참조하세요.

예시 2: 계정의 시크릿 목록 필터링

다음 `list-secrets` 예시에서는 계정에서 이름에 `Test`가 있는 시크릿 목록을 가져옵니다. 이름별 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 사용자 안내서의 시크릿 찾기를 참조하세요.

예시 3: 다른 서비스에서 관리하는 계정의 시크릿 나열

다음 `list-secrets` 예시에서는 Amazon RDS에서 관리하는 계정의 시크릿을 반환합니다.

```
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 사용자 안내서의 다른 서비스에서 관리하는 시크릿을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListSecrets를 참조하세요.

put-resource-policy

다음 코드 예시에서는 put-resource-policy 코드를 사용하는 방법을 보여줍니다.

AWS CLI

보안 암호에 리소스 기반 정책 추가

다음 put-resource-policy 예시에서는 시크릿에 권한 정책을 추가하여 해당 정책이 시크릿에 대한 광범위한 액세스 권한을 제공하지 않는지 먼저 확인합니다. 파일에서 해당 정책을 읽습니다. 자세한 내용은 AWS CLI 사용자 안내서의 파일에서 AWS CLI 파라미터 로드를 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 Attach a permissions policy to a secret을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutResourcePolicy 섹션을 참조하세요.

put-secret-value

다음 코드 예시에서는 put-secret-value 코드를 사용하는 방법을 보여줍니다.

AWS CLI

예시 1: 시크릿에 새 암호 값 저장

다음 put-secret-value 예시에서는 2개의 키와 값 페어로 새 버전의 시크릿을 생성합니다.

```
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 사용자 안내서의 시크릿 수정을 참조하세요.

예시 2: JSON 파일의 자격 증명으로 새 시크릿 값 저장

다음 `put-secret-value` 예시에서는 파일의 자격 증명으로 새 버전의 시크릿을 생성합니다. 자세한 내용은 AWS CLI 사용자 안내서의 파일에서 AWS CLI 파라미터 로드를 참조하세요.

```
aws secretsmanager put-secret-value \
    --secret-id MyTestSecret \
    --secret-string file://mycreds.json
```

`mycreds.json`의 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 사용자 안내서의 시크릿 수정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutSecretValue를 참조하세요.

remove-regions-from-replication

다음 코드 예시에서는 remove-regions-from-replication 코드를 사용하는 방법을 보여줍니다.

AWS CLI

복제본 보안 암호 삭제

다음 remove-regions-from-replication 예시에서는 eu-west-3의 복제 시크릿을 삭제합니다. 다른 리전에 복제된 기본 시크릿을 삭제하려면 먼저 복제본을 삭제한 다음 delete-secret을 직접적으로 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 Delete a replica secret을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RemoveRegionsFromReplication 섹션을 참조하세요.

replicate-secret-to-regions

다음 코드 예시에서는 replicate-secret-to-regions 코드를 사용하는 방법을 보여줍니다.

AWS CLI

다른 리전으로 보안 암호 복제

다음 `replicate-secret-to-regions` 예시에서는 eu-west-3으로 시크릿을 복제합니다. 복제본은 AWS 관리형 키 `aws/secretsmanager`로 암호화됩니다.

```
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 Replicate a secret to another Region을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ReplicateSecretToRegions 섹션을 참조하세요.

`restore-secret`

다음 코드 예시에서는 `restore-secret` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

이전에 삭제한 보안 암호 복원

다음 `restore-secret` 예시에서는 이전에 삭제가 예정된 시크릿을 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 사용자 안내서의 시크릿 삭제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RestoreSecret 섹션을 참조하세요.

rotate-secret

다음 코드 예시에서는 `rotate-secret` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

예시 1: 보안 암호에 대한 자동 교체를 구성하고 시작하는 방법

다음 `rotate-secret` 예시에서는 보안 암호의 자동 교체를 구성하고 시작합니다. Secrets Manager는 보안 암호를 즉시 한 번 교체한 다음 2시간 간격으로 8시간마다 교체합니다. 출력에 교체 결과 생성된 새 보안 암호 버전의 `VersionId`가 표시됩니다.

```
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 Rotate secrets를 참조하세요.

예시 2: 교체 간격에서 자동 교체를 구성하고 시작하는 방법

다음 `rotate-secret` 예시에서는 보안 암호의 자동 교체를 구성하고 시작합니다. Secrets Manager는 즉시 한 번 그리고 10일마다 보안 암호를 교체합니다. 출력에 교체 결과 생성된 새 보안 암호 버전의 `VersionId`가 표시됩니다.

```
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 Rotate secrets를 참조하세요.

예시 3: 보안 암호 즉시 교체

다음 `rotate-secret` 예시에서는 즉시 교체를 시작합니다. 출력에 교체 결과 생성된 새 보안 암호 버전의 `VersionId`가 표시됩니다. 시크릿에 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 Rotate secrets를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RotateSecret 섹션을 참조하세요.

stop-replication-to-replica

다음 코드 예시에서는 `stop-replication-to-replica` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

복제 보안 암호를 기본으로 승격

다음 `stop-replication-to-replica` 예시에서는 복제 시크릿과 기본 시크릿 간의 연결을 제거합니다. 복제 시크릿은 복제본 리전의 기본 시크릿으로 승격됩니다. 복제 리전 내에서 `stop-replication-to-replica`를 직접적으로 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 Promote a replica secret을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StopReplicationToReplica 섹션을 참조하세요.

tag-resource

다음 코드 예시에서는 `tag-resource` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

예시 1: 보안 암호에 태그 추가

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 Tag your secrets를 참조하세요.

예시 2: 보안 암호에 여러 태그 추가

다음 `tag-resource` 예시에서는 두 개의 키와 값 태그를 시크릿에 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 Tag secrets를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

`untag-resource`

다음 코드 예시에서는 `untag-resource` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

보안 암호에서 태그 제거

다음 `untag-resource` 예시에서는 시크릿에서 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 Tag secrets를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

update-secret-version-stage

다음 코드 예시에서는 update-secret-version-stage 코드를 사용하는 방법을 보여줍니다.

AWS CLI

예시 1: 이전 버전으로 보안 암호 되돌리기

다음 AWS 예시에서는 update-secret-version-stageCURRENT 스테이징 레이블을 이전 버전의 보안 암호로 이동합니다. 이를 통해 보안 암호를 이전 버전으로 되돌립니다. 이전 버전의 ID를 찾으려면 list-secret-version-ids를 사용합니다. 이 예시에서 AWSCURRENT 레이블이 있는 버전은 a1b2c3d4-5678-90ab-cdef-EXAMPLE11111이고, AWSPREVIOUS 레이블이 있는 버전은 a1b2c3d4-5678-90ab-cdef-EXAMPLE22222입니다. 이 예시에서는 AWSCURRENT 레이블을 버전 11111에서 22222로 이동합니다. AWSCURRENT 레이블이 버전에서 제거되므로, 는 update-secret-version-stagePREVIOUS 레이블을 해당 버전(11111)으로 자동으로 이동합니다. 그 결과, AWSCURRENT 및 AWSPREVIOUS 버전이 서로 교체됩니다.

```
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 Version을 참조하세요.

예시 2: 보안 암호 버전에 연결된 스테이징 레이블 추가

다음 update-secret-version-stage 예시에서는 보안 암호의 버전에 스테이징 레이블을 추가합니다. list-secret-version-ids를 실행하고 영향을 받는 버전에 대한 VersionStages 응답 필드를 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 Version을 참조하세요.

예시 3: 보안 암호 버전에 연결된 스테이징 레이블 삭제

다음 `update-secret-version-stage` 예시에서는 보안 암호 버전에 연결된 스테이징 레이블을 삭제합니다. `list-secret-version-ids`를 실행하고 영향을 받는 버전에 대한 `VersionStages` 응답 필드를 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 Version을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateSecretVersionStage 섹션을 참조하세요.

`update-secret`

다음 코드 예시에서는 `update-secret` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

예시 1: 시크릿의 설명 업데이트

다음 `update-secret` 예시에서는 시크릿의 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 사용자 안내서의 시크릿 수정을 참조하세요.

예시 2: 시크릿에 연결된 암호화 키 업데이트

다음 `update-secret` 예시에서는 시크릿 값을 암호화하는 데 사용되는 KMS 키를 업데이트합니다. KMS 키는 시크릿과 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 사용자 안내서의 시크릿 수정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateSecret을 참조하세요.

`validate-resource-policy`

다음 코드 예시에서는 `validate-resource-policy` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

리소스 정책 검증

다음 `validate-resource-policy` 예시에서는 리소스 정책이 보안 암호에 대한 광범위한 액세스 권한을 부여하지 않는지 확인합니다. 정책은 디스크의 파일에서 읽습니다. 자세한 내용은 AWS CLI 사용자 안내서의 [파일에서 AWS CLI 파라미터 로드](https://docs.aws.amazon.com/cli/latest/userguide/cli-usage-parameters-file.html)를 참조하세요.

```
aws secretsmanager validate-resource-policy \
    --resource-policy file://mypolicy.json
```

`mypolicy.json`의 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 [Permissions reference for Secrets Manager](https://docs.aws.amazon.com/secretsmanager/latest/userguide/reference_iam-permissions.html)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ValidateResourcePolicy](https://awscli.amazonaws.com/v2/documentation/api/latest/reference/secretsmanager/validate-resource-policy.html) 섹션을 참조하세요.

AWS CLI를 사용한 Security Hub 예시

다음 코드 예시에서는 Security Hub에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

accept-administrator-invitation

다음 코드 예시에서는 accept-administrator-invitation의 사용 방법을 보여줍니다.

AWS CLI

관리자 계정에서 초대를 수락하는 방법

다음 accept-administrator-invitation 예시에서는 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 관리 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AcceptAdministratorInvitation 섹션을 참조하세요.

accept-invitation

다음 코드 예시에서는 accept-invitation의 사용 방법을 보여줍니다.

AWS CLI

관리자 계정에서 초대를 수락하는 방법

다음 accept-invitation 예시에서는 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 관리 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AcceptInvitation 섹션을 참조하세요.

batch-delete-automation-rules

다음 코드 예시에서는 batch-delete-automation-rules의 사용 방법을 보여줍니다.

AWS CLI

자동화 규칙을 삭제하는 방법

다음 batch-delete-automation-rules 예시에서는 지정된 자동화 규칙을 삭제합니다. 한 번의 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 삭제](#) 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [BatchDeleteAutomationRules](#) 섹션을 참조하세요.

batch-disable-standards

다음 코드 예시에서는 batch-disable-standards의 사용 방법을 보여줍니다.

AWS CLI

표준 비활성화

다음 batch-disable-standards 예시에서는 지정된 구독 ARN과 연결된 표준을 비활성화합니다.

```
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 활성화](#) 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [BatchDisableStandards](#) 섹션을 참조하세요.

batch-enable-standards

다음 코드 예시에서는 batch-enable-standards의 사용 방법을 보여줍니다.

AWS CLI

표준 활성화

다음 batch-enable-standards 예시에서는 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 활성화 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 BatchEnableStandards 섹션을 참조하세요.

batch-get-automation-rules

다음 코드 예시에서는 batch-get-automation-rules의 사용 방법을 보여줍니다.

AWS CLI

자동화 규칙에 대한 세부 정보를 가져오는 방법

다음 batch-get-automation-rules 예시에서는 지정된 자동화 규칙에 대한 세부 정보를 가져옵니다. 명령 한 번으로 하나 이상의 자동화 규칙에 대한 세부 정보를 얻을 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 보기 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 BatchGetAutomationRules 섹션을 참조하세요.

batch-get-configuration-policy-associations

다음 코드 예시에서는 batch-get-configuration-policy-associations의 사용 방법을 보여 줍니다.

AWS CLI

대상 배치에 대한 구성 연결 세부 정보를 가져오는 방법

다음 batch-get-configuration-policy-associations 예시에서는 지정된 대상에 대한 연결 세부 정보를 검색합니다. 계정 ID, 조직 단위 ID 또는 대상의 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 AWS Security Hub 사용 설명서의 Security Hub 구성 정책 보기 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 BatchGetConfigurationPolicyAssociations 섹션을 참조하세요.

batch-get-security-controls

다음 코드 예시에서는 `batch-get-security-controls`의 사용 방법을 보여줍니다.

AWS CLI

보안 제어 세부 정보를 가져오는 방법

다음 `batch-get-security-controls` 예시에서는 현재 AWS 계정 및 AWS 리전의 보안 제어 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 제어에 대한 세부 정보 보기 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 BatchGetSecurityControls 섹션을 참조하세요.

batch-get-standards-control-associations

다음 코드 예시에서는 batch-get-standards-control-associations의 사용 방법을 보여줍니다.

AWS CLI

제어의 활성화 상태 가져오기

다음 batch-get-standards-control-associations 예시에서는 지정된 제어가 지정된 표준에서 활성화되어 있는지 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 비활성화 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 BatchGetStandardsControlAssociations 섹션을 참조하세요.

batch-import-findings

다음 코드 예시에서는 `batch-import-findings`의 사용 방법을 보여줍니다.

AWS CLI

조사 결과 업데이트

다음 `batch-import-findings` 예시에서는 조사 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 설명서의 BatchImportFindings를 사용하여 조사 결과 생성 및 업데이트 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 BatchImportFindings 섹션을 참조하세요.

batch-update-automation-rules

다음 코드 예시에서는 batch-update-automation-rules의 사용 방법을 보여줍니다.

AWS CLI

자동화 규칙을 업데이트하는 방법

다음 batch-update-automation-rules 예시에서는 지정된 자동화 규칙을 업데이트합니다. 한 번의 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 편집] 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [BatchUpdateAutomationRules] 섹션을 참조하세요.

batch-update-findings

다음 코드 예시에서는 batch-update-findings의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 조사 결과 업데이트

다음 batch-update-findings 예시에서는 두 개의 결과를 업데이트하여 메모를 추가하고, 심각도 레이블을 변경하고, 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 BatchUpdateFindings를 사용하여 조사 결과 업데이트 섹션을 참조하세요.

예시 2: 단축형 구문을 사용하여 조사 결과 업데이트

다음 batch-update-findings 예시에서는 두 개의 결과를 업데이트하여 메모를 추가하고, 심각도 레이블을 변경하고, 속기 구문을 사용하여 문제를 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 BatchUpdateFindings를 사용하여 조사 결과 업데이트 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 BatchUpdateFindings 섹션을 참조하세요.

batch-update-standards-control-associations

다음 코드 예시에서는 batch-update-standards-control-associations의 사용 방법을 보여줍니다.

AWS CLI

활성화된 표준에서 제어의 활성화 상태를 업데이트하는 방법

다음 batch-update-standards-control-associations 예시에서는 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 섹션 및 모든 표준에서 제어 활성화 및 비활성화 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 BatchUpdateStandardsControlAssociations 섹션을 참조하세요.

`create-action-target`

다음 코드 예시에서는 `create-action-target`의 사용 방법을 보여줍니다.

AWS CLI

사용자 지정 작업 생성

다음 `create-action-target` 예시에서는 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 Events 규칙과 연결] 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [CreateActionTarget] 섹션을 참조하세요.

`create-automation-rule`

다음 코드 예시에서는 `create-automation-rule`의 사용 방법을 보여줍니다.

AWS CLI

자동화 규칙을 생성하는 방법

다음 `create-automation-rule` 예시에서는 현재 AWS 계정 및 AWS 리전에 자동화 규칙을 생성합니다. Security Hub는 지정된 기준에 따라 조사 결과를 필터링하고 일치하는 조사 결과에 대해 조치를 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 생성 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateAutomationRule 섹션을 참조하세요.

create-configuration-policy

다음 코드 예시에서는 `create-configuration-policy`의 사용 방법을 보여줍니다.

AWS CLI

구성 정책 생성

다음 create-configuration-policy 예시에서는 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 AWS Security Hub 사용 설명서의 [Security Hub 구성 정책 생성 및 연결](#) 섹션을 참조하세요

- API 세부 정보는 AWS CLI 명령 참조의 [CreateConfigurationPolicy](#) 섹션을 참조하세요.

`create-finding-aggregator`

다음 코드 예시에서는 `create-finding-aggregator`의 사용 방법을 보여줍니다.

AWS CLI

결과 집계를 활성화하는 방법

다음 `create-finding-aggregator` 예시에서는 조사 결과 집계를 구성합니다. 미국 동부(버지니아)를 집계 영역으로 지정하는 미국 동부(버지니아)에서 운영됩니다. 지정된 리전만 연결하고 새 리전은 자동으로 연결하지 않음을 나타냅니다. 미국 서부(캘리포니아 북부)와 미국 서부(오레곤)를 연결 리전으로 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 조사 결과 집계 활성화 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateFindingAggregator 섹션을 참조하세요.

create-insight

다음 코드 예시에서는 create-insight의 사용 방법을 보여줍니다.

AWS CLI

사용자 지정 인사이트 생성

다음 create-insight 예시에서는 AWS 역할과 관련된 중요한 조사 결과를 반환하는 중요한 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 관리 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateInsight 섹션을 참조하세요.

create-members

다음 코드 예시에서는 create-members의 사용 방법을 보여줍니다.

AWS CLI

계정을 멤버 계정으로 추가하는 방법

다음 create-members 예시에서는 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 관리 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateMembers 섹션을 참조하세요.

decline-invitations

다음 코드 예시에서는 decline-invitations의 사용 방법을 보여줍니다.

AWS CLI

멤버 계정 초대 거부

다음 decline-invitations 예시에서는 지정된 관리자 계정의 멤버 계정이 되기 위한 초대를 거부합니다. 멤버 계정은 요청하는 계정입니다.

```
aws securityhub decline-invitations \
    --account-ids "123456789012"
```

출력:

```
{
    "UnprocessedAccounts": []
}
```

자세한 내용은 AWS Security Hub 사용 설명서의 관리자 및 멤버 계정 관리 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeclineInvitations 섹션을 참조하세요.

delete-action-target

다음 코드 예시에서는 `delete-action-target`의 사용 방법을 보여줍니다.

AWS CLI

사용자 지정 작업 삭제

다음 `delete-action-target` 예시에서는 지정된 ARN으로 식별된 사용자 지정 작업을 삭제합니다.

```
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 Events 규칙과 연결 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteActionTarget 섹션을 참조하세요.

delete-configuration-policy

다음 코드 예시에서는 `delete-configuration-policy`의 사용 방법을 보여줍니다.

AWS CLI

구성 정책 삭제

다음 `delete-configuration-policy` 예시에서는 지정한 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 AWS Security Hub 사용 설명서의 Security Hub 구성 정책 삭제 및 연결 해제 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteConfigurationPolicy 섹션을 참조하세요.

delete-finding-aggregator

다음 코드 예시에서는 `delete-finding-aggregator`의 사용 방법을 보여줍니다.

AWS CLI

집계 찾기를 중지하는 방법

다음 `delete-finding-aggregator` 예시에서는 조사 결과 집계를 중지합니다. 집계 영역인 미국 동부(버지니아)에서 운영됩니다.

```
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 조사 결과 집계 중지 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteFindingAggregator 섹션을 참조하세요.

delete-insight

다음 코드 예시에서는 `delete-insight`의 사용 방법을 보여줍니다.

AWS CLI

사용자 지정 인사이트 삭제

다음 delete-insight 예시에서는 지정된 ARN을 사용하여 사용자 지정 인사이트를 삭제합니다.

```
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 관리 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteInsight 섹션을 참조하세요.

delete-invitations

다음 코드 예시에서는 delete-invitations의 사용 방법을 보여줍니다.

AWS CLI

멤버 계정 초대 거부

다음 delete-invitations 예시에서는 지정된 관리자 계정의 멤버 계정이 되기 위한 초대를 삭제합니다. 멤버 계정은 요청하는 계정입니다.

```
aws securityhub delete-invitations \
    --account-ids "123456789012"
```

출력:

```
{
    "UnprocessedAccounts": []
}
```

자세한 내용은 AWS Security Hub 사용 설명서의 관리자 및 멤버 계정 관리 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteInvitations 섹션을 참조하세요.

delete-members

다음 코드 예시에서는 delete-members의 사용 방법을 보여줍니다.

AWS CLI

멤버 계정 삭제

다음 delete-members 예시에서는 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 관리 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteMembers 섹션을 참조하세요.

describe-action-targets

다음 코드 예시에서는 describe-action-targets의 사용 방법을 보여줍니다.

AWS CLI

사용자 지정 작업에 대한 세부 정보를 검색하는 방법

다음 describe-action-targets 예시에서는 지정된 ARN으로 식별된 사용자 지정 작업에 대한 정보를 검색합니다.

```
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 Events 규칙과 연결 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeActionTargets 섹션을 참조하세요.

describe-hub

다음 코드 예시에서는 describe-hub의 사용 방법을 보여줍니다.

AWS CLI

허브 리소스에 대한 정보 가져오기

다음 describe-hub 예시에서는 지정된 허브 리소스의 구독 날짜를 반환합니다. 허브 리소스는 ARN으로 식별됩니다.

```
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::Hub 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeHub 섹션을 참조하세요.

`describe-organization-configuration`

다음 코드 예시에서는 `describe-organization-configuration`의 사용 방법을 보여줍니다.

AWS CLI

조직에 대해 Security Hub가 구성된 방법을 보는 방법

다음 `describe-organization-configuration` 예시는 Security Hub에서 조직이 구성된 방식에 대한 정보를 반환합니다. 이 예시에서는 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 관리 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeOrganizationConfiguration 섹션을 참조하세요.

`describe-products`

다음 코드 예시에서는 `describe-products`의 사용 방법을 보여줍니다.

AWS CLI

사용 가능한 제품 통합에 대한 정보를 반환하는 방법

다음 `describe-products` 예시에서는 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 관리 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeProducts 섹션을 참조하세요.

`describe-standards-controls`

다음 코드 예시에서는 `describe-standards-controls`의 사용 방법을 보여줍니다.

AWS CLI

활성화된 표준에 대한 제어 목록 요청

다음 `describe-standards-controls` 예시에서는 PCI DSS 표준에 대한 요청자 계정의 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 제어에 대한 세부 정보 보기 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeStandardsControls 섹션을 참조하세요.

describe-standards

다음 코드 예시에서는 `describe-standards`의 사용 방법을 보여줍니다.

AWS CLI

사용 가능한 표준 목록을 반환하는 방법

다음 `describe-standards` 예시에서는 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 AWS Security Hub 사용 설명서의 [AWS Security Hub의 보안 표준](#) 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeStandards 섹션을 참조하세요.

disable-import-findings-for-product

다음 코드 예시에서는 disable-import-findings-for-product의 사용 방법을 보여줍니다.

AWS CLI

제품 통합에서 결과 수신을 중지하는 방법

다음 disable-import-findings-for-product 예시에서는 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 관리 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisableImportFindingsForProduct 섹션을 참조하세요.

disable-organization-admin-account

다음 코드 예시에서는 disable-organization-admin-account의 사용 방법을 보여줍니다.

AWS CLI

Security Hub 관리자 계정 제거

다음 disable-organization-admin-account 예시에서는 지정된 계정의 AWS Organization용 Security Hub 관리자 계정 할당을 취소합니다.

```
aws securityhub disable-organization-admin-account \
    --admin-account-id 777788889999
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Security Hub 사용 설명서의 Security Hub 관리자 계정 지정 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisableOrganizationAdminAccount 섹션을 참조하세요.

`disable-security-hub`

다음 코드 예시에서는 `disable-security-hub`의 사용 방법을 보여줍니다.

AWS CLI

AWS Security Hub 비활성화

다음 `disable-security-hub` 예시에서는 요청 계정의 AWS Security Hub를 비활성화합니다.

```
aws securityhub disable-security-hub
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Security Hub 사용 안내서의 AWS Security Hub 비활성화 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisableSecurityHub 섹션을 참조하세요.

`disassociate-from-administrator-account`

다음 코드 예시에서는 `disassociate-from-administrator-account`의 사용 방법을 보여줍니다.

AWS CLI

관리자 계정에서 연결 해제

다음 `disassociate-from-administrator-account` 예시에서는 요청 계정을 현재 관리자 계정에서 연결 해제합니다.

```
aws securityhub disassociate-from-administrator-account
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Security Hub 사용 설명서의 관리자 및 멤버 계정 관리 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisassociateFromAdministratorAccount 섹션을 참조하세요.

disassociate-from-master-account

다음 코드 예시에서는 `disassociate-from-master-account`의 사용 방법을 보여줍니다.

AWS CLI

관리자 계정에서 연결 해제

다음 `disassociate-from-master-account` 예시에서는 요청 계정을 현재 관리자 계정에서 연결 해제합니다.

```
aws securityhub disassociate-from-master-account
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Security Hub 사용 설명서의 관리자 및 멤버 계정 관리 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisassociateFromMasterAccount 섹션을 참조하세요.

disassociate-members

다음 코드 예시에서는 `disassociate-members`의 사용 방법을 보여줍니다.

AWS CLI

멤버 계정 연결 해제

다음 `disassociate-members` 예시에서는 요청하는 관리자 계정에서 지정된 멤버 계정을 연결 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 관리 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisassociateMembers 섹션을 참조하세요.

enable-import-findings-for-product

다음 코드 예시에서는 `enable-import-findings-for-product`의 사용 방법을 보여줍니다.

AWS CLI

제품 통합에서 결과 수신을 시작하는 방법

다음 enable-import-findings-for-product 예시에서는 지정된 제품 통합의 조사 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 관리 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 EnableImportFindingsForProduct 섹션을 참조하세요.

enable-organization-admin-account

다음 코드 예시에서는 enable-organization-admin-account의 사용 방법을 보여줍니다.

AWS CLI

조직의 계정을 Security Hub 관리자 계정으로 지정

다음 enable-organization-admin-account 예시에서는 지정한 계정을 Security Hub 관리자 계정으로 지정합니다.

```
aws securityhub enable-organization-admin-account \
    --admin-account-id 777788889999
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Security Hub 사용 설명서의 Security Hub 관리자 계정 지정 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 EnableOrganizationAdminAccount 섹션을 참조하세요.

enable-security-hub

다음 코드 예시에서는 enable-security-hub의 사용 방법을 보여줍니다.

AWS CLI

AWS Security Hub 활성화

다음 enable-security-hub 예시에서는 요청 계정에 대한 AWS Security Hub를 활성화합니다. 기본 표준을 활성화하도록 Security Hub를 구성합니다. 허브 리소스의 경우 태그 Department에 값 Security를 할당합니다.

```
aws securityhub enable-security-hub \
    --enable-default-standards \
    --tags '{"Department": "Security"}'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Security Hub 사용 안내서의 Security Hub 활성화 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 EnableSecurityHub 섹션을 참조하세요.

get-administrator-account

다음 코드 예시에서는 get-administrator-account의 사용 방법을 보여줍니다.

AWS CLI

관리자 계정에 대한 정보를 검색하는 방법

다음 get-administrator-account 예시에서는 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 관리 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetAdministratorAccount 섹션을 참조하세요.

get-configuration-policy-association

다음 코드 예시에서는 get-configuration-policy-association의 사용 방법을 보여줍니다.

AWS CLI

대상에 대한 구성 연결 세부 정보를 가져오는 방법

다음 get-configuration-policy-association 예시에서는 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 AWS Security Hub 사용 설명서의 Security Hub 구성 정책 보기 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetConfigurationPolicyAssociation 섹션을 참조하세요.

get-configuration-policy

다음 코드 예시에서는 get-configuration-policy의 사용 방법을 보여줍니다.

AWS CLI

구성 정책 세부 정보를 보는 방법

다음 get-configuration-policy 예시에서는 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 AWS Security Hub 사용 설명서의 Security Hub 구성 정책 보기 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetConfigurationPolicy 섹션을 참조하세요.

get-enabled-standards

다음 코드 예시에서는 get-enabled-standards의 사용 방법을 보여줍니다.

AWS CLI

활성화된 표준에 대한 정보를 검색하는 방법

다음 get-enabled-standards 예시에서는 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 AWS Security Hub 사용 설명서의 AWS Security Hub의 보안 표준 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetEnabledStandards 섹션을 참조하세요.

get-finding-aggregator

다음 코드 예시에서는 `get-finding-aggregator`의 사용 방법을 보여줍니다.

AWS CLI

현재 결과 집계 구성을 검색하는 방법

다음 `get-finding-aggregator` 예시에서는 현재 조사 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 조사 결과 집계 구성 보기 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetFindingAggregator 섹션을 참조하세요.

get-finding-history

다음 코드 예시에서는 `get-finding-history`의 사용 방법을 보여줍니다.

AWS CLI

조사 결과 기록을 가져오는 방법

다음 `get-finding-history` 예시는 지정된 조사 결과에 대한 최근 90일간의 기록을 가져옵니다. 이 예시에서는 결과가 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 기록 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetFindingHistory 섹션을 참조하세요.

get-findings

다음 코드 예시에서는 `get-findings`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 특정 표준에 대해 생성된 조사 결과 반환

다음 get-findings 예시에서는 PCI DSS 표준에 대한 조사 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=="
}
```


예시 2: 워크플로 상태가 NOTIFIED인 중요 심각도 조사 결과 반환

다음 `get-findings` 예시에서는 심각도 레이블 값이 CRITICAL이고 워크플로 상태가 NOTIFIED 인 조사 결과를 반환합니다. 결과는 신뢰도 값에 따라 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 조사 결과 필터링 및 그룹화 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetFindings 섹션을 참조하세요.

get-insight-results

다음 코드 예시에서는 `get-insight-results`의 사용 방법을 보여줍니다.

AWS CLI

인사이트에 대한 결과를 검색하는 방법

다음 `get-insight-results` 예시에서는 지정된 ARN을 가진 인사이트에 대한 인사이트 결과 목록을 반환합니다.

```
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 조사 결과 보기 및 조치 실행 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetInsightResults 섹션을 참조하세요.

get-insights

다음 코드 예시에서는 get-insights의 사용 방법을 보여줍니다.

AWS CLI

인사이트에 대한 세부 정보를 검색하는 방법

다음 get-insights 예시에서는 지정된 ARN을 사용하여 인사이트에 대한 구성 세부 정보를 검색합니다.

```
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 AWS Security Hub의 인사이트 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetInsights 섹션을 참조하세요.

get-invitations-count

다음 코드 예시에서는 `get-invitations-count`의 사용 방법을 보여줍니다.

AWS CLI

수락되지 않은 초대 수를 검색하는 방법

다음 get-invitations-count 예시에서는 요청하는 계정이 초대를 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 관리 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetInvitationsCount 섹션을 참조하세요.

get-master-account

다음 코드 예시에서는 get-master-account의 사용 방법을 보여줍니다.

AWS CLI

관리자 계정에 대한 정보를 검색하는 방법

다음 get-master-account 예시에서는 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 관리 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [GetMasterAccount](#) 섹션을 참조하세요.

get-members

다음 코드 예시에서는 get-members의 사용 방법을 보여줍니다.

AWS CLI

선택한 멤버 계정에 대한 정보 검색

다음 get-members 예시에서는 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 관리 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetMembers 섹션을 참조하세요.

get-security-control-definition

다음 코드 예시에서는 `get-security-control-definition`의 사용 방법을 보여줍니다.

AWS CLI

보안 제어 정의 세부 정보를 가져오는 방법

다음 `get-security-control-definition` 예시에서는 Security Hub 보안 제어에 대한 정의 세부 정보를 검색합니다. 세부 정보에는 제어 제목, 설명, 리전 사용 가능성, 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 파라미터 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetSecurityControlDefinition 섹션을 참조하세요.

invite-members

다음 코드 예시에서는 invite-members의 사용 방법을 보여줍니다.

AWS CLI

멤버 계정 초대장 보내기

다음 invite-members 예시에서는 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 관리 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 InviteMembers 섹션을 참조하세요.

list-automation-rules

다음 코드 예시에서는 list-automation-rules의 사용 방법을 보여줍니다.

AWS CLI

자동화 규칙 목록을 보는 방법

다음 `list-automation-rules` 예시에서는 AWS 계정에 대한 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 보기 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListAutomationRules 섹션을 참조하세요.

list-configuration-policies

다음 코드 예시에서는 `list-configuration-policies`의 사용 방법을 보여줍니다.

AWS CLI

구성 정책 요약을 나열하는 방법

다음 `list-configuration-policies` 예시에서는 조직의 구성 정책 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 AWS Security Hub 사용 설명서의 Security Hub 구성 정책 보기 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListConfigurationPolicies 섹션을 참조하세요.

`list-configuration-policy-associations`

다음 코드 예시에서는 `list-configuration-policy-associations`의 사용 방법을 보여줍니다.

AWS CLI

구성 연결을 나열하는 방법

다음 `list-configuration-policy-associations` 예시에서는 조직의 구성 연결 요약을 나열합니다. 응답에는 구성 정책 및 자체 관리 동작과의 연결이 포함됩니다.

```
aws securityhub list-configuration-policy-associations \
    --filters '{"AssociationType": "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 AWS Security Hub 사용 설명서의 구성 정책 상태 및 세부 정보 보기 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListConfigurationPolicyAssociations 섹션을 참조하세요.

list-enabled-products-for-import

다음 코드 예시에서는 `list-enabled-products-for-import`의 사용 방법을 보여줍니다.

AWS CLI

활성화된 제품 통합 목록을 반환하는 방법

다음 `list-enabled-products-for-import` 예시에서는 현재 활성화된 제품 통합에 대한 구독 ARNS 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 관리 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListEnabledProductsForImport 섹션을 참조하세요.

`list-finding-aggregators`

다음 코드 예시에서는 `list-finding-aggregators`의 사용 방법을 보여줍니다.

AWS CLI

사용 가능한 위젯을 나열하는 방법

다음 `list-finding-aggregators` 예시에서는 조사 결과 집계 구성의 ARN을 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 조사 결과 집계 구성 보기 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListFindingAggregators 섹션을 참조하세요.

list-invitations

다음 코드 예시에서는 `list-invitations`의 사용 방법을 보여줍니다.

AWS CLI

초대 목록을 표시하는 방법

다음 `list-invitations` 예시에서는 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 관리 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListInvitations 섹션을 참조하세요.

list-members

다음 코드 예시에서는 `list-members`의 사용 방법을 보여줍니다.

AWS CLI

멤버 계정 목록 검색

다음 `list-members` 예시에서는 요청하는 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 관리 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListMembers 섹션을 참조하세요.

`list-organization-admin-accounts`

다음 코드 예시에서는 `list-organization-admin-accounts`의 사용 방법을 보여줍니다.

AWS CLI

지정된 Security Hub 관리자 계정 나열

다음 `list-organization-admin-accounts` 예시에서는 조직의 Security Hub 관리자 계정입니다.

```
aws securityhub list-organization-admin-accounts
```

출력:

```
{
```

```
    AdminAccounts": [
        { "AccountId": "777788889999" },
        { "Status": "ENABLED" }
    ]
}
```

자세한 내용은 AWS Security Hub 사용 설명서의 Security Hub 관리자 계정 지정 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListOrganizationAdminAccounts 섹션을 참조하세요.

list-security-control-definitions

다음 코드 예시에서는 `list-security-control-definitions`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 사용 가능한 모든 보안 제어를 나열

다음 `list-security-control-definitions` 예시에서는 모든 Security Hub 표준에서 사용 가능한 보안 제어를 나열합니다. 이 예시에서는 결과를 세 가지 제어로 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 보기](#) 섹션을 참조하세요.

예시 2: 특정 표준에 사용 가능한 보안 제어 나열

다음 `list-security-control-definitions` 예시에서는 CIS AWS Foundation Benchmark v1.4.0에 사용 가능한 보안 제어를 나열합니다. 이 예시에서는 결과를 세 가지 제어로 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 보기](#) 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ListSecurityControlDefinitions](#) 섹션을 참조하세요.

`list-standards-control-associations`

다음 코드 예시에서는 `list-standards-control-associations`의 사용 방법을 보여줍니다.

AWS CLI

활성화된 각 표준에서 제어의 활성화 상태를 가져오는 방법

다음 `list-standards-control-associations` 예시에서는 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 비활성화 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListStandardsControlAssociations 섹션을 참조하세요.

`list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`의 사용 방법을 보여줍니다.

AWS CLI

리소스에 할당된 태그를 검색하는 방법

다음 `list-tags-for-resource` 예시에서는 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::Hub 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource를 참조하세요.

`start-configuration-policy-association`

다음 코드 예시에서는 `start-configuration-policy-association`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 구성 정책 연결

다음 `start-configuration-policy-association` 예시에서는 지정된 구성 정책을 지정된 조직 단위와 연결합니다. 구성은 대상 계정, 조직 단위 또는 루트에 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 AWS Security Hub 사용 설명서의 Security Hub 구성 정책 생성 및 연결 섹션을 참조하세요

예시 2: 자체 관리형 구성 연결

다음 `start-configuration-policy-association` 예시에서는 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 AWS Security Hub 사용 설명서의 Security Hub 구성 정책 생성 및 연결 섹션을 참조하세요

- API 세부 정보는 AWS CLI 명령 참조의 StartConfigurationPolicyAssociation 섹션을 참조하세요.

`start-configuration-policy-disassociation`

다음 코드 예시에서는 `start-configuration-policy-disassociation`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 구성 정책의 연결을 해제

다음 `start-configuration-policy-disassociation` 예시에서는 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 OU에서 구성 연결 해제 섹션을 참조하세요.

예시 2: 자체 관리형 구성의 연결 해제

다음 `start-configuration-policy-disassociation` 예시에서는 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 OU에서 구성 연결 해제 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartConfigurationPolicyDisassociation 섹션을 참조하세요.

`tag-resource`

다음 코드 예시에서는 `tag-resource`의 사용 방법을 보여줍니다.

AWS CLI

리소스에 태그 할당

다음 `tag-resource` 예시에서는 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::Hub 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

untag-resource

다음 코드 예제에서는 `untag-resource`의 사용 방법을 보여줍니다.

AWS CLI

리소스에서 태그 값 제거

다음 `untag-resource` 예시에서는 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::Hub 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

update-action-target

다음 코드 예시에서는 `update-action-target`의 사용 방법을 보여줍니다.

AWS CLI

사용자 지정 작업을 업데이트하는 방법

다음 `update-action-target` 예시에서는 지정된 ARN으로 식별된 사용자 지정 작업의 이름을 업데이트합니다.

```
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 Events 규칙과 연결 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateActionTarget 섹션을 참조하세요.

update-configuration-policy

다음 코드 예시에서는 update-configuration-policy의 사용 방법을 보여줍니다.

AWS CLI

구성 정책 업데이트

다음 update-configuration-policy 예시에서는 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 AWS Security Hub 사용 안내서의 Security Hub 구성 정책 업데이트 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateConfigurationPolicy 섹션을 참조하세요.

update-finding-aggregator

다음 코드 예시에서는 update-finding-aggregator의 사용 방법을 보여줍니다.

AWS CLI

현재 결과 집계 구성을 업데이트하는 방법

다음 update-finding-aggregator 예시에서는 조사 결과 집계 구성을 선택한 리전에서 연결하도록 변경합니다. 집계 영역인 미국 동부(버지니아)에서 운영됩니다. 미국 서부(캘리포니아 북부)와 미국 서부(오레곤)를 연결 리전으로 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 안내서의 조사 결과 집계 구성 업데이트 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateFindingAggregator 섹션을 참조하세요.

update-insight

다음 코드 예시에서는 update-insight의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 사용자 지정 인사이트의 필터 변경

다음 update-insight 예시에서는 사용자 지정 인사이트의 필터를 변경합니다. 업데이트된 인사이트는 AWS 역할과 관련된 심각도가 높은 조사 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"
```

예시 2: 사용자 지정 인사이트의 그룹화 속성 변경

다음 `update-insight` 예시에서는 지정된 ARN으로 사용자 지정 인사이트의 그룹화 속성을 변경합니다. 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 관리 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateInsight 섹션을 참조하세요.

update-organization-configuration

다음 코드 예시에서는 update-organization-configuration의 사용 방법을 보여줍니다.

AWS CLI

조직에 대해 Security Hub를 구성하는 방법을 업데이트하는 방법

다음 update-organization-configuration 예시에서는 Security Hub가 중앙 구성을 사용하여 조직을 구성하도록 지정합니다. 이 명령을 실행한 후 위임받은 Security Hub 관리자는 조직을 구성하기 위한 구성 정책을 만들고 관리할 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 사용 설정할지 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 Organizations를 사용하여 계정 관리 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateOrganizationConfiguration 섹션을 참조하세요.

update-security-control

다음 코드 예시에서는 update-security-control의 사용 방법을 보여줍니다.

AWS CLI

보안 제어 속성을 업데이트하는 방법

다음 update-security-control 예시에서는 Security Hub 보안 제어 파라미터에 대한 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 파라미터 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateSecurityControl 섹션을 참조하세요.

update-security-hub-configuration

다음 코드 예시에서는 `update-security-hub-configuration`의 사용 방법을 보여줍니다.

AWS CLI

Security Hub 구성 업데이트

다음 `update-security-hub-configuration` 예시에서는 활성화된 표준에 대해 새로운 제어 기능을 자동으로 사용하도록 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 제어 자동 활성화 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateSecurityHubConfiguration 섹션을 참조하세요.

update-standards-control

다음 코드 예시에서는 `update-standards-control`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 제어 비활성화

다음 `update-standards-control` 예시에서는 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.

예시 2: 제어 활성화

다음 `update-standards-control` 예시에서는 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 활성화 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateStandardsControl 섹션을 참조하세요.

AWS CLI를 사용한 Security Lake 예시

다음 코드 예시에서는 Security Lake에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`create-aws-log-source`

다음 코드 예시는 `create-aws-log-source`의 사용 방법을 보여 줍니다.

AWS CLI

기본적으로 지원되는 Amazon Web Service를 Amazon Security Lake 소스로 추가하는 방법

다음 `create-aws-logsource` 예시에서는 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 Adding an AWS service as a source를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의CreateAwsLogSource 섹션을 참조하세요.

`create-custom-log-source`

다음 코드 예시는 `create-custom-log-source`의 사용 방법을 보여 줍니다.

AWS CLI

사용자 지정 소스를 Amazon Security Lake 소스로 추가하는 방법

다음 `create-custom-logsource` 예시에서는 사용자 지정 소스를 지정된 로그 공급자 계정과 지정된 리전에 보안 레이크 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": "amzn-s3-demo-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 Adding a custom source를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateCustomLogSource를 참조하세요.

`create-data-lake-exception-subscription`

다음 코드 예시에서는 `create-data-lake-exception-subscription` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Security Lake 예외 알림을 보내려면

다음 `create-data-lake-exception-subscription` 예시에서는 SMS 전송을 통해 지정된 계정으로 보안 레이크 예외 알림을 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 Troubleshooting Amazon Security Lake을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateDataLakeExceptionSubscription을 참조하세요.

`create-data-lake-organization-configuration`

다음 코드 예시에서는 `create-data-lake-organization-configuration` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

새 조직 계정에서 Security Lake를 구성하는 방법

다음 `create-data-lake-organization-configuration` 예시에서는 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 Managing multiple accounts with AWS Organizations를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateDataLakeOrganizationConfiguration을 참조하세요.

`create-data-lake`

다음 코드 예시에서는 `create-data-lake` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

예시 1: 여러 리전에서 데이터 레이크를 구성하는 방법

다음 `create-data-lake` 예시에서는 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 시작하기]를 참조하세요.

예시 2: 단일 리전에서 데이터 레이크를 구성하는 방법

다음 `create-data-lake` 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateDataLake를 참조하세요.

create-subscriber-notification

다음 코드 예시는 create-subscriber-notification의 사용 방법을 보여 줍니다.

AWS CLI

구독자 알림을 생성하는 방법

다음 create-subscriber-notification 예시에서는 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 Subscriber management를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateSubscriberNotification를 참조하세요.

create-subscriber

다음 코드 예시는 `create-subscriber`의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 데이터에 액세스할 수 있는 구독자 생성

다음 `create-subscriber` 예시에서는 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 [Creating a subscriber with data access]를 참조하세요.

예제 2: 쿼리에 액세스할 수 있는 구독자 생성

다음 `create-subscriber` 예시에서는 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 Creating a subscriber with query access를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateSubscriber를 참조하세요.

delete-aws-log-source

다음 코드 예시는 delete-aws-log-source의 사용 방법을 보여 줍니다.

AWS CLI

기본적으로 지원되는 AWS 서비스를 제거합니다.

다음 delete-aws-logsource 예시에서는 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 Removing an AWS service as a source를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteAwsLogSource를 참조하세요.

delete-custom-log-source

다음 코드 예시는 delete-custom-log-source의 사용 방법을 보여 줍니다.

AWS CLI

사용자 지정 소스를 제거합니다.

다음 delete-custom-logsource 예시에서는 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 Deleting a custom source를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteCustomLogSource를 참조하세요.

delete-data-lake-organization-configuration

다음 코드 예시에서는 delete-data-lake-organization-configuration 코드를 사용하는 방법을 보여줍니다.

AWS CLI

멤버 계정에서 자동 소스 수집을 중지하는 방법

다음 delete-data-lake-organization-configuration 예시에서는 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 Managing multiple accounts with AWS Organizations를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteDataLakeOrganizationConfiguration를 참조하세요.

`delete-data-lake`

다음 코드 예시에서는 `delete-data-lake` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

데이터 레이크를 비활성화하는 방법

다음 `delete-data-lake` 예시에서는 지정된 AWS 리전에서 데이터 레이크를 비활성화합니다. 지정된 리전에서 소스는 더 이상 데이터 레이크에 데이터를 제공하지 않습니다. AWS Organizations를 사용하는 Security Lake 배포의 경우, 조직에 대해 위임된 Security Lake 관리자만 조직 내 계정에 대해 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 Disabling Amazon Security Lake를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteDataLake를 참조하세요.

`delete-subscriber-notification`

다음 코드 예시에서는 `delete-subscriber-notification` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

구독자 알림을 삭제하는 방법

다음 `delete-subscriber-notification` 예시에서는 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 Subscriber management를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteSubscriberNotification를 참조하세요.

delete-subscriber

다음 코드 예시에서는 `delete-subscriber` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

구독자를 삭제하는 방법

다음 `delete-subscriber` 예시에서는 구독자가 더 이상 Security Lake에서 데이터를 소비하지 않도록 하려는 경우 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 Subscriber management를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteSubscriber를 참조하세요.

get-data-lake-exception-subscription

다음 코드 예시에서는 `get-data-lake-exception-subscription` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

예외 구독에 대한 세부 정보를 가져오는 방법

다음 `get-data-lake-exception-subscription` 예시에서는 Security Lake 예외 구독에 대한 세부 정보를 제공합니다. 이 예시에서는 지정된 AWS 계정의 사용자에게 SMS 전송을 통해 오류 알림을 보냅니다. 예외 메시지는 지정된 기간 동안 계정에 남아 있습니다. 예외 구독은 요청자가 선호하는 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 Troubleshooting data lake status을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetDataLakeExceptionSubscription을 참조하세요.

get-data-lake-organization-configuration

다음 코드 예시에서는 `get-data-lake-organization-configuration` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

새 조직 계정의 구성에 대한 세부 정보를 가져오는 방법

다음 `get-data-lake-organization-configuration` 예시에서는 새 조직 계정이 Amazon Security Lake에 온보딩한 후 전송하는 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 Managing multiple accounts with AWS Organizations를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetDataLakeOrganizationConfiguration를 참조하세요.

get-data-lake-sources

다음 코드 예시에서는 get-data-lake-sources 코드를 사용하는 방법을 보여줍니다.

AWS CLI

로그 수집 상태를 가져오는 방법

다음 get-data-lake-sources 예시에서는 현재 AWS 리전에서 지정된 계정에 대한 로그 수집의 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 Collecting data from AWS services를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetDataLakeSources를 참조하세요.

get-subscriber

다음 코드 예시에서는 get-subscriber 코드를 사용하는 방법을 보여줍니다.

AWS CLI

구독 정보를 검색하는 방법

다음 get-subscriber 예시에서는 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 Subscriber management를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetSubscriber을 참조하세요.

`list-data-lake-exceptions`

다음 코드 예시에서는 `list-data-lake-exceptions` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

데이터 레이크에 영향을 미치는 문제를 나열하는 방법

다음 `list-data-lake-exceptions` 예시에서는 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 Troubleshooting Amazon Security Lake을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListDataLakeExceptions를 참조하세요.

`list-data-lakes`

다음 코드 예시에서는 `list-data-lakes` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Security Lake 구성 객체를 나열하는 방법

다음 `list-data-lakes` 예시에서는 지정된 AWS 리전에 대한 Amazon Security Lake 구성 객체를 나열합니다. 이 명령을 사용하여 지정된 리전에서 Security Lake가 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 Checking Region status를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListDataLakes를 참조하세요.

list-log-sources

다음 코드 예시에서는 `list-log-sources` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Amazon Security Lake 로그 소스를 검색하는 방법

다음 `list-log-sources` 예시에서는 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 Source management를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListLogSources를 참조하세요.

list-subscribers

다음 코드 예시에서는 `list-subscribers` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Amazon Security Lake 구독자를 검색하는 방법

다음 `list-subscribers` 예시에서는 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": "amzn-s3-demo-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": "amzn-s3-demo-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 Subscriber management를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListSubscribers 섹션을 참조하세요.

list-tags-for-resource

다음 코드 예시에서는 `list-tags-for-resource` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

기존 리소스의 태그 나열

다음 `list-tags-for-resource` 예시에서는 지정된 Amazon Security Lake 구독자의 태그를 나열합니다. 이 예시에서는 Owner 태그 키에 연결된 태그 값이 없습니다. 이 작업을 사용하여 기존의 다른 Security Lake 리소스에 대한 태그도 나열할 수 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 리소스에 태그 지정](#)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ListTagsForResource](#)를 참조하세요.

`register-data-lake-delegated-administrator`

다음 코드 예시에서는 `register-data-lake-delegated-administrator` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

위임된 관리자 지정

다음 `register-data-lake-delegated-administrator` 예시에서는 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 [Managing multiple accounts with AWS Organizations](https://docs.aws.amazon.com/)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RegisterDataLakeDelegatedAdministrator를 참조하세요.

`tag-resource`

다음 코드 예시에서는 `tag-resource` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

기존 리소스에 태그 추가

다음 `tag-resource` 예시에서는 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 리소스에 태그 지정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

untag-resource

다음 코드 예시에서는 untag-resource 코드를 사용하는 방법을 보여줍니다.

AWS CLI

기존 리소스에서 태그를 제거하는 방법

다음 untag-resource 예시에서는 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 리소스에 태그 지정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

update-data-lake-exception-subscription

다음 코드 예시에서는 update-data-lake-exception-subscription 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Security Lake 예외에 대한 알림 구독을 업데이트하는 방법

다음 update-data-lake-exception-subscription 예시에서는 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 Troubleshooting Amazon Security Lake을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateDataLakeExceptionSubscription를 참조하세요.

update-data-lake

다음 코드 예시에서는 update-data-lake 코드를 사용하는 방법을 보여줍니다.

AWS CLI

예시 1: 데이터 레이크 설정 업데이트

다음 update-data-lake 예시에서는 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 시작하기]를 참조하세요.

예시 2: 단일 리전에서 데이터 레이크를 구성하는 방법

다음 `create-data-lake` 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateDataLake를 참조하세요.

`update-subscriber-notification`

다음 코드 예시에서는 `update-subscriber-notification` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

구독자 알림을 업데이트하는 방법

다음 `update-subscriber-notification` 예시에서는 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 Subscriber management를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateSubscriberNotification을 참조하세요.

update-subscriber

다음 코드 예시에서는 `update-subscriber` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Amazon Security Lake 구독자를 업데이트하는 방법

다음 `update-subscriber` 예시에서는 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": "amzn-s3-demo-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 Subscriber management를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateSubscriber를 참조하세요.

AWS CLI를 사용한 AWS Serverless Application Repository 예시

다음 코드 예시에서는 AWS Serverless Application Repository에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

put-application-policy

다음 코드 예시에서는 put-application-policy의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 애플리케이션을 공개적으로 공유

다음 put-application-policy는 공개적으로 애플리케이션을 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"
        }
    ]
}
```

예시 2: 애플리케이션을 비공개로 공유

다음 put-application-policy는 애플리케이션을 비공개로 공유하므로 특정 AWS 계정만 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=111111111111,222222222222,Actions=Deploy
```

출력:

```
{
    "Statements": [
        {
            "Actions": [
                "Deploy"
            ],
            "Principals": [
                "111111111111",
                "222222222222"
            ],
            "StatementId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
        }
    ]
}
```

자세한 내용은 AWS Serverless Application Repository 개발자 안내서의 콘솔을 통한 애플리케이션 공유를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutApplicationPolicy를 참조하세요.

AWS CLI를 사용한 Service Catalog 예시

다음 코드 예시는 Service Catalog와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

accept-portfolio-share

다음 코드 예시에서는 accept-portfolio-share의 사용 방법을 보여줍니다.

AWS CLI

포트폴리오 공유 수락

다음 accept-portfolio-share 예시에서는 지정된 포트폴리오를 공유하는 다른 사용자의 제안을 수락합니다.

```
aws servicecatalog accept-portfolio-share \
    --portfolio-id port-2s6wuabcdefghijk
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 AcceptPortfolioShare를 참조하세요.

associate-principal-with-portfolio

다음 코드 예시에서는 associate-principal-with-portfolio의 사용 방법을 보여줍니다.

AWS CLI

포트폴리오에 위탁자 연결

다음 associate-principal-with-portfolio 예시에서는 사용자를 지정된 포트폴리오에 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.

- API 세부 정보는 AWS CLI 명령 참조의 AssociatePrincipalWithPortfolio를 참조하세요.

associate-product-with-portfolio

다음 코드 예시에서는 associate-product-with-portfolio의 사용 방법을 보여줍니다.

AWS CLI

포트폴리오에 제품 연결

다음 `associate-product-with-portfolio` 예시에서는 지정된 제품을 지정된 포트폴리오에 연결합니다.

```
aws servicecatalog associate-product-with-portfolio
     --product-id prod-3p5abcdef3oyk
     --portfolio-id port-2s6abcdef5wdh4
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 [AssociateProductWithPortfolio](https://docs.aws.amazon.com/goto/aws-cli/servicecatalog-2015-12-10/AssociateProductWithPortfolio)를 참조하세요.

`associate-tag-option-with-resource`

다음 코드 예시에서는 `associate-tag-option-with-resource`의 사용 방법을 보여줍니다.

AWS CLI

리소스에 TagOption 연결

다음 `associate-tag-option-with-resource` 예시에서는 지정된 TagOption을 지정된 리소스에 연결합니다.

```
aws servicecatalog associate-tag-option-with-resource \
    --resource-id port-2s6abcdq5wdh4 \
    --tag-option-id tag-p3abc2pkpz5qc
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 [AssociateTagOptionWithResource](https://docs.aws.amazon.com/goto/aws-cli/servicecatalog-2015-12-10/AssociateTagOptionWithResource)를 참조하세요.

`copy-product`

다음 코드 예시에서는 `copy-product`의 사용 방법을 보여줍니다.

AWS CLI

제품 복사

다음 copy-product 예시에서는 JSON 파일을 사용해 파라미터를 전달하여 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CopyProduct를 참조하세요.

create-portfolio-share

다음 코드 예시에서는 create-portfolio-share의 사용 방법을 보여줍니다.

AWS CLI

계정과 포트폴리오 공유

다음 create-portfolio-share 예시에서는 지정된 포트폴리오를 지정된 계정과 공유합니다.

```
aws servicecatalog create-portfolio-share \
    --portfolio-id port-2s6abcdef5wdh4 \
    --account-id 794123456789
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 [CreatePortfolioShare](#)를 참조하세요.

create-portfolio

다음 코드 예시에서는 create-portfolio의 사용 방법을 보여줍니다.

AWS CLI

포트폴리오 생성

다음 create-portfolio 예시에서는 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [CreatePortfolio](#)를 참조하세요.

create-product

다음 코드 예시에서는 create-product의 사용 방법을 보여줍니다.

AWS CLI

제품 생성

다음 create-product 예시에서는 JSON 파일을 사용해 파라미터를 전달하여 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateProduct를 참조하세요.

create-provisioning-artifact

다음 코드 예시에서는 `create-provisioning-artifact`의 사용 방법을 보여줍니다.

AWS CLI

프로비저닝 아티팩트 생성

다음 `create-provisioning-artifact` 예시에서는 JSON 파일을 사용해 파라미터를 전달하여 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
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateProvisioningArtifact를 참조하세요.

`create-tag-option`

다음 코드 예시에서는 `create-tag-option`의 사용 방법을 보여줍니다.

AWS CLI

TagOption 생성

다음 `create-tag-option` 예시에서는 TagOption을 생성합니다.

```
aws servicecatalog create-tag-option
    --key 1234
    --value name
```

출력:

```
{
    "TagOptionDetail": {
    "Id": "tag-iabcdn4fzjjms",
    "Value": "name",
    "Active": true,
    "Key": "1234"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateTagOption을 참조하세요.

`delete-portfolio-share`

다음 코드 예시에서는 `delete-portfolio-share`의 사용 방법을 보여줍니다.

AWS CLI

계정과의 포트폴리오 공유 중지

다음 `delete-portfolio-share` 예시에서는 지정된 계정과의 포트폴리오 공유를 중지합니다.

```
aws servicecatalog delete-portfolio-share \
    --portfolio-id port-2s6abcdq5wdh4 \
    --account-id 123456789012
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 DeletePortfolioShare를 참조하세요.

delete-portfolio

다음 코드 예시에서는 delete-portfolio의 사용 방법을 보여줍니다.

AWS CLI

포트폴리오 삭제

다음 delete-portfolio 예시에서는 지정된 포트폴리오를 삭제합니다.

```
aws servicecatalog delete-portfolio \
    --id port-abcdlx4gox4do
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 DeletePortfolio를 참조하세요.

delete-product

다음 코드 예시에서는 delete-product의 사용 방법을 보여줍니다.

AWS CLI

제품 삭제

다음 delete-product 예시에서는 지정된 제품을 삭제합니다.

```
aws servicecatalog delete-product \
    --id prod-abcdcek6yhbxi
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteProduct를 참조하세요.

delete-provisioning-artifact

다음 코드 예시에서는 delete-provisioning-artifact의 사용 방법을 보여줍니다.

AWS CLI

프로비저닝 아티팩트 삭제

다음 delete-provisioning-artifact 예시에서는 지정된 프로비저닝 아티팩트를 삭제합니다.

```
aws servicecatalog delete-provisioning-artifact \
    --product-id prod-abc2uebuplcpw \
    --provisioning-artifact-id pa-pqabcddii7ouc
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteProvisioningArtifact를 참조하세요.

delete-tag-option

다음 코드 예시에서는 delete-tag-option의 사용 방법을 보여줍니다.

AWS CLI

TagOption 삭제

다음 delete-tag-option 예시에서는 지정된 TagOption을 삭제합니다.

```
aws servicecatalog delete-tag-option \
    --id tag-iabcdn4fzjjms
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteTagOption을 참조하세요.

describe-copy-product-status

다음 코드 예시에서는 describe-copy-product-status의 사용 방법을 보여줍니다.

AWS CLI

제품 복사 작업의 상태 설명

다음 describe-copy-product-status 예시에서는 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeCopyProductStatus를 참조하세요.

describe-portfolio

다음 코드 예시에서는 describe-portfolio의 사용 방법을 보여줍니다.

AWS CLI

포트폴리오 설명

다음 describe-portfolio 예시에서는 지정된 포트폴리오의 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": []
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribePortfolio를 참조하세요.

describe-product-as-admin

다음 코드 예시에서는 describe-product-as-admin의 사용 방법을 보여줍니다.

AWS CLI

관리자로서 제품 설명

다음 describe-product-as-admin 예시에서는 관리자 권한을 사용하여 지정된 제품의 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeProductAsAdmin을 참조하세요.

describe-provisioned-product

다음 코드 예시에서는 describe-provisioned-product의 사용 방법을 보여줍니다.

AWS CLI

프로비저닝된 제품 설명

다음 describe-provisioned-product 예시에서는 지정된 프로비저닝된 제품의 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": []
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeProvisionedProduct를 참조하세요.

describe-provisioning-artifact

다음 코드 예시에서는 describe-provisioning-artifact의 사용 방법을 보여줍니다.

AWS CLI

프로비저닝 아티팩트 설명

다음 describe-provisioning-artifact 예시에서는 지정된 프로비저닝 아티팩트의 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeProvisioningArtifact를 참조하세요.

describe-tag-option

다음 코드 예시에서는 describe-tag-option의 사용 방법을 보여줍니다.

AWS CLI

TagOption 설명

다음 describe-tag-option 예시에서는 지정된 TagOption의 세부 정보를 표시합니다.

```
aws servicecatalog describe-tag-option \
    --id tag-p3tej2abcd5qc
```

출력:

```
{
    "TagOptionDetail": {
        "Active": true,
        "Id": "tag-p3tej2abcd5qc",
        "Value": "value-3",
        "Key": "1234"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeTagOption을 참조하세요.

disassociate-principal-from-portfolio

다음 코드 예시에서는 disassociate-principal-from-portfolio의 사용 방법을 보여줍니다.

AWS CLI

포트폴리오에서 위탁자 연결 해제

다음 disassociate-principal-from-portfolio 예시에서는 지정된 위탁자를 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.

- API 세부 정보는 AWS CLI 명령 참조의 DisassociatePrincipalFromPortfolio를 참조하세요.

disassociate-product-from-portfolio

다음 코드 예시에서는 disassociate-product-from-portfolio의 사용 방법을 보여줍니다.

AWS CLI

포트폴리오에서 제품 연결 해제

다음 disassociate-product-from-portfolio 예시에서는 지정된 제품을 포트폴리오에서 연결 해제합니다.

```
aws servicecatalog disassociate-product-from-portfolio \
    --product-id prod-3p5abcdmu3oyk \
    --portfolio-id port-2s6abcdq5wdh4
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 DisassociateProductFromPortfolio를 참조하세요.

disassociate-tag-option-from-resource

다음 코드 예시에서는 disassociate-tag-option-from-resource의 사용 방법을 보여줍니다.

AWS CLI

리소스에서 TagOption 연결 해제

다음 disassociate-tag-option-from-resource 예시에서는 지정된 TagOption을 리소스에서 연결 해제합니다.

```
aws servicecatalog disassociate-tag-option-from-resource \
     --resource-id port-2s6abcdq5wdh4 \
     --tag-option-id tag-p3abc2pkpz5qc
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 DisassociateTagOptionFromResource를 참조하세요.

list-accepted-portfolio-shares

다음 코드 예시에서는 `list-accepted-portfolio-shares`의 사용 방법을 보여줍니다.

AWS CLI

수락된 포트폴리오 공유 나열

다음 `list-accepted-portfolio-shares` 예시에서는 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListAcceptedPortfolioShares를 참조하세요.

list-portfolio-access

다음 코드 예시에서는 `list-portfolio-access`의 사용 방법을 보여줍니다.

AWS CLI

포트폴리오에 액세스할 수 있는 계정 나열

다음 `list-portfolio-access` 예시에서는 지정된 포트폴리오에 액세스할 수 있는 AWS 계정을 나열합니다.

```
aws servicecatalog list-portfolio-access \
    --portfolio-id port-2s6abcdq5wdh4
```

출력:

```
{
    "AccountIds": [
        "123456789012"
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListPortfolioAccess를 참조하세요.

list-portfolios-for-product

다음 코드 예시에서는 `list-portfolios-for-product`의 사용 방법을 보여줍니다.

AWS CLI

제품에 연결된 포트폴리오 나열

다음 `list-portfolios-for-product` 예시에서는 지정된 제품에 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListPortfoliosForProduct를 참조하세요.

`list-portfolios`

다음 코드 예시에서는 `list-portfolios`의 사용 방법을 보여줍니다.

AWS CLI

포트폴리오 나열

다음 `list-portfolios` 예시에서는 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListPortfolios를 참조하세요.

list-principals-for-portfolio

다음 코드 예시에서는 list-principals-for-portfolio의 사용 방법을 보여줍니다.

AWS CLI

포트폴리오의 모든 위탁자 나열

다음 list-principals-for-portfolio 예시에서는 지정된 포트폴리오의 모든 위탁자를 나열합니다.

```
aws servicecatalog list-principals-for-portfolio \
    --portfolio-id port-2s6abcdq5wdh4
```

출력:

```
{
    "Principals": [
        {
            "PrincipalARN": "arn:aws:iam::123456789012:user/usertest",
            "PrincipalType": "IAM"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListPrincipalsForPortfolio를 참조하세요.

list-provisioning-artifacts

다음 코드 예시에서는 list-provisioning-artifacts의 사용 방법을 보여줍니다.

AWS CLI

제품의 모든 프로비저닝 아티팩트 나열

다음 `list-provisioning-artifacts` 예시에서는 지정된 제품의 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListProvisioningArtifacts를 참조하세요.

`list-resources-for-tag-option`

다음 코드 예시에서는 `list-resources-for-tag-option`의 사용 방법을 보여줍니다.

AWS CLI

TagOption에 연결된 리소스 나열

다음 `list-resources-for-tag-option` 예시에서는 지정된 TagOption에 연결된 리소스를 나열합니다.

```
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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListResourcesForTagOption을 참조하세요.

`list-tag-options`

다음 코드 예시에서는 `list-tag-options`의 사용 방법을 보여줍니다.

AWS CLI

다음 `list-tag-options` 예시에서는 TagOptions의 모든 값을 나열합니다.

```
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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [ListTagOptions](https://docs.aws.amazon.com/goto/aws-cli/servicecatalog-2015-12-10/ListTagOptions)를 참조하세요.

`provision-product`

다음 코드 예시에서는 `provision-product`의 사용 방법을 보여줍니다.

AWS CLI

제품 프로비저닝

다음 `provision-product` 예시에서는 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"
        }
    }
```

- API 세부 정보는 AWS CLI 명령 참조의 [ProvisionProduct](https://docs.aws.amazon.com/goto/aws-cli/servicecatalog-2015-12-10/ProvisionProduct)를 참조하세요.

reject-portfolio-share

다음 코드 예시에서는 reject-portfolio-share의 사용 방법을 보여줍니다.

AWS CLI

포트폴리오 공유 거부

다음 reject-portfolio-share 예시에서는 지정된 포트폴리오의 포트폴리오 공유를 거부합니다.

```
aws servicecatalog reject-portfolio-share \
    --portfolio-id port-2s6wuabcdefghijk
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 [RejectPortfolioShare](https://docs.aws.amazon.com/goto/aws-cli/servicecatalog-2015-12-10/RejectPortfolioShare)를 참조하세요.

scan-provisioned-products

다음 코드 예시에서는 scan-provisioned-products의 사용 방법을 보여줍니다.

AWS CLI

모든 사용 가능한 프로비저닝된 제품 나열

다음 scan-provisioned-products 예시에서는 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ScanProvisionedProducts를 참조하세요.

search-products-as-admin

다음 코드 예시에서는 search-products-as-admin의 사용 방법을 보여줍니다.

AWS CLI

관리자 권한으로 제품 검색

다음 search-products-as-admin 예시에서는 포트폴리오 ID를 필터로 사용하여 관리자 권한으로 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 SearchProductsAsAdmin을 참조하세요.

`search-provisioned-products`

다음 코드 예시에서는 `search-provisioned-products`의 사용 방법을 보여줍니다.

AWS CLI

프로비저닝된 제품 검색

다음 `search-provisioned-products` 예시에서는 JSON 파일을 사용해 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 SearchProvisionedProducts를 참조하세요.

update-portfolio

다음 코드 예시에서는 update-portfolio의 사용 방법을 보여줍니다.

AWS CLI

포트폴리오 업데이트

다음 update-portfolio 예시에서는 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": []
}
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdatePortfolio를 참조하세요.

`update-product`

다음 코드 예시에서는 update-product의 사용 방법을 보여줍니다.

AWS CLI

제품 업데이트

다음 update-product 예시에서는 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
    }
```

```
}
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateProduct를 참조하세요.

update-provisioning-artifact

다음 코드 예시에서는 update-provisioning-artifact의 사용 방법을 보여줍니다.

AWS CLI

프로비저닝 아티팩트 업데이트

다음 update-provisioning-artifact 예시에서는 JSON 파일을 사용해 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
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateProvisioningArtifact를 참조하세요.

update-tag-option

다음 코드 예시에서는 update-tag-option의 사용 방법을 보여줍니다.

AWS CLI

TagOption 업데이트

다음 update-tag-option 예시에서는 지정된 JSON 파일을 사용하여 TagOption의 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateTagOption을 참조하세요.

AWS CLI를 사용한 Service Quotas 예시

다음 코드 예시는 Service Quotas와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

작업

`get-aws-default-service-quota`

다음 코드 예시에서는 `get-aws-default-service-quota`의 사용 방법을 보여줍니다.

AWS CLI

기본 서비스 할당량 설명

다음 `get-aws-default-service-quota` 예시에서는 지정된 할당량의 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"
            }
        }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetAwsDefaultServiceQuota를 참조하세요.

get-requested-service-quota-change

다음 코드 예시에서는 get-requested-service-quota-change의 사용 방법을 보여줍니다.

AWS CLI

서비스 할당량 증가 요청 설명

다음 get-requested-service-quota-change 예시에서는 지정된 할당량 증가 요청에 대해 설명합니다.

```
aws service-quotas get-requested-service-quota-change \
    --request-id d187537d15254312a9609aa51bbf7624u7W49tPO
```

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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetRequestedServiceQuotaChange를 참조하세요.

get-service-quota

다음 코드 예시에서는 get-service-quota의 사용 방법을 보여줍니다.

AWS CLI

서비스 할당량 설명

다음 get-service-quota 예시에서는 지정된 할당량의 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"
        }
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetServiceQuota를 참조합니다.

list-aws-default-service-quotas

다음 코드 예시에서는 `list-aws-default-service-quotas`의 사용 방법을 보여줍니다.

AWS CLI

서비스의 기본 할당량 나열

다음 `list-aws-default-service-quotas` 예시에서는 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
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListAwsDefaultServiceQuotas를 참조하세요.

list-requested-service-quota-change-history-by-quota

다음 코드 예시에서는 `list-requested-service-quota-change-history-by-quota`의 사용 방법을 보여줍니다.

AWS CLI

할당량 증가 요청 나열

다음 `list-requested-service-quota-change-history-by-quota` 예시에서는 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListRequestedServiceQuotaChangeHistoryByQuota를 참조합니다.

`list-requested-service-quota-change-history`

다음 코드 예시에서는 `list-requested-service-quota-change-history`의 사용 방법을 보여줍니다.

AWS CLI

할당량 증가 요청 나열

다음 `list-requested-service-quota-change-history` 예시에서는 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListRequestedServiceQuotaChangeHistory를 참조하세요.

list-service-quotas

다음 코드 예시에서는 `list-service-quotas`의 사용 방법을 보여줍니다.

AWS CLI

서비스 할당량 나열

다음 `list-service-quotas` 예시에서는 AWS CloudFormation의 할당량에 대한 세부 정보를 표시합니다.

```
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
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListServiceQuotas를 참조하세요.

`list-services`

다음 코드 예시에서는 `list-services`의 사용 방법을 보여줍니다.

AWS CLI

사용 가능한 서비스 나열

다음 명령은 Service Quotas에서 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 표시 내용을 필터링할 수 있습니다. 다음 예시에서는 서비스 코드만 표시합니다.

```
aws service-quotas list-services \
    --query Services[*].ServiceCode
```

출력:

```
[
    "AWSCloudMap",
    "access-analyzer",
    "acm",
    "acm-pca",
    "amplify",
    "apigateway",
    "application-autoscaling",
        ...truncated...
    "xray"
]
```

- API 세부 정보는 AWS CLI 명령 참조의 ListServices를 참조하세요.

request-service-quota-increase

다음 코드 예시에서는 `request-service-quota-increase`의 사용 방법을 보여줍니다.

AWS CLI

서비스 할당량 증가 요청

다음 `request-service-quota-increase` 예시에서는 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 RequestServiceQuotaIncrease를 참조하세요.

AWS CLI를 사용한 Amazon SES 예시

다음 코드 예시는 Amazon SES와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

delete-identity

다음 코드 예시에서는 `delete-identity` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

ID 삭제

다음 예시에서는 `delete-identity` 명령을 사용하여 Amazon SES로 확인된 ID 목록에서 ID를 삭제합니다.

```
aws ses delete-identity --identity user@example.com
```

확인된 ID에 대한 자세한 내용은 Amazon Simple Email Service 개발자 안내서의 Amazon SES에서 이메일 주소 및 도메인 확인을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteIdentity](#)를 참조하세요.

get-identity-dkim-attributes

다음 코드 예시에서는 `get-identity-dkim-attributes` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

자격 증명 목록에 대한 Amazon SES Easy DKIM 속성을 가져오는 방법

다음 예시에서는 `get-identity-dkim-attributes` 명령을 사용하여 ID 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 ID를 사용하여 이 명령을 직접적으로 호출하는 경우 해당 ID는 출력에 표시되지 않습니다.

Easy DKIM에 대한 자세한 내용은 Amazon Simple Email Service 개발자 안내서의 Amazon SES의 간편한 DKIM을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [GetIdentityDkimAttributes](https://docs.aws.amazon.com/goto/aws-cli/email-2010-12-01/GetIdentityDkimAttributes) 섹션을 참조하세요.

get-identity-notification-attributes

다음 코드 예시에서는 `get-identity-notification-attributes` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

자격 증명 목록에 대한 Amazon SES 알림 속성을 가져오는 방법

다음 예시에서는 `get-identity-notification-attributes` 명령을 사용하여 ID 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 상태와 해당되는 경우 반송, 불만 및 배송 알림이 전송되는 Amazon SNS 주제의 Amazon 리소스 이름(ARN)을 반환합니다.

확인을 위해 제출한 적이 없는 ID를 사용하여 이 명령을 직접적으로 호출하는 경우 해당 ID는 출력에 표시되지 않습니다.

알림에 대한 자세한 내용은 Amazon Simple Email Service 개발자 안내서의 Amazon SES에서 알림 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetIdentityNotificationAttributes 섹션을 참조하세요.

get-identity-verification-attributes

다음 코드 예시에서는 get-identity-verification-attributes 코드를 사용하는 방법을 보여줍니다.

AWS CLI

ID 목록의 Amazon SES 확인 상태 가져오기

다음 예시에서는 get-identity-verification-attributes 명령을 사용하여 ID 목록에 대한 Amazon SES 확인 상태를 가져옵니다.

```
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 ID를 사용하여 이 명령을 직접적으로 호출하는 경우 해당 ID는 출력에 표시되지 않습니다.

확인된 ID에 대한 자세한 내용은 Amazon Simple Email Service 개발자 안내서의 Amazon SES에서 이메일 주소 및 도메인 확인을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetIdentityVerificationAttributes를 참조하세요.

get-send-quota

다음 코드 예시에서는 get-send-quota 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Amazon SES 발신 한도 가져오기

다음 예시에서는 `get-send-quota` 명령을 사용하여 Amazon SES 발신 한도를 반환합니다.

```
aws ses get-send-quota
```

출력:

```
{
   "Max24HourSend": 200.0,
   "SentLast24Hours": 1.0,
   "MaxSendRate": 1.0
}
```

Max24HourSend는 발신 할당량으로, 24시간 내에 보낼 수 있는 최대 이메일 수입니다. 발신 할당량은 롤링 기간을 반영합니다. 이메일 전송을 시도할 때마다 Amazon SES에서 지난 24시간 내에 보낸 이메일 수를 확인합니다. 보낸 이메일의 총 수가 할당량보다 적으면 전송 요청이 수락되고 이메일이 전송됩니다.

SentLast24Hours는 지난 24시간 내에 보낸 이메일 수입니다.

MaxSendRate는 초당 보낼 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 발신 한도 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetSendQuota를 참조하세요.

`get-send-statistics`

다음 코드 예시에서는 `get-send-statistics` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Amazon SES 발신 통계 가져오기

다음 예시에서는 `get-send-statistics` 명령을 사용하여 Amazon SES 발신 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
       }
   ]
}
```

결과는 지난 2주 동안의 전송 활동을 나타내는 데이터 포인트 목록입니다. 목록의 각 데이터 포인트에는 15분 간격의 통계가 포함됩니다.

이 예시에서는 지난 2주 동안 사용자가 보낸 유일한 이메일이 두 번의 15분 간격 내에 속했기 때문에 두 개의 데이터 포인트만 있습니다.

자세한 내용은 Amazon Simple Email Service 개발자 안내서의 Amazon SES 사용 통계 모니터링을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetSendStatistics 섹션을 참조하세요.

`list-identities`

다음 코드 예시에서는 `list-identities` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

특정 AWS 계정에 대한 모든 ID(이메일 주소 및 도메인) 나열

다음 예시에서는 `list-identities` 명령을 사용하여 Amazon SES로 확인하기 위해 제출된 모든 ID를 나열합니다.

```
aws ses list-identities
```

출력:

```
{
    "Identities": [
      "user@example.com",
      "example.com"
    ]
}
```

반환되는 목록에는 확인 상태(확인됨, 확인 보류 중, 실패 등)와 상관없이 모든 ID가 포함됩니다.

이 예시에서는 ID 유형 파라미터를 지정하지 않았으므로 이메일 주소와 도메인이 모두 반환됩니다.

확인에 대한 자세한 내용은 Amazon Simple Email Service 개발자 안내서의 Amazon SES에서 이메일 주소 및 도메인 확인을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListIdentities를 참조하세요.

send-email

다음 코드 예시에서는 `send-email` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Amazon SES를 사용하여 서식 지정된 이메일 전송

다음 예시에서는 `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"
}
```

대상 및 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. 단, 발신자의 이메일 주소는 Amazon SES로 확인해야 합니다. Amazon SES에 대한 프로덕션 액세스 권한을 부여받기 전까지는 각 수신자의 이메일 주소도 확인해야 합니다. 단, 수신자가 Amazon SES 메일박스 시뮬레이터인 경우는 예외입니다. 확인에 대한 자세한 내용은 Amazon Simple Email Service 개발자 안내서의 Amazon SES에서 이메일 주소 및 도메인 확인을 참조하세요.

출력의 Message ID는 직접적인 send-email 호출이 성공했음을 나타냅니다.

이메일을 받지 못한 경우 정크 박스를 확인해 보세요.

서식 지정된 이메일을 보내는 방법에 대한 자세한 내용은 Amazon Simple Email Service 개발자 안내서의 Amazon SES API를 사용하여 서식 지정된 이메일 전송을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 SendEmail을 참조하세요.

send-raw-email

다음 코드 예시에서는 `send-raw-email` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Amazon SES를 사용하여 원시 이메일 전송

다음 예시에서는 `send-raw-email` 명령을 사용하여 TXT 첨부 파일이 있는 이메일을 보냅니다.

```
aws ses send-raw-email --raw-message file://message.json
```

출력:

```
{
   "MessageId": "EXAMPLEf3f73d99b-c63fb06f-d263-41f8-a0fb-d0dc67d56c07-000000"
}
```

원시 메시지는 현재 디렉터리의 `message.json` 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 'Data'는 attachment.txt 첨부 파일을 포함하여 MIME 형식의 원시 이메일 콘텐츠 전체를 포함하는 하나의 긴 문자열입니다.

sender@example.com 및 recipient@example.com을 사용하려는 주소로 바꿉니다. 단, 발신자의 이메일 주소는 Amazon SES로 확인해야 합니다. Amazon SES에 대한 프로덕션 액세스 권한을 부

여받기 전까지는 수신자의 이메일 주소도 확인해야 합니다. 단, 수신자가 Amazon SES 메일박스 시뮬레이터가 아닌 경우는 예외입니다. 확인에 대한 자세한 내용은 Amazon Simple Email Service 개발자 안내서의 Amazon SES에서 이메일 주소 및 도메인 확인을 참조하세요.

출력의 Message ID는 직접적인 send-raw-email 호출이 성공했음을 나타냅니다.

이메일을 받지 못한 경우 정크 박스를 확인해 보세요.

원시 이메일을 보내는 방법에 대한 자세한 내용은 Amazon Simple Email Service 개발자 안내서의 Amazon SES API를 사용하여 원시 이메일 전송을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [SendRawEmail](#)을 참조하세요.

`set-identity-dkim-enabled`

다음 코드 예시에서는 `set-identity-dkim-enabled` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Amazon SES 확인 자격 증명에 대해 Easy DKIM을 활성화 또는 비활성화하는 방법

다음 예시에서는 `set-identity-dkim-enabled` 명령을 사용하여 인증된 이메일 주소에 대해 DKIM을 비활성화합니다.

```
aws ses set-identity-dkim-enabled --identity user@example.com --no-dkim-enabled
```

Easy DKIM에 대한 자세한 내용은 Amazon Simple Email Service 개발자 안내서의 Amazon SES의 간편한 DKIM을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [SetIdentityDkimEnabled](#) 섹션을 참조하세요.

`set-identity-feedback-forwarding-enabled`

다음 코드 예시에서는 `set-identity-feedback-forwarding-enabled` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Amazon SES 확인 자격 증명에 대한 반송 및 불만 이메일 피드백 전달을 활성화 또는 비활성화하는 방법

다음 예시에서는 `set-identity-feedback-forwarding-enabled` 명령을 사용하여 인증된 이메일 주소로 반송 및 불만 알림을 이메일로 받을 수 있도록 설정합니다.

```
aws ses set-identity-feedback-forwarding-enabled --identity user@example.com --
forwarding-enabled
```

반송 및 불만 알림은 Amazon SNS 또는 이메일 피드백 전달을 통해 수신해야 하므로 반송 및 불만 알림 모두에 대해 Amazon SNS 주제를 선택한 경우에만 이메일 피드백 전달을 비활성화할 수 있습니다.

알림에 대한 자세한 내용은 Amazon Simple Email Service 개발자 안내서의 Amazon SES에서 알림 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 SetIdentityFeedbackForwardingEnabled 섹션을 참조하세요.

`set-identity-notification-topic`

다음 코드 예시에서는 `set-identity-notification-topic` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Amazon SES가 확인된 자격 증명에 대한 반송 메일, 불만 사항 및/또는 전송 알림을 게시할 Amazon SNS 주제를 설정하는 방법

다음 예시에서는 `set-identity-notification-topic` 명령을 사용하여 인증된 이메일 주소가 반송 알림을 수신할 Amazon SNS 토픽을 지정할 수 있습니다.

```
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 Amazon SES에서 알림 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 SetIdentityNotificationTopic 섹션을 참조하세요.

`verify-domain-dkim`

다음 코드 예시에서는 `verify-domain-dkim` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Amazon SES를 사용한 DKIM 서명을 위해 확인된 도메인의 DKIM 토큰을 생성하는 방법

다음 예시에서는 `verify-domain-dkim` 명령을 사용하여 Amazon SES로 확인된 도메인에 대한 DKIM 토큰을 생성합니다.

```
aws ses verify-domain-dkim --domain example.com
```

출력:

```
{
   "DkimTokens": [
       "EXAMPLEq76owjnks3lnluwg65scbemvw",
       "EXAMPLEi3dnsj67hstzaj673klariwx2",
       "EXAMPLEwfbtcukvimehexktmdtaz6naj"
   ]
}
```

DKIM을 설정하려면 반환된 DKIM 토큰을 사용하여 Amazon SES 에서 호스팅하는 DKIM 퍼블릭 키를 가리키는 CNAME 레코드로 도메인의 DNS 설정을 업데이트해야 합니다. 자세한 내용은 Amazon Simple Email Service 개발자 안내서의 Easy DKIM in Amazon SES를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 VerifyDomainDkim 섹션을 참조하세요.

`verify-domain-identity`

다음 코드 예시에서는 `verify-domain-identity` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Amazon SES를 사용하여 도메인 확인

다음 예시에서는 `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 도메인의 DNS 설정에 추가해야 합니다. 자세한 내용은 Amazon Simple Email Service 개발자 안내서의 Amazon SES에서 도메인 확인을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 VerifyDomainIdentity를 참조하세요.

`verify-email-identity`

다음 코드 예시에서는 `verify-email-identity` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

Amazon SES로 이메일 주소 확인

다음 예시에서는 `verify-email-identity` 명령을 사용하여 이메일 주소를 확인합니다.

```
aws ses verify-email-identity --email-address user@example.com
```

Amazon SES를 사용하여 이메일을 보내려면 발신 이메일 주소 또는 도메인이 사용자 본인의 소유인지 확인해야 합니다. 프로덕션 액세스 권한이 아직 없는 경우Amazon SES 메일박스 시뮬레이터에서 제공하는 이메일 주소를 제외하고 이메일을 전송하는 모든 이메일 주소도 확인해야 합니다.

verify-email-identity가 직접적으로 호출되면 해당 이메일 주소에 확인 이메일이 전송됩니다. 확인 프로세스를 완료하려면 이메일에 포함된 링크를 클릭해야 합니다.

자세한 내용은 Amazon Simple Email Service 개발자 안내서의 Amazon SES에서 이메일 주소 확인을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 VerifyEmailIdentity를 참조하세요.

AWS CLI를 사용한 Shield 예시

다음 코드 예시는 Shield와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- [작업](#작업)

작업

associate-drt-log-bucket

다음 코드 예시에서는 `associate-drt-log-bucket`의 사용 방법을 보여줍니다.

AWS CLI

DRT가 Amazon S3 버킷에 액세스할 수 있는 권한 부여

다음 `associate-drt-log-bucket` 예시에서는 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 Advanced 개발자 안내서의 DDoS 대응 팀에 권한 부여를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AssociateDrtLogBucket을 참조하세요.

associate-drt-role

다음 코드 예시에서는 `associate-drt-role`의 사용 방법을 보여줍니다.

AWS CLI

DRT가 사용자를 대신하여 잠재적 공격을 완화할 수 있는 권한 부여

다음 `associate-drt-role` 예시에서는 DRT와 지정된 역할 간의 연결을 생성합니다. DRT는 역할을 사용하여 계정을 액세스 및 관리할 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 Advanced 개발자 안내서의 DDoS 대응 팀에 권한 부여를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AssociateDrtRole을 참조하세요.

create-protection

다음 코드 예시에서는 create-protection의 사용 방법을 보여줍니다.

AWS CLI

단일 AWS 리소스에 대해 AWS Shield Advanced 보호 활성화

다음 create-protection 예시에서는 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 Advanced 개발자 안내서의 보호할 리소스 지정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateProtection을 참조하세요.

create-subscription

다음 코드 예시에서는 create-subscription의 사용 방법을 보여줍니다.

AWS CLI

계정에 대해 AWS Shield Advanced 보호 활성화

다음 create-subscription 예시에서는 계정에 대해 Shield Advanced 보호를 활성화합니다.

```
aws shield create-subscription
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Shield Advanced 개발자 안내서의 AWS Shield Advanced 시작하기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateSubscription을 참조합니다.

delete-protection

다음 코드 예시에서는 delete-protection의 사용 방법을 보여줍니다.

AWS CLI

AWS 리소스에서 AWS Shield Advanced 보호 제거

다음 delete-protection 예시에서는 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 AWS Shield Advanced 제거를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteProtection을 참조하세요.

describe-attack

다음 코드 예시에서는 describe-attack의 사용 방법을 보여줍니다.

AWS CLI

공격에 대한 자세한 설명 가져오기

다음 describe-attack 예시에서는 지정된 공격 ID를 가진 DDoS 공격의 세부 정보를 표시합니다. 공격 ID는 list-attacks 명령을 실행하여 가져올 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 Advanced 개발자 안내서의 DDoS 인시던트 검토를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeAttack을 참조하세요.

describe-drt-access

다음 코드 예시에서는 describe-drt-access의 사용 방법을 보여줍니다.

AWS CLI

DRT가 사용자를 대신하여 공격을 완화할 수 있는 권한에 대한 설명 가져오기

다음 describe-drt-access 예시에서는 사용자를 대신하여 잠재적 공격에 대응할 수 있도록 하는 DRT의 역할 및 S3 버킷 권한을 가져옵니다.

```
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 Advanced 개발자 안내서의 DDoS 대응 팀에 권한 부여를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDrtAccess를 참조하세요.

describe-emergency-contact-settings

다음 코드 예시에서는 describe-emergency-contact-settings의 사용 방법을 보여줍니다.

AWS CLI

DRT에 저장된 긴급 이메일 주소 가져오기

다음 describe-emergency-contact-settings 예시에서는 계정의 DRT에 저장된 이메일 주소를 가져옵니다. 다음은 의심되는 공격에 대응할 때 DRT가 연락해야 하는 주소입니다.

```
aws shield describe-emergency-contact-settings
```

출력:

```
{
    "EmergencyContactList": [
        {
            "EmailAddress": "ops@example.com"
        },
        {
            "EmailAddress": "ddos-notifications@example.com"
       }
    ]
}
```

자세한 내용은 AWS Shield Advanced 개발자 안내서의 AWS Shield 작동 방식<https://docs.aws.amazon.com/waf/latest/developerguide/ddos-overview.html>을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeEmergencyContactSettings를 참조하세요.

describe-protection

다음 코드 예시에서는 describe-protection의 사용 방법을 보여줍니다.

AWS CLI

AWS Shield Advanced 보호의 세부 정보 가져오기

다음 describe-protection 예시에서는 지정된 ID를 가진 Shield Advanced 보호의 세부 정보를 표시합니다. 보호 ID는 list-protections 명령을 실행하여 얻을 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 Advanced 개발자 안내서의 보호할 리소스 지정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeProtection을 참조하세요.

describe-subscription

다음 코드 예시에서는 describe-subscription의 사용 방법을 보여줍니다.

AWS CLI

계정에 대한 AWS Shield Advanced 보호의 세부 정보 가져오기

다음 describe-subscription 예시에서는 계정에 제공된 Shield Advanced 보호의 세부 정보를 표시합니다.

```
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 Advanced 개발자 안내서의 AWS Shield 작동 방식을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeSubscription을 참조하세요.

disassociate-drt-log-bucket

다음 코드 예시에서는 disassociate-drt-log-bucket의 사용 방법을 보여줍니다.

AWS CLI

DRT가 사용자를 대신하여 Amazon S3 버킷에 액세스할 수 있는 권한 제거

다음 disassociate-drt-log-bucket 예시에서는 DRT와 지정된 S3 버킷 간의 연결을 제거합니다. 이 명령이 완료되면 DRT는 더 이상 계정을 대신하여 버킷에 액세스할 수 없습니다.

```
aws shield disassociate-drt-log-bucket \
    --log-bucket flow-logs-for-website-lb
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Shield Advanced 개발자 안내서의 DDoS 대응 팀에 권한 부여를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisassociateDrtLogBucket을 참조하세요.

disassociate-drt-role

다음 코드 예시에서는 disassociate-drt-role의 사용 방법을 보여줍니다.

AWS CLI

DRT가 사용자를 대신하여 잠재적 공격을 완화할 수 있는 권한 제거

다음 `disassociate-drt-role` 예시에서는 DRT와 계정 간의 연결을 제거합니다. 이 직접적 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 Advanced 개발자 안내서의 DDoS 대응 팀에 권한 부여를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisassociateDrtRole을 참조하세요.

`get-subscription-state`

다음 코드 예시에서는 `get-subscription-state`의 사용 방법을 보여줍니다.

AWS CLI

계정의 AWS Shield Advanced 구독의 현재 상태 가져오기

다음 `get-subscription-state` 예시에서는 계정에 대한 Shield Advanced 보호의 상태를 가져옵니다.

```
aws shield get-subscription-state
```

출력:

```
{
    "SubscriptionState": "ACTIVE"
}
```

자세한 내용은 AWS Shield Advanced 개발자 안내서의 AWS Shield 작동 방식을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetSubscriptionState를 참조하세요.

`list-attacks`

다음 코드 예시에서는 `list-attacks`의 사용 방법을 보여줍니다.

AWS CLI

AWS Shield Advanced에서 공격 요약 가져오기

다음 `list-attacks` 예시에서는 지정된 기간 동안 지정된 AWS CloudFront 배포에 대한 공격 요약을 가져옵니다. 응답에는 공격에 대한 자세한 정보를 얻기 위해 `describe-attack` 명령에 제공할 수 있는 공격 ID가 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 Advanced 개발자 안내서의 DDoS 인시던트 검토를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListAttacks를 참조하세요.

list-protections

다음 코드 예시에서는 `list-protections`의 사용 방법을 보여줍니다.

AWS CLI

AWS Shield Advanced에서 보호 요약 가져오기

다음 `list-protections` 예시에서는 계정에 대해 활성화된 보호에 대한 요약을 가져옵니다.

```
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 Advanced 개발자 안내서의 보호할 리소스 지정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListProtections를 참조하세요.

update-emergency-contact-settings

다음 코드 예시에서는 update-emergency-contact-settings의 사용 방법을 보여줍니다.

AWS CLI

DRT에 저장된 긴급 이메일 주소 정의

다음 update-emergency-contact-settings 예시에서는 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 Advanced 개발자 안내서의 AWS Shield 작동 방식을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateEmergencyContactSettings를 참조하세요.

update-subscription

다음 코드 예시에서는 update-subscription의 사용 방법을 보여줍니다.

AWS CLI

계정의 AWS Shield Advanced 구독 수정

다음 `update-subscription` 예시에서는 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 Advanced 개발자 안내서의 [AWS Shield 작동 방식]을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [UpdateSubscription]을 참조하세요.

AWS CLI를 사용한 Signer 예시

다음 코드 예시는 Signer와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`cancel-signing-profile`

다음 코드 예시에서는 `cancel-signing-profile`의 사용 방법을 보여줍니다.

AWS CLI

서명 프로파일 삭제

다음 `cancel-signing-profile` 예시에서는 AWS Signer에서 기존 서명 프로파일을 제거합니다.

```
aws signer cancel-signing-profile \
    --profile-name MyProfile1
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 CancelSigningProfile을 참조하세요.

describe-signing-job

다음 코드 예시에서는 describe-signing-job의 사용 방법을 보여줍니다.

AWS CLI

서명 작업의 세부 정보 표시

다음 describe-signing-job 예시에서는 지정된 서명 작업의 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeSigningJob을 참조하세요.

get-signing-platform

다음 코드 예시에서는 get-signing-platform의 사용 방법을 보여줍니다.

AWS CLI

서명 플랫폼의 세부 정보 표시

다음 get-signing-platform 예시에서는 지정된 서명 플랫폼의 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"
        ]
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [GetSigningPlatform](https://docs.aws.amazon.com/goto/aws-cli/signer-2017-08-25/GetSigningPlatform)을 참조하세요.

get-signing-profile

다음 코드 예시에서는 `get-signing-profile`의 사용 방법을 보여줍니다.

AWS CLI

서명 프로파일의 세부 정보 표시

다음 `get-signing-profile` 예시에서는 지정된 서명 프로파일의 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [GetSigningProfile](https://docs.aws.amazon.com/goto/aws-cli/signer-2017-08-25/GetSigningProfile)을 참조하세요.

list-signing-jobs

다음 코드 예시에서는 `list-signing-jobs`의 사용 방법을 보여줍니다.

AWS CLI

모든 서명 작업 나열

다음 `list-signing-jobs` 예시에서는 계정의 모든 서명 작업에 대한 세부 정보를 표시합니다.

```
aws signer list-signing-jobs
```

이 예시에서는 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListSigningJobs를 참조하세요.

list-signing-platforms

다음 코드 예시에서는 list-signing-platforms의 사용 방법을 보여줍니다.

AWS CLI

모든 서명 플랫폼 나열

다음 list-signing-platforms 예시에서는 사용 가능한 모든 서명 플랫폼의 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"
                    ]
                }
            }
        ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListSigningPlatforms를 참조하세요.

list-signing-profiles

다음 코드 예시에서는 `list-signing-profiles`의 사용 방법을 보여줍니다.

AWS CLI

모든 서명 프로파일 나열

다음 `list-signing-profiles` 예시에서는 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"
            }
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListSigningProfiles를 참조하세요.

put-signing-profile

다음 코드 예시에서는 put-signing-profile의 사용 방법을 보여줍니다.

AWS CLI

서명 프로파일 생성

다음 put-signing-profile 예시에서는 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 PutSigningProfile을 참조하세요.

start-signing-job

다음 코드 예시에서는 start-signing-job의 사용 방법을 보여줍니다.

AWS CLI

서명 작업 시작

다음 start-signing-job 예시에서는 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 StartSigningJob을 참조하세요.

AWS CLI를 사용한 Snowball Edge 예시

다음 코드 예시에서는 Snowball Edge와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

작업

`get-snowball-usage`

다음 코드 예시에서는 `get-snowball-usage`의 사용 방법을 보여줍니다.

AWS CLI

계정의 Snowball 서비스 제한에 대한 정보 가져오기

다음 `get-snowball-usage` 코드 예시에서는 계정의 Snowball 서비스 제한과 계정이 사용 중인 Snowball 개수에 대한 정보를 표시합니다.

```
aws snowball get-snowball-usage
```

출력:

```
{
    "SnowballLimit": 1,
    "SnowballsInUse": 0
}
```

자세한 내용은 AWS Snowball 개발자 안내서의 AWS Snowball Edge 제한을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetSnowballUsage를 참조하세요.

list-jobs

다음 코드 예시에서는 `list-jobs`의 사용 방법을 보여줍니다.

AWS CLI

계정의 현재 Snowball 작업 나열

다음 `list-jobs` 예시에서는 `JobListEntry` 객체 배열을 표시합니다. 이 예시에서는 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 Edge 개발자 안내서의 AWS Snowball Edge 디바이스 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListJobs를 참조하세요.

AWS CLI를 사용한 Amazon SNS 예제

다음 코드 예제는 Amazon SNS와 함께 AWS Command Line Interface를 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.

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 예시는 add-permission의 사용 방법을 보여 줍니다.

AWS CLI

주제에 권한을 추가하는 방법

다음 add-permission 예제에서는 AWS 계정 987654321098이 AWS 계정 123456789012에서 지정된 주제로 Publish 작업을 사용할 수 있는 권한을 추가합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 AddPermission 섹션을 참조하세요.

check-if-phone-number-is-opted-out

다음 코드 예시는 check-if-phone-number-is-opted-out의 사용 방법을 보여 줍니다.

AWS CLI

전화번호의 SMS 메시지 옵트아웃을 확인하려면

다음 check-if-phone-number-is-opted-out예시에서는 지정된 전화번호가 현재 AWS계정의 SMS 메시지 수신을 옵트아웃했는지 여부를 확인합니다.

```
aws sns check-if-phone-number-is-opted-out \
    --phone-number +1555550100
```

출력:

```
{
    "isOptedOut": false
}
```

- API 세부 정보는 AWS CLI명령 참조의 CheckIfPhoneNumberIsOptedOut을 참조하세요.

confirm-subscription

다음 코드 예시는 confirm-subscription의 사용 방법을 보여 줍니다.

AWS CLI

구독을 확인하려면

다음 confirm-subscription명령은 my-topic라는 SNS 주제를 구독할 때 시작된 확인 프로세스를 완료합니다. --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"
}
```

- API 세부 정보는 AWS CLI명령 참조의 ConfirmSubscription을 참조하세요.

create-platform-application

다음 코드 예시는 create-platform-application의 사용 방법을 보여 줍니다.

AWS CLI

플랫폼 애플리케이션을 생성하는 방법

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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreatePlatformApplication 섹션을 참조하세요.

create-topic

다음 코드 예시는 create-topic의 사용 방법을 보여 줍니다.

AWS CLI

SNS 주제를 생성하려면

다음 create-topic예제에서는 my-topic이라는 SNS 주제를 생성합니다.

```
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 AWS명령줄 인터페이스 사용 설명서의 Amazon SQS 및 Amazon SNS와 함께 AWS 명령줄 인터페이스 사용을 참조하세요.

- API 세부 정보는 AWS CLI명령 참조의 CreateTopic을 참조하세요.

delete-endpoint

다음 코드 예시는 delete-endpoint의 사용 방법을 보여 줍니다.

AWS CLI

플랫폼 애플리케이션 엔드포인트를 생성하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteEndpoint 섹션을 참조하세요.

delete-platform-application

다음 코드 예시는 delete-platform-application의 사용 방법을 보여 줍니다.

AWS CLI

플랫폼 애플리케이션을 삭제하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 DeletePlatformApplication 섹션을 참조하세요.

delete-topic

다음 코드 예시는 delete-topic의 사용 방법을 보여 줍니다.

AWS CLI

SNS 주제를 삭제하려면

다음 delete-topic예제에서는 지정된 SNS 주제를 삭제합니다.

```
aws sns delete-topic \
    --topic-arn "arn:aws:sns:us-west-2:123456789012:my-topic"
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI명령 참조의 DeleteTopic을 참조하세요.

get-endpoint-attributes

다음 코드 예시는 get-endpoint-attributes의 사용 방법을 보여 줍니다.

AWS CLI

플랫폼 애플리케이션 엔드포인트 속성을 나열하는 방법

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..."
    }
```

```
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetEndpointAttributes 섹션을 참조하세요.

get-platform-application-attributes

다음 코드 예시는 get-platform-application-attributes의 사용 방법을 보여 줍니다.

AWS CLI

플랫폼 애플리케이션 속성을 나열하는 방법

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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetPlatformApplicationAttributes 섹션을 참조하세요.

get-sms-attributes

다음 코드 예시는 get-sms-attributes의 사용 방법을 보여 줍니다.

AWS CLI

기본 SMS 메시지 속성을 나열하려면

다음 get-sms-attributes예제에서는 SMS 메시지 전송의 기본 속성을 나열합니다.

```
aws sns get-sms-attributes
```

출력:

```
{
    "attributes": {
        "DefaultSenderID": "MyName"
    }
}
```

- API 세부 정보는 AWS CLI명령 참조의 GetSMSAttributes를 참조하세요.

get-subscription-attributes

다음 코드 예시는 get-subscription-attributes의 사용 방법을 보여 줍니다.

AWS CLI

주제에 대한 구독 속성을 검색하는 방법

다음 get-subscription-attributes 명령은 지정된 구독의 속성을 표시합니다. list-subscriptions 명령의 출력에서 subscription-arn를 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetSubscriptionAttributes 섹션을 참조하세요.

get-topic-attributes

다음 코드 예시는 get-topic-attributes의 사용 방법을 보여 줍니다.

AWS CLI

주제의 속성을 검색하려면

다음 get-topic-attributes예제에서는 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"
    }
}
```

- API 세부 정보는 AWS CLI명령 참조의 GetTopicAttributes를 참조하세요.

list-endpoints-by-platform-application

다음 코드 예시는 list-endpoints-by-platform-application의 사용 방법을 보여 줍니다.

AWS CLI

플랫폼 애플리케이션의 엔드포인트를 나열하는 방법

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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [ListEndpointsByPlatformApplication](#) 섹션을 참조하세요.

`list-phone-numbers-opted-out`

다음 코드 예시는 `list-phone-numbers-opted-out`의 사용 방법을 보여 줍니다.

AWS CLI

SMS 메시지 옵트아웃을 나열하려면

다음 `list-phone-numbers-opted-out`예제에서는 SMS 메시지 수신을 옵트아웃한 전화번호를 나열합니다.

```
aws sns list-phone-numbers-opted-out
```

출력:

```
{
    "phoneNumbers": [
        "+15555550100"
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListPhoneNumbersOptedOut을 참조하세요.

list-platform-applications

다음 코드 예시는 `list-platform-applications`의 사용 방법을 보여 줍니다.

AWS CLI

플랫폼 애플리케이션을 나열하는 방법

다음 `list-platform-applications` 예제에서는 ADM 및 MPNS용 플랫폼 애플리케이션을 나열합니다.

```
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"
```

```
            }
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListPlatformApplications 섹션을 참조하세요.

list-subscriptions-by-topic

다음 코드 예시는 `list-subscriptions-by-topic`의 사용 방법을 보여 줍니다.

AWS CLI

주제와 연결된 구독을 나열하는 방법

다음 `list-subscriptions-by-topic` 명령은 지정된 주제와 연결된 SNS 구독 목록을 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListSubscriptionsByTopic 섹션을 참조하세요.

list-subscriptions

다음 코드 예시는 `list-subscriptions`의 사용 방법을 보여 줍니다.

AWS CLI

SNS 구독을 나열하려면

다음 `list-subscriptions`예제에서는 AWS계정의 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [ListSubscriptions](https://docs.aws.amazon.com/goto/aws-cli/sns-2010-03-31/ListSubscriptions)를 참조하세요.

`list-tags-for-resource`

다음 코드 예시는 `list-tags-for-resource`의 사용 방법을 보여 줍니다.

AWS CLI

주제에 대한 태그를 나열하는 방법

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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource 섹션을 참조하세요.

list-topics

다음 코드 예시는 list-topics의 사용 방법을 보여 줍니다.

AWS CLI

SNS 주제를 나열하려면

다음 list-topics예제에서는 AWS계정의 모든 SNS 주제를 나열합니다.

```
aws sns list-topics
```

출력:

```
{
    "Topics": [
        {
            "TopicArn": "arn:aws:sns:us-west-2:123456789012:my-topic"
        }
    ]
}
```

- API 세부 정보는 AWS CLI명령 참조의 ListTopics를 참조하세요.

opt-in-phone-number

다음 코드 예시는 opt-in-phone-number의 사용 방법을 보여 줍니다.

AWS CLI

SMS 메시지에 옵트인하는 방법

다음 opt-in-phone-number 예제에서는 SMS 메시지 수신을 옵트아웃한 전화번호를 나열합니다.

```
aws sns opt-in-phone-number \
    --phone-number +15555550100
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 OptInPhoneNumber 섹션을 참조하세요.

publish

다음 코드 예시는 publish의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 주제에 메시지를 게시하려면

다음 publish예제에서는 지정된 Amazon SNS 주제에 지정된 메시지를 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 publish예제에서는 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"
}
```

- API 세부 정보는 AWS CLI명령 참조의 Publish를 참조하세요.

put-data-protection-policy

다음 코드 예시는 put-data-protection-policy의 사용 방법을 보여 줍니다.

AWS CLI

데이터 보호 정책을 설정하는 방법

예제 1: 게시자가 CreditCardNumber로 메시지를 게시하지 못하도록 거부하는 방법

다음 put-data-protection-policy 예제에서는 게시자가 CreditCardNumber 를 사용하여 메시지를 게시하는 것을 거부합니다.

```
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

다음 put-data-protection-policy 명령은 파일에서 파라미터를 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.

- API 세부 정보는 AWS CLI 명령 참조의 PutDataProtectionPolicy 섹션을 참조하세요.

remove-permission

다음 코드 예시는 remove-permission의 사용 방법을 보여 줍니다.

AWS CLI

주제에서 권한 집합을 제거하는 방법

다음 remove-permission 예제에서는 지정된 주제에서 Publish-Permission 권한을 제거합니다.

```
aws sns remove-permission \
    --topic-arn arn:aws:sns:us-west-2:123456789012:MyTopic \
    --label Publish-Permission
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 RemovePermission을 참조하세요.

set-endpoint-attributes

다음 코드 예시는 set-endpoint-attributes의 사용 방법을 보여 줍니다.

AWS CLI

엔드포인트 속성을 설정하는 방법

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..."
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 SetEndpointAttributes 섹션을 참조하세요.

set-platform-application-attributes

다음 코드 예시는 set-platform-application-attributes의 사용 방법을 보여 줍니다.

AWS CLI

플랫폼 애플리케이션 속성을 설정하는 방법

다음 set-platform-application-attributes 예제에서는 지정된 플랫폼 애플리케이션의 EventDeliveryFailure 속성을 지정된 Amazon SNS 주제의 ARN으로 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.

- API 세부 정보는 AWS CLI 명령 참조의 SetPlatformApplicationAttributes 섹션을 참조하세요.

set-sms-attributes

다음 코드 예시는 set-sms-attributes의 사용 방법을 보여 줍니다.

AWS CLI

SMS 메시지 속성을 설정하려면

다음 set-sms-attributes예제에서는 SMS 메시지의 기본 발신자 ID를 MyName으로 설정합니다.

```
aws sns set-sms-attributes \
```

```
    --attributes DefaultSenderID=MyName
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI명령 참조의 SetSMSAttributes를 참조하세요.

set-subscription-attributes

다음 코드 예시는 set-subscription-attributes의 사용 방법을 보여 줍니다.

AWS CLI

구독 속성을 설정하려면

다음 set-subscription-attributes예제에서는 SQS 구독에 RawMessageDelivery속성을 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-attributes예제에서는 SQS 구독에 FilterPolicy속성을 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-attributes예제에서는 SQS 구독에서 FilterPolicy속성을 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.

- API 세부 정보는 AWS CLI명령 참조의 SetSubscriptionAttributes를 참조하세요.

set-topic-attributes

다음 코드 예시는 set-topic-attributes의 사용 방법을 보여 줍니다.

AWS CLI

주제에 대한 속성을 설정하려면

다음 set-topic-attributes예제에서는 지정된 주제에 DisplayName속성을 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.

- API 세부 정보는 AWS CLI명령 참조의 SetTopicAttributes를 참조하세요.

subscribe

다음 코드 예시는 subscribe의 사용 방법을 보여 줍니다.

AWS CLI

주제를 구독하려면

다음 subscribe명령은 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 세부 정보는 AWS CLI명령 참조의 Subscribe를 참조하세요.

tag-resource

다음 코드 예시는 tag-resource의 사용 방법을 보여 줍니다.

AWS CLI

주제에 태그를 추가하려면

다음 tag-resource예제에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

unsubscribe

다음 코드 예시는 unsubscribe의 사용 방법을 보여 줍니다.

AWS CLI

주제 구독을 취소하려면

다음 unsubscribe예제에서는 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 세부 정보는 AWS CLI명령 참조의 Unsubscribe를 참조하세요.

untag-resource

다음 코드 예시는 untag-resource의 사용 방법을 보여 줍니다.

AWS CLI

주제에서 태그 제거

다음 untag-resource 예제에서는 지정된 Amazon SNS 주제에서 지정된 키가 있는 태그를 제거합니다.

```
aws sns untag-resource \
    --resource-arn arn:aws:sns:us-west-2:123456789012:MyTopic \
    --tag-keys Team
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

시나리오

푸시 알림에 대한 플랫폼 엔드포인트 생성

다음 코드 예제에서는 Amazon SNS 푸시 알림에 대한 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"
}
```

AWS CLI를 사용한 Amazon SQS 예시

다음 코드 예시는 Amazon SQS와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

add-permission

다음 코드 예시에서는 add-permission의 사용 방법을 보여줍니다.

AWS CLI

대기열에 권한 추가

이 예시에서는 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.
```

- API 세부 정보는 AWS CLI 명령 참조의 AddPermission을 참조하세요.

cancel-message-move-task

다음 코드 예시에서는 cancel-message-move-task의 사용 방법을 보여줍니다.

AWS CLI

메시지 이동 작업 취소

다음 cancel-message-move-task 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 CancelMessageMoveTask를 참조하세요.

change-message-visibility-batch

다음 코드 예시에서는 change-message-visibility-batch의 사용 방법을 보여줍니다.

AWS CLI

여러 메시지의 제한 시간 가시성 일괄 변경

이 예시에서는 지정된 2개 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"
    }
  ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [ChangeMessageVisibilityBatch](https://docs.aws.amazon.com/goto/aws-cli/sqs-2012-11-05/ChangeMessageVisibilityBatch)를 참조하세요.

`change-message-visibility`

다음 코드 예시에서는 `change-message-visibility`의 사용 방법을 보여줍니다.

AWS CLI

메시지의 제한 시간 가시성 변경

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.
```

- API 세부 정보는 AWS CLI 명령 참조의 [ChangeMessageVisibility](#)를 참조하세요.

`create-queue`

다음 코드 예시에서는 `create-queue`의 사용 방법을 보여줍니다.

AWS CLI

대기열 생성

이 예시에서는 지정된 이름의 대기열을 만들고, 메시지 보존 기간을 3일(3일 * 24시간 * 60분 * 60초)로 설정하고, 대기열의 Dead Letter Queue(DLQ)를 최대 메시지 수신 개수가 1,000개인 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [CreateQueue](#)를 참조하세요.

`delete-message-batch`

다음 코드 예시에서는 `delete-message-batch`의 사용 방법을 보여줍니다.

AWS CLI

여러 메시지 일괄 삭제

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"
    }
  ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteMessageBatch를 참조하세요.

`delete-message`

다음 코드 예시에서는 `delete-message`의 사용 방법을 보여줍니다.

AWS CLI

메시지 삭제

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.
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteMessage를 참조하세요.

`delete-queue`

다음 코드 예시에서는 `delete-queue`의 사용 방법을 보여줍니다.

AWS CLI

대기열 삭제

이 예시에서는 지정된 대기열을 삭제합니다.

명령:

```
aws sqs delete-queue --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyNewerQueue
```

출력:

```
None.
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteQueue를 참조하세요.

get-queue-attributes

다음 코드 예시에서는 get-queue-attributes의 사용 방법을 보여줍니다.

AWS CLI

대기열의 속성 가져오기

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"
  }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetQueueAttributes를 참조하세요.

get-queue-url

다음 코드 예시에서는 get-queue-url의 사용 방법을 보여줍니다.

AWS CLI

대기열 URL 가져오기

이 예시에서는 지정된 대기열의 URL을 가져옵니다.

명령:

```
aws sqs get-queue-url --queue-name MyQueue
```

출력:

```
{
  "QueueUrl": "https://queue.amazonaws.com/80398EXAMPLE/MyQueue"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetQueueUrl을 참조하세요.

list-dead-letter-source-queues

다음 코드 예시에서는 list-dead-letter-source-queues의 사용 방법을 보여줍니다.

AWS CLI

Dead Letter 소스 대기열 나열

이 예시에서는 지정된 Dead Letter 소스 대기열에 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"
  ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListDeadLetterSourceQueues를 참조하세요.

list-message-move-tasks

다음 코드 예시에서는 `list-message-move-tasks`의 사용 방법을 보여줍니다.

AWS CLI

메시지 이동 작업 나열

다음 `list-message-move-tasks` 예시에서는 지정된 대기열의 최근 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListMessageMoveTasks를 참조하세요.

list-queue-tags

다음 코드 예시에서는 `list-queue-tags`의 사용 방법을 보여줍니다.

AWS CLI

대기열의 모든 비용 할당 태그 나열

다음 `list-queue-tags` 예시에서는 지정된 대기열에 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListQueueTags를 참조하세요.

list-queues

다음 코드 예시에서는 `list-queues`의 사용 방법을 보여줍니다.

AWS CLI

대기열 나열

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"
  ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListQueues를 참조하세요.

purge-queue

다음 코드 예시에서는 purge-queue의 사용 방법을 보여줍니다.

AWS CLI

대기열 제거

이 예시에서는 지정된 대기열의 모든 메시지를 삭제합니다.

명령:

```
aws sqs purge-queue --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/
MyNewQueue
```

출력:

```
None.
```

- API 세부 정보는 AWS CLI 명령 참조의 PurgeQueue를 참조하세요.

receive-message

다음 코드 예시에서는 receive-message의 사용 방법을 보여줍니다.

AWS CLI

메시지 수신

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 예시에서는 다음으로 사용 가능한 메시지를 수신하여 SenderID 및 SentTimestamp 속성과 PostalCode 메시지 속성만 반환합니다.

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"
        }
      }
    }
  ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ReceiveMessage를 참조하세요.

`remove-permission`

다음 코드 예시에서는 `remove-permission`의 사용 방법을 보여줍니다.

AWS CLI

권한 제거

이 예시에서는 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.
```

- API 세부 정보는 AWS CLI 명령 참조의 RemovePermission을 참조하세요.

`send-message-batch`

다음 코드 예시에서는 `send-message-batch`의 사용 방법을 보여줍니다.

AWS CLI

여러 메시지 일괄 전송

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"
    }
  ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 SendMessageBatch를 참조하세요.

send-message

다음 코드 예시에서는 send-message의 사용 방법을 보여줍니다.

AWS CLI

메시지 전송

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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 SendMessage를 참조하세요.

`set-queue-attributes`

다음 코드 예시에서는 `set-queue-attributes`의 사용 방법을 보여줍니다.

AWS CLI

대기열 속성 설정

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 최대 메시지 수신 개수가 1,000개인 지정된 Dead Letter Queue(DLQ)를 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.
```

- API 세부 정보는 AWS CLI 명령 참조의 SetQueueAttributes를 참조하세요.

start-message-move-task

다음 코드 예시에서는 start-message-move-task의 사용 방법을 보여줍니다.

AWS CLI

예시 1: *메시지 이동 작업 시작*

다음 start-message-move-task 예시에서는 메시지 이동 작업을 시작하여 지정된 Dead Letter Queue(DLQ)에서 소스 대기열로 메시지를 이동시킵니다.

```
aws sqs start-message-move-task \
    --source-arn arn:aws:sqs:us-west-2:80398EXAMPLE:MyQueue
```

출력:

```
{
    "TaskHandle": "AQEB6nR4...HzlvZQ=="
}
```

자세한 내용은 안내서의 이름의 주제 제목을 참조하세요.

예시 2: *메시지 이동 작업을 최대 속도로 시작*

다음 start-message-move-task 예시에서는 메시지 이동 작업을 시작하여 지정된 Dead Letter Queue(DLQ)에서 지정된 대상 대기열로 메시지를 초당 최대 50개의 메시지 속도로 이동시킵니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 StartMessageMoveTask를 참조하세요.

tag-queue

다음 코드 예시에서는 tag-queue의 사용 방법을 보여줍니다.

AWS CLI

대기열에 비용 할당 태그 추가

다음 tag-queue 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 TagQueue를 참조하세요.

untag-queue

다음 코드 예시에서는 untag-queue의 사용 방법을 보여줍니다.

AWS CLI

대기열에서 비용 할당 태그 제거

다음 untag-queue 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 UntagQueue를 참조하세요.

AWS CLI를 사용한 Storage Gateway 예시

다음 코드 예시는 Storage Gateway와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

describe-gateway-information

다음 코드 예시에서는 describe-gateway-information의 사용 방법을 보여줍니다.

AWS CLI

게이트웨이 설명

다음 describe-gateway-information 명령은 지정된 게이트웨이에 대한 메타데이터를 반환합니다. 설명할 게이트웨이를 지정하려면 명령에서 게이트웨이의 Amazon 리소스 이름(ARN)을 사용합니다.

이 예시에서는 계정 sgw-12A3456B에 ID가 123456789012인 게이트웨이를 지정합니다.

```
aws storagegateway describe-gateway-information --gateway-
arn "arn:aws:storagegateway:us-west-2:123456789012:gateway/sgw-12A3456B"
```

이 명령은 게이트웨이의 이름, 네트워크 인터페이스, 구성된 시간대 및 상태(게이트웨이가 실행 중인지 여부) 등 게이트웨이에 대한 메타데이터를 반환합니다.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeGatewayInformation을 참조하세요.

list-file-shares

다음 코드 예시에서는 list-file-shares의 사용 방법을 보여줍니다.

AWS CLI

파일 공유 나열

다음 command-name 예시에서는 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 AWS Storage Gateway Service API 참조의 ListFileShares를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ListFileShares]를 참조하세요.

list-gateways

다음 코드 예시에서는 list-gateways의 사용 방법을 보여줍니다.

AWS CLI

계정의 게이트웨이 나열

다음 list-gateways 명령은 계정에 정의된 모든 게이트웨이를 나열합니다.

```
aws storagegateway list-gateways
```

이 명령은 게이트웨이의 Amazon 리소스 이름(ARN) 목록을 포함하는 JSON 블록을 출력합니다.

- API 세부 정보는 AWS CLI 명령 참조의 [ListGateways]를 참조하세요.

list-volumes

다음 코드 예시에서는 list-volumes의 사용 방법을 보여줍니다.

AWS CLI

게이트웨이에 대해 구성된 볼륨 나열

다음 list-volumes 명령은 지정된 게이트웨이에 대해 구성된 볼륨 목록을 반환합니다. 설명할 게이트웨이를 지정하려면 명령에서 게이트웨이의 Amazon 리소스 이름(ARN)을 사용합니다.

이 예시에서는 계정 sgw-12A3456B에 ID가 123456789012인 게이트웨이를 지정합니다.

```
aws storagegateway list-volumes --gateway-arn "arn:aws:storagegateway:us-west-2:123456789012:gateway/sgw-12A3456B"
```

이 명령은 각 볼륨의 유형과 ARN이 있는 볼륨 목록을 포함하는 JSON 블록을 출력합니다.

- API 세부 정보는 AWS CLI 명령 참조의 [ListVolumes]를 참조하세요.

refresh-cache

다음 코드 예시에서는 refresh-cache의 사용 방법을 보여줍니다.

AWS CLI

파일 공유 캐시 새로 고치기

다음 `refresh-cache` 예시에서는 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 AWS Storage Gateway Service API 참조의 ListFileShares를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RefreshCache를 참조하세요.

AWS CLI를 사용한 AWS STS 예시

다음 코드 예시에서는 AWS STS에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`assume-role-with-saml`

다음 코드 예시에서는 `assume-role-with-saml`의 사용 방법을 보여줍니다.

AWS CLI

SAML로 인증된 역할에 대한 단기 자격 증명 가져오기

다음 assume-role-with-saml 명령은 IAM 역할 TestSaml에 대한 단기 자격 증명 세트를 가져옵니다. 이 예시의 요청은 인증 시 ID 공급자가 제공한 SAML 어설션을 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 사용자 안내서의 임시 보안 자격 증명 요청을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AssumeRoleWithSaml을 참조하세요.

`assume-role-with-web-identity`

다음 코드 예시에서는 `assume-role-with-web-identity`의 사용 방법을 보여줍니다.

AWS CLI

웹 ID(OAuth 2.0)로 인증된 역할에 대한 단기 자격 증명 가져오기

다음 `assume-role-with-web-identity` 명령은 IAM 역할 app1에 대한 단기 자격 증명 세트를 가져옵니다. 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 사용자 안내서의 임시 보안 자격 증명 요청을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AssumeRoleWithWebIdentity를 참조하세요.

assume-role

다음 코드 예시에서는 assume-role의 사용 방법을 보여줍니다.

AWS CLI

역할 수임

다음 assume-role 명령은 IAM 역할 s3-access-example에 대한 단기 자격 증명 세트를 가져옵니다.

```
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 AWS 인증에 사용할 수 있는 액세스 키, 시크릿 키 및 세션 토큰이 포함됩니다.

AWS CLI를 사용하는 경우 역할과 연결된 이름이 지정된 프로파일을 설정할 수 있습니다. 프로파일을 사용하면 AWS CLI에서 assume-role을 호출하고 대신 자격 증명을 관리합니다. 자세한 내용은 AWS IAM 사용자 안내서의 AWS CLI에서 IAM 역할 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AssumeRole을 참조하세요.

`assume-root`

다음 코드 예시는 `assume-root`의 사용 방법을 보여 줍니다.

AWS CLI

권한 있는 세션을 시작하려면

다음 `assume-root` 명령은 조직의 멤버 계정에 대해 잘못 구성된 Amazon S3 버킷 정책을 제거하는 데 사용할 수 있는 단기 자격 증명 세트를 검색합니다.

```
aws sts assume-root \
    --duration-seconds 900 \
    --target-principal 111122223333 \
    --task-policy-arn arn=arn:aws:iam::aws:policy/root-task/S3UnlockBucketPolicy
```

출력:

```
{
    "Credentials": {
        "SecretAccessKey": "9drTJvcXLB89EXAMPLELB8923FB892xMFI",
        "SessionToken": "AQoXdzELDDY//////////
wEaoAK1wvxJY12r2IrDFT2IvAzTCn3zHoZ7YNtpiQLF0MqZye/qwjzP2iEXAMPLEbw/
m3hsj8VBTkPORGvr9jM5sgP+w9IZWZnU+LWhmg
+a5fDi2oTGUYcdg9uexQ4mtCHIHfi4citgqZTgco40Yqr4lIlo4V2b2Dyauk0eYFNebHtYlFVgAUj
+7Indz3LU0aTWk1WKIjHmmMCIoTkyYp/k7kUG7moeEYKSitwQIi6Gjn+nyzM
+PtoA3685ixzv0R7i5rjQi0YE0lf1oeie3bDiNHncmzosRM6SFiPzSvp6h/32xQuZsjcypmwsPSDtTPYcs0+YN/8BRi2
IcrxSpnWEXAMPLEXSDFTAQAM6Dl9zR0tXoybnlrZIwMLlMi1Kcgo5OytwU=",
        "Expiration": "2024-11-15T00:05:07Z",
        "AccessKeyId": "ASIAJEXAMPLEXEG2JICEA"
```

```
    },
    "SourceIdentity": "Alice",
}
```

명령의 출력에는 멤버 계정에서 권한 있는 작업을 수행하는 데 사용할 수 있는 액세스 키, 시크릿 키 및 세션 토큰이 포함됩니다. 자세한 내용은 AWS IAM 사용 설명서의 AWS Organizations 멤버 계정에서 권한 있는 작업 수행을 참조하세요.

- API 세부 정보는 AWS CLI명령 참조의 AssumeRoot를 참조하세요.

decode-authorization-message

다음 코드 예시는 decode-authorization-message의 사용 방법을 보여 줍니다.

AWS CLI

요청에 대한 응답으로 반환된 인코딩된 인증 메시지 디코딩

다음 decode-authorization-message 예시에서는 Amazon Web Services 요청에 대한 응답으로 반환되는 인코딩 메시지에서 받은 요청의 권한 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 사용자 안내서의 정책 평가 로직을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DecodeAuthorizationMessage를 참조하세요.

get-caller-identity

다음 코드 예시에서는 `get-caller-identity`의 사용 방법을 보여줍니다.

AWS CLI

현재 IAM ID의 세부 정보 가져오기

다음 `get-caller-identity` 명령은 요청을 인증하는 데 사용되는 IAM 자격 증명의 정보를 표시합니다. 호출자는 IAM 사용자입니다.

```
aws sts get-caller-identity
```

출력:

```
{
    "UserId": "AIDASAMPLEUSERID",
    "Account": "123456789012",
    "Arn": "arn:aws:iam::123456789012:user/DevAdmin"
```

```
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetCallerIdentity를 참조하세요.

get-federation-token

다음 코드 예시에서는 get-federation-token의 사용 방법을 보여줍니다.

AWS CLI

IAM 사용자 액세스 키 자격 증명을 사용하여 임시 보안 자격 증명 세트 반환

다음 get-federation-token 예시에서는 사용자의 임시 보안 자격 증명 세트(액세스 키 ID, 시크릿 액세스 키 및 보안 토큰으로 구성)를 반환합니다. IAM 사용자의 장기 보안 자격 증명을 사용하여 GetFederationToken 작업을 직접적으로 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//////////
wEQAxoMNDUyOTI1MTcwNTA3IgxuywAOpuuoLsk3MJwqgQPg8QOd9HuoClUxq26wnc/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 사용자 안내서의 임시 보안 자격 증명 요청을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetFederationToken을 참조하세요.

get-session-token

다음 코드 예시에서는 `get-session-token`의 사용 방법을 보여줍니다.

AWS CLI

IAM ID용 단기 자격 증명 세트 가져오기

다음 `get-session-token` 명령은 직접 호출을 위한 IAM ID용 단기 자격 증명 세트를 가져옵니다. 정책에 따라 다중 인증(MFA)이 필요한 경우 요청에 이 자격 증명을 사용할 수 있습니다. 자격 증명은 생성 후 15분 뒤에 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 사용자 안내서의 [임시 보안 자격 증명 요청](#)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [GetSessionToken](#)을 참조하세요.

AWS CLI를 사용한 지원 예시

다음 코드 예시에서는 지원에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

add-attachments-to-set

다음 코드 예시에서는 add-attachments-to-set의 사용 방법을 보여줍니다.

AWS CLI

세트에 첨부 파일 추가

다음 add-attachments-to-set 예시에서는 AWS 계정의 지원 사례에 지정할 수 있는 이미지를 세트에 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 사용자 안내서의 사례 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AddAttachmentsToSet를 참조하세요.

add-communication-to-case

다음 코드 예시에서는 add-communication-to-case의 사용 방법을 보여줍니다.

AWS CLI

사례에 커뮤니케이션 추가

다음 add-communication-to-case 예시에서는 AWS 계정의 지원 사례에 커뮤니케이션을 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 사용자 안내서의 사례 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AddCommunicationToCase를 참조하세요.

`create-case`

다음 코드 예시에서는 `create-case`의 사용 방법을 보여줍니다.

AWS CLI

사례 생성

다음 `create-case` 예시에서는 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 사용자 안내서의 사례 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateCase를 참조하세요.

describe-attachment

다음 코드 예시에서는 describe-attachment의 사용 방법을 보여줍니다.

AWS CLI

첨부 파일 설명

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 사용자 안내서의 사례 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeAttachment를 참조하세요.

describe-cases

다음 코드 예시에서는 describe-cases의 사용 방법을 보여줍니다.

AWS CLI

사례 설명

다음 describe-cases 예시에서는 AWS 계정의 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 사용자 안내서의 사례 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeCases를 참조하세요.

describe-communications

다음 코드 예시에서는 describe-communications의 사용 방법을 보여줍니다.

AWS CLI

사례에 대한 최근 커뮤니케이션 설명

다음 describe-communications 예시에서는 AWS 계정의 지정된 지원 사례에 대한 최근 커뮤니케이션을 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": "eyJuZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQEXAMPLE=="
}
```

자세한 내용은 AWS Support 사용자 안내서의 사례 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeCommunications를 참조하세요.

describe-services

다음 코드 예시에서는 describe-services의 사용 방법을 보여줍니다.

AWS CLI

AWS 서비스 및 서비스 범주 나열

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 사용자 안내서의 사례 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeServices를 참조하세요.

`describe-severity-levels`

다음 코드 예시에서는 `describe-severity-levels`의 사용 방법을 보여줍니다.

AWS CLI

사용 가능한 심각도 수준 나열

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 사용자 안내서의 심각도 선택을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeSeverityLevels를 참조하세요.

`describe-trusted-advisor-check-refresh-statuses`

다음 코드 예시에서는 `describe-trusted-advisor-check-refresh-statuses`의 사용 방법을 보여줍니다.

AWS CLI

AWS Trusted Advisor 검사의 새로 고침 상태 나열

다음 `describe-trusted-advisor-check-refresh-statuses` 예시에서는 Amazon S3 버킷 권한 및 IAM 사용이라는 두 가지 Trusted Advisor 검사의 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 Support 사용자 안내서의 AWS Trusted Advisor를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeTrustedAdvisorCheckRefreshStatuses를 참조하세요.

`describe-trusted-advisor-check-result`

다음 코드 예시에서는 `describe-trusted-advisor-check-result`의 사용 방법을 보여줍니다.

AWS CLI

AWS Trusted Advisor 검사 결과 나열

다음 `describe-trusted-advisor-check-result` 예시에서는 IAM 사용 검사 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 Support 사용자 안내서의 AWS Trusted Advisor를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeTrustedAdvisorCheckResult를 참조하세요.

describe-trusted-advisor-check-summaries

다음 코드 예시에서는 describe-trusted-advisor-check-summaries의 사용 방법을 보여줍니다.

AWS CLI

AWS Trusted Advisor 검사 요약 나열

다음 describe-trusted-advisor-check-summaries 예시에서는 Amazon S3 버킷 권한 및 IAM 사용이라는 두 가지 Trusted Advisor 검사의 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 Support 사용자 안내서의 AWS Trusted Advisor를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeTrustedAdvisorCheckSummaries를 참조하세요.

describe-trusted-advisor-checks

다음 코드 예시에서는 `describe-trusted-advisor-checks`의 사용 방법을 보여줍니다.

AWS CLI

사용 가능한 AWS Trusted Advisor 검사 나열

다음 `describe-trusted-advisor-checks` 예시에서는 AWS 계정에서 사용 가능한 Trusted Advisor 검사를 나열합니다. 이 정보에는 검사 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 Support 사용자 안내서의 AWS Trusted Advisor를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeTrustedAdvisorChecks를 참조하세요.

refresh-trusted-advisor-check

다음 코드 예시에서는 `refresh-trusted-advisor-check`의 사용 방법을 보여줍니다.

AWS CLI

AWS Trusted Advisor 검사 새로 고침

다음 `refresh-trusted-advisor-check` 예시에서는 AWS 계정의 Amazon S3 버킷 권한 Trusted Advisor 검사를 새로 고칩니다.

```
aws support refresh-trusted-advisor-check \
    --check-id "Pfx0RwqBli"
```

출력:

```
{
    "status": {
        "checkId": "Pfx0RwqBli",
        "status": "enqueued",
        "millisUntilNextRefreshable": 3599992
```

```
    }
}
```

자세한 내용은 AWS Support 사용자 안내서의 AWS Trusted Advisor를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RefreshTrustedAdvisorCheck을 참조하세요.

`resolve-case`

다음 코드 예시에서는 `resolve-case`의 사용 방법을 보여줍니다.

AWS CLI

지원 사례 해결

다음 `resolve-case` 예시에서는 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 사용자 안내서의 사례 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ResolveCase를 참조하세요.

AWS CLI를 사용한 Amazon SWF 예시

다음 코드 예시는 Amazon SWF와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

count-closed-workflow-executions

다음 코드 예시에서는 count-closed-workflow-executions의 사용 방법을 보여줍니다.

AWS CLI

종료된 워크플로 실행 수 계산

swf count-closed-workflow-executions를 사용하면 지정된 도메인에서 종료된 워크플로 실행 수를 가져올 수 있습니다. 필터를 지정하면 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
}
```

'truncated'가 true이면 'count'는 Amazon SWF에서 반환할 수 있는 최대 수를 나타냅니다. 추가 결과는 잘립니다.

반환되는 결과 수를 줄이려면 다음을 수행할 수 있습니다.

--close-time-filter 또는 --start-time-filter 값을 수정하여 검색되는 시간 범위를 좁힙니다. 각 항목은 서로 함께 사용할 수 없습니다. 요청에서 이 중 하나만 지정할 수 있습니다. --close-status-filter, --execution-filter, --tag-filter 또는 --type-filter 인수

를 사용하면 결과를 추가로 필터링할 수 있습니다. 단, 이러한 인수는 서로 함께 사용할 수 없습니다.

또한 Amazon Simple Workflow Service API 참조의 CountClosedWorkflowExecutions도 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CountClosedWorkflowExecutions를 참조하세요.

count-open-workflow-executions

다음 코드 예시에서는 `count-open-workflow-executions`의 사용 방법을 보여줍니다.

AWS CLI

진행 중인 워크플로 실행 수 계산

`swf count-open-workflow-executions`를 사용하여 지정된 도메인에서 진행 중인 워크플로 실행 수를 가져올 수 있습니다. 필터를 지정하면 특정 실행 클래스를 계산할 수 있습니다.

`--domain` 및 `--start-time-filter` 인수는 필수입니다. 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
}
```

'truncated'가 `true`이면 'count'는 Amazon SWF에서 반환할 수 있는 최대 수를 나타냅니다. 추가 결과는 잘립니다.

반환되는 결과 수를 줄이려면 다음을 수행할 수 있습니다.

`--start-time-filter` 값을 수정하여 검색되는 시간 범위를 좁히고, `--close-status-filter`, `--execution-filter`, `--tag-filter` 또는 `--type-filter` 인수를 사용하여 결과를 추가로 필터링할 수 있습니다. 각 항목은 서로 함께 사용할 수 없습니다. 요청에서 이 중 하나만 지정할 수 있습니다.

자세한 내용은 Amazon Simple Workflow Service API 참조의 CountOpenWorkflowExecutions를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CountOpenWorkflowExecutions를 참조하세요.

deprecate-domain

다음 코드 예시에서는 deprecate-domain의 사용 방법을 보여줍니다.

AWS CLI

도메인 사용 중지

도메인을 사용 중지하려면(도메인을 여전히 볼 수 있지만 해당 도메인에서 새 워크플로 실행을 생성하거나 유형을 등록할 수 없음) swf deprecate-domain을 사용합니다. 필수 파라미터가 하나 있으며 --name입니다. 이 파라미터는 사용 중지할 도메인의 이름을 가져옵니다.

```
aws swf deprecate-domain \
    --name MyNeatNewDomain ""
```

register-domain을 사용했을 때와 마찬가지로 출력이 반환되지 않습니다. list-domains를 사용하여 등록된 도메인을 확인하면 해당 도메인이 사용 중지되어 더 이상 반환된 데이터에 표시되지 않는 것을 볼 수 있습니다.

```
aws swf list-domains \
    --registration-status REGISTERED
        {
            "domainInfos": [
                {
                    "status": "REGISTERED",
                    "name": "DataFrobotz"
                },
                {
                    "status": "REGISTERED",
                    "name": "erontest"
                }
            ]
        }
```

list-domains와 함께 --registration-status DEPRECATED를 사용하면 더 이상 사용되지 않는 도메인이 표시됩니다.

```
aws swf list-domains \
    --registration-status DEPRECATED
        {
            "domainInfos": [
                {
                    "status": "DEPRECATED",
                    "name": "MyNeatNewDomain"
                }
            ]
        }
```

또한 `describe-domain`을 사용하여 사용 중지된 도메인의 정보를 가져올 수 있습니다.

```
aws swf describe-domain \
    --name MyNeatNewDomain
        {
            "domainInfo": {
                "status": "DEPRECATED",
                "name": "MyNeatNewDomain"
            },
            "configuration": {
                "workflowExecutionRetentionPeriodInDays": "0"
            }
        }
```

또한 Amazon Simple Workflow Service API 참조의 DeprecateDomain도 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeprecateDomain을 참조하세요.

describe-domain

다음 코드 예시에서는 `describe-domain`의 사용 방법을 보여줍니다.

AWS CLI

도메인 정보 가져오기

특정 도메인의 세부 정보를 가져오려면 `swf describe-domain`을 사용합니다. 필수 파라미터가 하나 있으며 `--name`입니다. 이 파라미터는 정보를 확인할 도메인의 이름을 가져옵니다.

```
aws swf describe-domain \
    --name DataFrobotz
```

```
        {
            "domainInfo": {
                "status": "REGISTERED",
                "name": "DataFrobotz"
            },
            "configuration": {
                "workflowExecutionRetentionPeriodInDays": "1"
            }
        }
```

또한 describe-domain을 사용하여 사용 중지된 도메인의 정보를 가져올 수 있습니다.

```
aws swf describe-domain \
    --name MyNeatNewDomain
        {
            "domainInfo": {
                "status": "DEPRECATED",
                "name": "MyNeatNewDomain"
            },
            "configuration": {
                "workflowExecutionRetentionPeriodInDays": "0"
            }
        }
```

또한 Amazon Simple Workflow Service API 참조의 DescribeDomain도 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDomain을 참조하세요.

list-activity-types

다음 코드 예시에서는 list-activity-types의 사용 방법을 보여줍니다.

AWS CLI

활동 유형 나열

도메인의 활동 유형 목록을 가져오려면 swf list-activity-types를 사용합니다. --domain 및 --registration-status 인수는 필수입니다.

```
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 사용하면 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"
        }
    ]
}
```

페이지의 결과를 가져오려면 --maximum-page-size 인수를 설정하면 됩니다. 한 결과 페이지를 넘어가는 더 많은 결과가 반환되면 결과 세트에 'nextPageToken'이 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 list-activity-types에 대한 다음 번 직접적 호출에 nextPageToken 값을 전달하면 결과의 다음 페이지를 가져올 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": "AAAAKgAAAAEAAAAAAAAAAAw+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 'nextPageToken'이 결과와 함께 반환됩니다. 더 이상 가져올 결과 페이지가 없으면 'nextPageToken'이 결과 세트에 반환되지 않습니다.

`--reverse-order` 인수를 사용하면 반환된 결과의 순서를 반대로 만들 수 있습니다. 이는 페이지 지정된 결과에도 영향을 미칩니다.

```
aws swf list-activity-types \
    --domain DataFrobtzz \
    --registration-status REGISTERED \
    --maximum-page-size 2 \
    --reverse-order
```

출력:

```
{
    "nextPageToken": "AAAAKgAAAAEAAAAAAAAAAAwXcpu5ePSyQkrC
+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"
        }
    ]
}
```

또한 Amazon Simple Workflow Service API 참조의 ListActivityTypes도 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListActivityTypes를 참조하세요.

list-domains

다음 코드 예시에서는 list-domains의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 등록된 도메인 나열

다음 list-domains 명령 예시에서는 계정에 등록된 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 참조의 ListDomains를 참조하세요.

예시 2: 사용 중지된 도메인 나열

다음 `list-domains` 명령 예시에서는 계정에 등록된 DEPRECATED SWF 도메인을 나열합니다. 사용 중지된 도메인은 새 워크플로 또는 활동을 등록할 수 없지만 여전히 쿼리는 받을 수 있는 도메인입니다.

```
aws swf list-domains \
    --registration-status DEPRECATED
```

출력:

```
{
  "domainInfos": [
    {
      "status": "DEPRECATED",
      "name": "MyNeatNewDomain"
    }
  ]
}
```

자세한 내용은 Amazon Simple Workflow Service API 참조의 ListDomains를 참조하세요.

예시 3: 등록된 도메인의 첫 페이지 나열

다음 `list-domains` 명령 예시에서는 `--maximum-page-size` 옵션을 통해 계정에 등록된 REGISTERED SWF 도메인의 첫 페이지를 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 참조의 ListDomains를 참조하세요.

예시 4: 등록된 도메인의 지정된 단일 페이지 나열

다음 `list-domains` 명령 예시에서는 `--maximum-page-size` 옵션을 통해 계정에 등록된 REGISTERED SWF 도메인의 첫 페이지를 나열합니다.

`nextPageToken` 인수에 `--next-page-token` 값을 입력하여 다시 직접적으로 호출하면 다른 결과 페이지가 나옵니다.

```
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 `nextPageToken`이 결과에 반환됩니다.

자세한 내용은 Amazon Simple Workflow Service API 참조의 ListDomains를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListDomains를 참조하세요.

`list-workflow-types`

다음 코드 예시에서는 `list-workflow-types`의 사용 방법을 보여줍니다.

AWS CLI

워크플로 유형 나열

도메인의 워크플로 유형 목록을 가져오려면 `swf list-workflow-types`를 사용합니다. `--domain` 및 `--registration-status` 인수는 필수입니다. 다음은 간단한 예시입니다.

```
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"
            }
        }
    ]
}
```

list-activity-types를 사용할 때와 마찬가지로 --name 인수를 사용하여 특정 이름의 워크플로 유형만 선택하고 --next-page-token과 함께 --maximum-page-size 인수를 사용하여 결과를 페이지 지정할 수 있습니다. 결과가 반환되는 순서를 반대로 하려면 --reverse-order를 사용합니다.

또한 Amazon Simple Workflow Service API 참조의 ListWorkflowTypes도 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListWorkflowTypes를 참조하세요.

register-domain

다음 코드 예시에서는 register-domain의 사용 방법을 보여줍니다.

AWS CLI

도메인 등록

AWS CLI를 사용하여 새 도메인을 등록할 수 있습니다. swf register-domain 명령을 사용합니다. 필수 파라미터가 두 개 있습니다. --name은 도메인 이름을 가져오고 --workflow-execution-retention-period-in-days는 이 도메인에서 워크플로 실행 데이터를 유지하는 기간(일)을 최대 90일로 지정하는 정수를 가져옵니다(자세한 내용은 SWF FAQ<https://aws.amazon.com/swf/faqs/#retain_limit> 참조). 지정된 기간이 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 swf describe-domain을 사용하면 새 도메인을 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"

  }
}
```

또한 Amazon Simple Workflow Service API 참조의 RegisterDomain도 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RegisterDomain을 참조하세요.

register-workflow-type

다음 코드 예시에서는 register-workflow-type의 사용 방법을 보여줍니다.

AWS CLI

워크플로 유형 등록

AWS CLI에 워크플로 유형을 등록하려면 swf register-workflow-type 명령을 사용합니다.

```
aws swf register-workflow-type \
    --domain DataFrobtzz \
    --name "MySimpleWorkflow" \
    --workflow-version "v1"
```

이 명령은 성공 시 출력을 생성하지 않습니다.

오류(예: 동일한 워크플로 유형을 2회 등록하거나 존재하지 않는 도메인을 지정하려는 경우)가 발생하면 JSON에서 응답을 받게 됩니다.

```
{
    "message": "WorkflowType=[name=MySimpleWorkflow, version=v1]",
    "__type": "com.amazonaws.swf.base.model#TypeAlreadyExistsFault"
}
```

`--domain`, `--name` 및 `--workflow-version`은 필수입니다. 워크플로 설명, 제한 시간 및 하위 워크플로 정책을 설정할 수도 있습니다.

자세한 내용은 Amazon Simple Workflow Service API 참조의 ListDomains를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RegisterWorkflowType을 참조하세요.

AWS CLI를 사용한 Systems Manager 예시

다음 코드 예제는 Systems Manager와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`add-tags-to-resource`

다음 코드 예시는 `add-tags-to-resource`의 사용 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 추가하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 AddTagsToResource를 참조하세요.

`associate-ops-item-related-item`

다음 코드 예시는 `associate-ops-item-related-item`의 사용 방법을 보여 줍니다.

AWS CLI

관련 항목을 연결하는 방법

다음 `associate-ops-item-related-item` 예시에서는 OpsItem에 관련 항목을 연결합니다.

```
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 OpsCenter에서 Incident Manager 인스턴스 작업 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AssociateOpsItemRelatedItem 섹션을 참조하세요.

`cancel-command`

다음 코드 예시는 `cancel-command`의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 CancelCommand를 참조하세요.

cancel-maintenance-window-execution

다음 코드 예시는 cancel-maintenance-window-execution의 사용 방법을 보여 줍니다.

AWS CLI

유지 관리 기간 실행을 취소하는 방법

이 cancel-maintenance-window-execution 예시는 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 AWS Systems Manager 사용 설명서의 Systems Manager 유지 관리 기간 자습서 (AWS CLI)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CancelMaintenanceWindowExecution 섹션을 참조하세요.

create-activation

다음 코드 예시는 create-activation의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateActivation을 참조하세요.

`create-association-batch`

다음 코드 예시는 `create-association-batch`의 사용 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": []
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateAssociationBatch를 참조하세요.

`create-association`

다음 코드 예시는 `create-association`의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 인스턴스 ID를 사용하여 문서를 연결하는 방법

이 예제에서는 인스턴스 ID를 사용하여 구성 문서를 인스턴스와 연결합니다.

```
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 AWS Systems Manager API 참조의 CreateAssociation을 참조하세요.

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 AWS Systems Manager API 참조의 CreateAssociation을 참조하세요.

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 AWS Systems Manager API 참조의 CreateAssociation 또는 AWS Systems Manager 사용 설명서의 참조: Systems Manager의 Cron 및 Rate 표현식을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateAssociation을 참조하세요.

`create-document`

다음 코드 예시는 `create-document`의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateDocument를 참조하세요.

`create-maintenance-window`

다음 코드 예시는 `create-maintenance-window`의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateMaintenanceWindow를 참조하세요.

create-ops-item

다음 코드 예시는 `create-ops-item`의 사용 방법을 보여 줍니다.

AWS CLI

OpsItems를 생성하는 방법

다음 `create-ops-item` 예시에서는 OperationalData의 /aws/resources 키를 사용하여 Amazon DynamoDB 관련 리소스가 있는 OpsItem을 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 Creating OpsItems를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateOpsItem을 참조하세요.

create-patch-baseline

다음 코드 예시는 `create-patch-baseline`의 사용 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 규칙을 사용하여 패치 기준을 생성하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 CreatePatchBaseline을 참조하세요.

`create-resource-data-sync`

다음 코드 예시는 `create-resource-data-sync`의 사용 방법을 보여 줍니다.

AWS CLI

리소스 데이터 동기화 생성

이 예시에서는 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"
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateResourceDataSync 섹션을 참조하세요.

delete-activation

다음 코드 예시는 delete-activation의 사용 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 환경에 대한 AWS Systems Manager 설정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteActivation을 참조하세요.

delete-association

다음 코드 예시는 delete-association의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteAssociation을 참조하세요.

delete-document

다음 코드 예시는 delete-document의 사용 방법을 보여 줍니다.

AWS CLI

문서 삭제

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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteDocument 섹션을 참조하세요.

delete-inventory

다음 코드 예시는 delete-inventory의 사용 방법을 보여 줍니다.

AWS CLI

사용자 지정 인벤토리 유형을 삭제하는 방법

이 예시에서는 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 비활성화하는 방법

이 예시에서는 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": []
  }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteInventory 섹션을 참조하세요.

delete-maintenance-window

다음 코드 예시는 delete-maintenance-window의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteMaintenanceWindow를 참조하세요.

delete-parameter

다음 코드 예시는 delete-parameter의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteParameter를 참조하세요.

delete-parameters

다음 코드 예시는 delete-parameters의 사용 방법을 보여 줍니다.

AWS CLI

파라미터 목록 삭제

다음 delete-parameters 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteParameters 섹션을 참조하세요.

delete-patch-baseline

다음 코드 예시는 delete-patch-baseline의 사용 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 AWSSystems Manager 사용 설명서의 패치 기준 업데이트 또는 삭제(콘솔)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeletePatchBaseline을 참조하세요.

delete-resource-data-sync

다음 코드 예시는 delete-resource-data-sync의 사용 방법을 보여 줍니다.

AWS CLI

리소스 데이터 동기화 삭제

이 예시에서는 리소스 데이터 동기화를 삭제합니다. 명령이 성공해도 출력은 없습니다.

명령:

```
aws ssm delete-resource-data-sync --sync-name "ssm-resource-data-sync"
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteResourceDataSync 섹션을 참조하세요.

deregister-managed-instance

다음 코드 예시는 deregister-managed-instance의 사용 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 및 멀티클라우드 환경에서 관리형 인스턴스 등록 취소를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeregisterManagedInstance를 참조하세요.

deregister-patch-baseline-for-patch-group

다음 코드 예시는 deregister-patch-baseline-for-patch-group의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 [DeregisterPatchBaselineForPatchGroup]을 참조하세요.

deregister-target-from-maintenance-window

다음 코드 예시는 deregister-target-from-maintenance-window의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 DeregisterTargetFromMaintenanceWindow를 참조하세요.

deregister-task-from-maintenance-window

다음 코드 예시는 deregister-task-from-maintenance-window의 사용 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 AWS Systems Manager 사용 설명서의 Systems Manager 유지 관리 기간 자습서 (AWS CLI)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeregisterTaskFromMaintenanceWindow를 참조하세요.

describe-activations

다음 코드 예시는 describe-activations의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeActivations를 참조하세요.

describe-association-execution-targets

다음 코드 예시는 describe-association-execution-targets의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeAssociationExecutionTargets를 참조하세요.

describe-association-executions

다음 코드 예시는 describe-association-executions의 사용 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 보기]를 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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeAssociationExecutions를 참조하세요.

describe-association

다음 코드 예시는 describe-association의 사용 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 생성]을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeAssociation을 참조하세요.

describe-automation-executions

다음 코드 예시는 describe-automation-executions의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeAutomationExecutions를 참조하세요.

describe-automation-step-executions

다음 코드 예시는 describe-automation-step-executions의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeAutomationStepExecutions를 참조하세요.

describe-available-patches

다음 코드 예시는 describe-available-patches의 사용 방법을 보여 줍니다.

AWS CLI

사용 가능한 패치를 가져오는 방법

다음 describe-available-patches 예제에서는 MSRC 심각도가 위험인 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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeAvailablePatches를 참조하세요.

describe-document-permission

다음 코드 예시는 describe-document-permission의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDocumentPermission을 참조하세요.

describe-document

다음 코드 예시는 describe-document의 사용 방법을 보여 줍니다.

AWS CLI

문서 세부 정보를 표시하는 방법

다음 describe-document 예제에서는 사용자 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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDocument를 참조하세요.

`describe-effective-instance-associations`

다음 코드 예시는 `describe-effective-instance-associations`의 사용 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeEffectiveInstanceAssociations를 참조하세요.

describe-effective-patches-for-patch-baseline

다음 코드 예시는 describe-effective-patches-for-patch-baseline의 사용 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 기준으로 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 기준에서 정의한 모든 패치를 가져오는 방법

다음 `describe-effective-patches-for-patch-baseline` 예제에서는 AWS 관리형 패치 기준으로 정의된 패치를 반환합니다. AWS 관리형 기준의 경우 `--baseline-id`에는 전체 기준 ARN이 필요합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeEffectivePatchesForPatchBaseline을 참조하세요.

describe-instance-associations-status

다음 코드 예시는 describe-instance-associations-status의 사용 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeInstanceAssociationsStatus를 참조하세요.

`describe-instance-information`

다음 코드 예시는 `describe-instance-information`의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeInstanceInformation을 참조하세요.

`describe-instance-patch-states-for-patch-group`

다음 코드 예시는 `describe-instance-patch-states-for-patch-group`의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeInstancePatchStatesForPatchGroup을 참조하세요.

describe-instance-patch-states

다음 코드 예시는 describe-instance-patch-states의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeInstancePatchStates를 참조하세요.

describe-instance-patches

다음 코드 예시는 describe-instance-patches의 사용 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: 인스턴스의 지정된 설치 시간 이후에 설치된 패치 목록을 가져오는 방법

다음 describe-instance-patches 예제에서는 --filters 및 --query의 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--"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeInstancePatches를 참조하세요.

describe-inventory-deletions

다음 코드 예시는 describe-inventory-deletions의 사용 방법을 보여 줍니다.

AWS CLI

인벤토리 삭제를 가져오는 방법

이 예시에서는 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 가져오는 방법

이 예시에서는 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
      }
  ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeInventoryDeletions 섹션을 참조하세요.

describe-maintenance-window-execution-task-invocations

다음 코드 예시는 describe-maintenance-window-execution-task-invocations의 사용 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 작업 및 작업 실행에 대한 정보 보기(AWS CLI)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeMaintenanceWindowExecutionTaskInvocations를 참조하세요.

describe-maintenance-window-execution-tasks

다음 코드 예시는 describe-maintenance-window-execution-tasks의 사용 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 작업 및 작업 실행에 대한 정보 보기(AWS CLI)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeMaintenanceWindowExecutionTasks를 참조하세요.

describe-maintenance-window-executions

다음 코드 예시는 describe-maintenance-window-executions의 사용 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 작업 및 작업 실행에 대한 정보 보기(AWS CLI)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeMaintenanceWindowExecutions를 참조하세요.

describe-maintenance-window-schedule

다음 코드 예시는 describe-maintenance-window-schedule의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 유지 관리 기간의 모든 실행 나열

다음 describe-maintenance-window-schedule 예시에서는 지정된 유지 관리 기간의 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"
        },
        ...
    ]
}
```

예시 2: 지정된 날짜 이전의 유지 관리 기간에 예정된 모든 실행 나열

다음 describe-maintenance-window-schedule 예시에서는 지정된 유지 관리 기간 동안 지정된 날짜 이전에 발생하는 모든 예정된 실행을 나열합니다.

```
aws ssm describe-maintenance-window-schedule \
    --window-id mw-0ecb1226dd7b2e9a6 \
    --filters "Key=ScheduledBefore,Values=2020-02-15T06:00:00Z"
```

예시 3: 지정된 날짜 이후 유지 관리 기간에 대한 모든 실행 나열

다음 describe-maintenance-window-schedule 예시에서는 지정된 유지 관리 기간 동안 지정된 날짜 이후에 발생하는 모든 예정된 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 기간에 대한 정보 보기(AWS CLI)](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeMaintenanceWindowSchedule](#) 섹션을 참조하세요.

describe-maintenance-window-targets

다음 코드 예시는 describe-maintenance-window-targets의 사용 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 기간에 대한 정보 보기(AWS CLI)]를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeMaintenanceWindowTargets]를 참조하세요.

describe-maintenance-window-tasks

다음 코드 예시는 `describe-maintenance-window-tasks`의 사용 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 명령 문서를 간접 호출하는 유지 관리 기간에 대한 모든 작업을 나열하는 방법

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 `Priority`인 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 1이 Priority이고 Run Command를 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 기간에 대한 정보 보기(AWS CLI)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeMaintenanceWindowTasks를 참조하세요.

describe-maintenance-windows-for-target

다음 코드 예시는 describe-maintenance-windows-for-target의 사용 방법을 보여 줍니다.

AWS CLI

특정 인스턴스와 연결된 모든 유지 관리 기간을 나열하는 방법

다음 describe-maintenance-windows-for-target 예시에서는 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 기간에 대한 정보 보기(AWS CLI)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeMaintenanceWindowsForTarget 섹션을 참조하세요.

describe-maintenance-windows

다음 코드 예시는 describe-maintenance-windows의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 모든 유지 관리 기간을 나열하는 방법

다음 describe-maintenance-windows 예제에서는 현재 리전 내 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 기간에 대한 정보 보기(AWS CLI)](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeMaintenanceWindows](#)를 참조하세요.

describe-ops-items

다음 코드 예시는 describe-ops-items의 사용 방법을 보여 줍니다.

AWS CLI

OpsItems 세트를 나열하는 방법

다음 describe-ops-items 예시에서는 AWS 계정에서 열려 있는 모든 OpsItems의 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 설명서의 OpsItems 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeOpsItems](https://docs.aws.amazon.com/cli/latest/reference/ssm/describe-ops-items.html)를 참조하세요.

describe-parameters

다음 코드 예시는 describe-parameters의 사용 방법을 보여 줍니다.

AWS CLI

예 1: 모든 파라미터를 나열하는 방법

다음 describe-parameters 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeParameters를 참조하세요.

`describe-patch-baselines`

다음 코드 예시는 `describe-patch-baselines`의 사용 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: AWS에서 제공하는 모든 패치 기준을 나열하는 방법

다음 `describe-patch-baselines` 예제에서는 AWS에서 제공하는 모든 패치 기준을 나열합니다.

```
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 정보]를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DescribePatchBaselines]를 참조하세요.

describe-patch-group-state

다음 코드 예시는 describe-patch-group-state의 사용 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>__)와 패치 규정 준수 상태 값 이해를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribePatchGroupState를 참조하세요.

describe-patch-groups

다음 코드 예시는 describe-patch-groups의 사용 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>__)과 패치 기준에 패치 그룹 추가를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribePatchGroups를 참조하세요.

`describe-patch-properties`

다음 코드 예시는 `describe-patch-properties`의 사용 방법을 보여 줍니다.

AWS CLI

Amazon Linux 패치 가용성을 나열하는 방법

다음 `describe-patch-properties` 예시에서는 AWS 계정에서 패치를 사용할 수 있는 Amazon Linux 제품 목록을 표시합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribePatchProperties 섹션을 참조하세요.

describe-sessions

다음 코드 예시는 describe-sessions의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 모든 활성 Session Manager 세션 나열

이 describe-sessions 지정된 사용자가 지난 30일 동안 가장 최근에 만든 활성 세션(연결된 세션과 연결이 끊긴 세션 모두)의 목록을 검색합니다. 이 명령은 Session Manager를 사용하여 시작된 대상에 대한 연결 결과만 반환합니다. 원격 데스크톱 연결이나 SSH와 같은 다른 수단을 통해 이루어진 연결은 나열되지 않습니다.

```
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": {}
        }
    ]
}
```

예시 2: 종료된 모든 Session Manager 세션 나열

이 describe-sessions 예시에서는 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 보기 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeSessions 섹션을 참조하세요.

disassociate-ops-item-related-item

다음 코드 예시는 disassociate-ops-item-related-item의 사용 방법을 보여 줍니다.

AWS CLI

관련 항목 연결을 삭제하는 방법

다음 disassociate-ops-item-related-item 예시에서는 OpsItem과 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 OpsCenter에서 Incident Manager 인스턴스 작업 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DisassociateOpsItemRelatedItem 섹션을 참조하세요.

get-automation-execution

다음 코드 예시는 get-automation-execution의 사용 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 AMI 패치(AWS CLI)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetAutomationExecution을 참조하세요.

get-calendar-state

다음 코드 예시는 get-calendar-state의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 변경 캘린더의 현재 상태 가져오기

이 get-calendar-state 예시에서는 현재 시점의 달력 상태를 반환합니다. 예시에서는 시간을 지정하지 않았으므로 캘린더의 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"
```

```
}
```

예시 2: 지정된 시간에 변경 달력의 상태 가져오기

이 `get-calendar-state` 예시에서는 지정된 시간의 달력 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 가져오기 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetCalendarState 섹션을 참조하세요.

get-command-invocation

다음 코드 예시는 `get-command-invocation`의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 GetCommandInvocation을 참조하세요.

get-connection-status

다음 코드 예시는 get-connection-status의 사용 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetConnectionStatus를 참조하세요.

get-default-patch-baseline

다음 코드 예시는 `get-default-patch-baseline`의 사용 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 정의된 패치 기준 및 사용자 지정 패치 기준 정보(<https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-patch-baselines.html>__)와 기존 패치 기준을 기본값으로 설정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetDefaultPatchBaseline을 참조하세요.

`get-deployable-patch-snapshot-for-instance`

다음 코드 예시는 `get-deployable-patch-snapshot-for-instance`의 사용 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 configure`를 실행하고 인스턴스의 리전만 지정합니다. `Access Key` 및 `Secret Key` 필드는 비워 둡니다.

팁: `uuidgen`을 사용하여 `snapshot-id`를 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.

- API 세부 정보는 AWS CLI 명령 참조의 GetDeployablePatchSnapshotForInstance를 참조하세요.

get-document

다음 코드 예시는 get-document의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 GetDocument 섹션을 참조하세요.

get-inventory-schema

다음 코드 예시는 get-inventory-schema의 사용 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 구성 요소 인벤토리 유형에 대한 인벤토리 스키마를 반환합니다.

명령:

```
aws ssm get-inventory-schema --type-name "AWS:AWSComponent"
```

- API 세부 정보는 AWS CLI 명령 참조의 GetInventorySchema를 참조하세요.

get-inventory

다음 코드 예시는 get-inventory의 사용 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"
      }
  ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetInventory를 참조하세요.

get-maintenance-window-execution-task-invocation

다음 코드 예시는 `get-maintenance-window-execution-task-invocation`의 사용 방법을 보여 줍니다.

AWS CLI

유지 관리 기간 태스크 호출에 대한 정보 가져오기

다음 `get-maintenance-window-execution-task-invocation` 예시에서는 지정된 유지 관리 기간 실행의 일부인 지정된 태스크 간접 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 작업 및 작업 실행에 대한 정보 보기(AWS CLI)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetMaintenanceWindowExecutionTaskInvocation 섹션을 참조하세요.

get-maintenance-window-execution-task

다음 코드 예시는 get-maintenance-window-execution-task의 사용 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 작업 및 작업 실행에 대한 정보 보기(AWS CLI)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetMaintenanceWindowExecutionTask를 참조하세요.

`get-maintenance-window-execution`

다음 코드 예시는 `get-maintenance-window-execution`의 사용 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 작업 및 작업 실행에 대한 정보 보기(AWS CLI)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetMaintenanceWindowExecution을 참조하세요.

`get-maintenance-window-task`

다음 코드 예시는 `get-maintenance-window-task`의 사용 방법을 보여 줍니다.

AWS CLI

유지 관리 기간 태스크에 대한 정보 가져오기

다음 `get-maintenance-window-task` 예시에서는 지정된 유지 관리 기간 태스크에 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 기간에 대한 정보 보기(AWS CLI)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetMaintenanceWindowTask 섹션을 참조하세요.

`get-maintenance-window`

다음 코드 예시는 `get-maintenance-window`의 사용 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 기간에 대한 정보 보기(AWS CLI)](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [GetMaintenanceWindow](#)를 참조하세요.

`get-ops-item`

다음 코드 예시는 `get-ops-item`의 사용 방법을 보여 줍니다.

AWS CLI

OpsItem에 대한 정보 보기

다음 `get-ops-item` 예시에서는 지정된 OpsItem에 대한 세부 정보를 표시합니다.

```
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 설명서의 OpsItems 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetOpsItem 섹션을 참조하세요.

get-ops-summary

다음 코드 예시는 get-ops-summary의 사용 방법을 보여 줍니다.

AWS CLI

모든 OpsItems의 요약을 보는 방법

다음 get-ops-summary 예시에서는 AWS 계정에 있는 모든 OpsItems의 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 설명서의 OpsItems 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetOpsSummary 섹션을 참조하세요.

get-parameter-history

다음 코드 예시는 get-parameter-history의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 GetParameterHistory를 참조하세요.

get-parameter

다음 코드 예시는 `get-parameter`의 사용 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 파라미터의 값을 해독하는 방법

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 작업]을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [GetParameter]를 참조하세요.

get-parameters-by-path

다음 코드 예시는 get-parameters-by-path의 사용 방법을 보여 줍니다.

AWS CLI

특정 경로의 파라미터를 나열하는 방법

다음 get-parameters-by-path 예시에서는 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 구조 작업 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetParametersByPath 섹션을 참조하세요.

get-parameters

다음 코드 예시는 get-parameters의 사용 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 작업]을 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 작업]을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [GetParameters](#)를 참조하세요.

get-patch-baseline-for-patch-group

다음 코드 예시는 `get-patch-baseline-for-patch-group`의 사용 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>__)과 [패치 기준에 패치 그룹 추가](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [GetPatchBaselineForPatchGroup](#)을 참조하세요.

get-patch-baseline

다음 코드 예시는 `get-patch-baseline`의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 GetPatchBaseline을 참조하세요.

get-service-setting

다음 코드 예시는 `get-service-setting`의 사용 방법을 보여 줍니다.

AWS CLI

Parameter Store 처리량에 대한 서비스 설정을 검색하는 방법

다음 `get-service-setting` 이 예시에서는 지정된 리전 내 파라미터 저장소 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 Parameter Store 처리량 증가 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetServiceSetting 섹션을 참조하세요.

`label-parameter-version`

다음 코드 예시는 `label-parameter-version`의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 최신 버전의 파라미터에 레이블 추가

다음 `label-parameter-version` 예시에서는 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.

예시 2: 파라미터의 특정 버전에 레이블 추가

다음 `label-parameter-version` 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 LabelParameterVersion 섹션을 참조하세요.

`list-association-versions`

다음 코드 예시는 `list-association-versions`의 사용 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 작업]을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListAssociationVersions를 참조하세요.

list-associations

다음 코드 예시는 list-associations의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 특정 인스턴스의 연결을 나열하는 방법

다음 list-associations 예제에서는 AssociationName인 UpdateSSMAgent인 모든 연결을 나열합니다.

```
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 작업](#)을 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListAssociations를 참조하세요.

`list-command-invocations`

다음 코드 예시는 `list-command-invocations`의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListCommandInvocations를 참조하세요.

list-commands

다음 코드 예시는 list-commands의 사용 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 요청한 모든 명령을 나열하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 ListCommands를 참조하세요.

list-compliance-items

다음 코드 예시는 list-compliance-items의 사용 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"
```

- API 세부 정보는 AWS CLI 명령 참조의 ListComplianceItems를 참조하세요.

list-compliance-summaries

다음 코드 예시는 `list-compliance-summaries`의 사용 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"
```

- API 세부 정보는 AWS CLI 명령 참조의 ListComplianceSummaries를 참조하세요.

list-document-metadata-history

다음 코드 예시는 `list-document-metadata-history`의 사용 방법을 보여 줍니다.

AWS CLI

예: 변경 템플릿의 승인 기록 및 상태를 보는 방법

다음 `list-document-metadata-history` 예시에서는 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 후 승인 또는 거부 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListDocumentMetadataHistory 섹션을 참조하세요.

list-document-versions

다음 코드 예시는 `list-document-versions`의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListDocumentVersions를 참조하세요.

list-documents

다음 코드 예시는 list-documents의 사용 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 AWS에서 소유하지 않은 프라이빗 공유 문서를 포함한 공유 문서를 나열합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 ListDocuments를 참조하세요.

`list-inventory-entries`

다음 코드 예시는 `list-inventory-entries`의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: 인스턴스의 특정 인벤토리 유형 항목을 보는 방법

다음 `list-inventory-entries` 예제에서는 특정 인스턴스의 AWS:Application 인벤토리 유형에 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"
    }
  ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListInventoryEntries를 참조하세요.

list-ops-item-related-items

다음 코드 예시는 `list-ops-item-related-items`의 사용 방법을 보여 줍니다.

AWS CLI

OpsItem의 관련 항목 리소스를 나열하는 방법

다음 `list-ops-item-related-items` 예시에서는 OpsItem의 관련 항목 리소스를 나열합니다.

```
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 OpsCenter에서 Incident Manager 인스턴스 작업 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListOpsItemRelatedItems 섹션을 참조하세요.

`list-resource-compliance-summaries`

다음 코드 예시는 `list-resource-compliance-summaries`의 사용 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"
```

- API 세부 정보는 AWS CLI 명령 참조의 ListResourceComplianceSummaries를 참조하세요.

`list-resource-data-sync`

다음 코드 예시는 `list-resource-data-sync`의 사용 방법을 보여 줍니다.

AWS CLI

리소스 데이터 동기화 구성을 나열하는 방법

이 예시에서는 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListResourceDataSync 섹션을 참조하세요.

`list-tags-for-resource`

다음 코드 예시는 `list-tags-for-resource`의 사용 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 Tagging AWS Resources를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource를 참조하세요.

modify-document-permission

다음 코드 예시는 modify-document-permission의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyDocumentPermission을 참조하세요.

put-compliance-items

다음 코드 예시는 put-compliance-items의 사용 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"
```

- API 세부 정보는 AWS CLI 명령 참조의 PutComplianceItems를 참조하세요.

put-inventory

다음 코드 예시는 put-inventory의 사용 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'}]"
```

- API 세부 정보는 AWS CLI 명령 참조의 PutInventory를 참조하세요.

put-parameter

다음 코드 예시는 put-parameter의 사용 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 AWS Systems Manager 사용 설명서의 [Systems Manager 파라미터 생성(AWS CLI)](), [파라미터 티어 관리]() 및 [파라미터 정책 작업]()을 참조하세요.

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 AWS Systems Manager 사용 설명서의 Systems Manager 파라미터 생성(AWS CLI), 파라미터 티어 관리 및 파라미터 정책 작업을 참조하세요.

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 AWS Systems Manager 사용 설명서의 Systems Manager 파라미터 생성(AWS CLI), 파라미터 티어 관리 및 파라미터 정책 작업을 참조하세요.

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 AWS Systems Manager 사용 설명서의 Systems Manager 파라미터 생성(AWS CLI), 파라미터 티어 관리 및 파라미터 정책 작업을 참조하세요.

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 AWS Systems Manager 사용 설명서의 Systems Manager 파라미터 생성(AWS CLI), 파라미터 티어 관리 및 파라미터 정책 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutParameter를 참조하세요.

`register-default-patch-baseline`

다음 코드 예시는 `register-default-patch-baseline`의 사용 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 CentOS용 AWS에서 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.

- API 세부 정보는 AWS CLI 명령 참조의 RegisterDefaultPatchBaseline을 참조하세요.

register-patch-baseline-for-patch-group

다음 코드 예시는 register-patch-baseline-for-patch-group의 사용 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>__)과 패치 기준에 패치 그룹 추가를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RegisterPatchBaselineForPatchGroup을 참조하세요.

`register-target-with-maintenance-window`

다음 코드 예시는 `register-target-with-maintenance-window`의 사용 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: 인스턴스 ID를 사용하여 유지 관리 기간에 여러 대상을 등록하는 방법

다음 `register-target-with-maintenance-window` 예제에서는 인스턴스 ID를 지정하여 유지 관리 기간에 두 인스턴스를 등록합니다.

```
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 기간에 대상 인스턴스 등록(AWS CLI)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RegisterTargetWithMaintenanceWindow를 참조하세요.

`register-task-with-maintenance-window`

다음 코드 예시는 `register-task-with-maintenance-window`의 사용 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 기간에 작업 등록(AWS CLI)]을 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 기간에 작업 등록(AWS CLI)을 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 기간에 작업 등록(AWS CLI)을 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 기간에 작업 등록(AWS CLI)](#)을 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 기간에 작업 등록(AWS CLI)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RegisterTaskWithMaintenanceWindow를 참조하세요.

`remove-tags-from-resource`

다음 코드 예시는 `remove-tags-from-resource`의 사용 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 Tagging AWS Resources를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RemoveTagsFromResource를 참조하세요.

`reset-service-setting`

다음 코드 예시는 `reset-service-setting`의 사용 방법을 보여 줍니다.

AWS CLI

Parameter Store 처리량에 대한 서비스 설정을 재설정하는 방법

다음 `reset-service-setting` 예시에서는 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 Parameter Store 처리량 증가 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ResetServiceSetting 섹션을 참조하세요.

resume-session

다음 코드 예시는 `resume-session`의 사용 방법을 보여 줍니다.

AWS CLI

Session Manager 세션을 재개하는 방법

이 `resume-session` 예시에서는 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 AWS CLI의 Session Manager 플러그인 설치 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ResumeSession 섹션을 참조하세요.

send-automation-signal

다음 코드 예시는 send-automation-signal의 사용 방법을 보여 줍니다.

AWS CLI

자동화 실행에 신호를 보내는 방법

다음 send-automation-signal 예시에서는 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 승인자를 사용하여 자동화 실행 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 SendAutomationSignal 섹션을 참조하세요.

send-command

다음 코드 예시는 send-command의 사용 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 실행하는 방법

다음 send-command 예제에서는 모든 알림 이벤트 및 Command 알림 유형에 대해 SNS 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 실행]을 참조하세요.

예제 5: S3 및 CloudWatch로 출력하는 명령을 실행하는 방법

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 실행]을 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.

- API 세부 정보는 AWS CLI 명령 참조의 SendCommand를 참조하세요.

start-associations-once

다음 코드 예시는 start-associations-once의 사용 방법을 보여 줍니다.

AWS CLI

연결을 즉시 한 번만 실행하는 방법

다음 `start-associations-once` 예시에서는 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 보기]를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [StartAssociationsOnce] 섹션을 참조하세요.

`start-automation-execution`

다음 코드 예시는 `start-automation-execution`의 사용 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 실행]을 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.

- API 세부 정보는 AWS CLI 명령 참조의 StartAutomationExecution을 참조하세요.

`start-change-request-execution`

다음 코드 예시는 `start-change-request-execution`의 사용 방법을 보여 줍니다.

AWS CLI

예시 1: 변경 요청 시작

다음 `start-change-request-execution` 예시에서는 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"
```

```
}
```

예시 2: 외부 JSON 파일을 사용하여 변경 요청 시작

다음 `start-automation-execution` 예시에서는 JSON 파일에 지정된 여러 옵션으로 변경 요청을 시작합니다.

```
aws ssm start-change-request-execution \
    --cli-input-json file://MyChangeRequest.json
```

`MyChangeRequest.json`의 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 생성 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartChangeRequestExecution 섹션을 참조하세요.

start-session

다음 코드 예시는 start-session의 사용 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
```

예 2: SSH를 사용하는 Session Manager 세션 시작

이 `start-session` 예에서는 SSH를 사용하는 Session Manager 세션을 위해 인스턴스에 연결합니다. 참고로 이 대화형 명령을 사용하려면 호출을 수행하는 클라이언트 시스템에 Session Manager 플러그인을 설치해야 하며, 명령은 인스턴스의 기본 사용자(예: Linux용 EC2 인스턴스를 위한 `ec2-user`)가 필요합니다.

```
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 AWS CLI용 Session Manager 플러그인 설치를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartSession를 참조하세요.

`stop-automation-execution`

다음 코드 예시는 `stop-automation-execution`의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 StopAutomationExecution을 참조하세요.

`terminate-session`

다음 코드 예시는 `terminate-session`의 사용 방법을 보여 줍니다.

AWS CLI

Session Manager 세션을 종료하는 방법

이 `terminate-session` 예시는 사용자가 생성한 'Shirley-Rodriguez' 세션을 영구적으로 종료하고 인스턴스의 Session Manager 클라이언트와 SSM Agent 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 종료 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TerminateSession 섹션을 참조하세요.

`unlabel-parameter-version`

다음 코드 예시는 `unlabel-parameter-version`의 사용 방법을 보여 줍니다.

AWS CLI

파라미터 레이블 삭제

다음 `unlabel-parameter-version` 예시에서는 주어진 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) 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UnlabelParameterVersion 섹션을 참조하세요.

update-association-status

다음 코드 예시는 update-association-status의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateAssociationStatus를 참조하세요.

update-association

다음 코드 예시는 update-association의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateAssociation을 참조하세요.

update-document-default-version

다음 코드 예시는 update-document-default-version의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateDocumentDefaultVersion을 참조하세요.

update-document-metadata

다음 코드 예시는 `update-document-metadata`의 사용 방법을 보여 줍니다.

AWS CLI

예시 4: 변경 템플릿의 최신 버전 승인

다음 `update-document-metadata`는 검토를 위해 제출된 변경 템플릿의 최신 버전에 대한 승인을 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 후 승인 또는 거부 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateDocumentMetadata 섹션을 참조하세요.

update-document

다음 코드 예시는 `update-document`의 사용 방법을 보여 줍니다.

AWS CLI

문서의 새 버전을 생성하는 방법

다음 update-document 예제에서는 Windows 컴퓨터에서 실행 시 문서의 새 버전을 생성합니다. --document에서 지정한 문서는 JSON 형식이어야 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 프롬프트에서는 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateDocument 섹션을 참조하세요.

update-maintenance-window-target

다음 코드 예시는 update-maintenance-window-target의 사용 방법을 보여 줍니다.

AWS CLI

유지 관리 기간 대상 업데이트

다음 update-maintenance-window-target 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateMaintenanceWindowTarget 섹션을 참조하세요.

`update-maintenance-window-task`

다음 코드 예시는 `update-maintenance-window-task`의 사용 방법을 보여 줍니다.

AWS CLI

유지 관리 기간 태스크 업데이트

다음 `update-maintenance-window-task` 예시에서는 유지 관리 기간 태스크의 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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateMaintenanceWindowTask 섹션을 참조하세요.

update-maintenance-window

다음 코드 예시는 update-maintenance-window의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 [UpdateMaintenanceWindow](#)를 참조하세요.

`update-managed-instance-role`

다음 코드 예시는 `update-managed-instance-role`의 사용 방법을 보여 줍니다.

AWS CLI

관리형 인스턴스의 IAM 역할을 업데이트하는 방법

다음 `update-managed-instance-role` 예제에서는 관리형 인스턴스의 IAM 인스턴스 프로파일을 업데이트합니다.

```
aws ssm update-managed-instance-role \
    --instance-id "mi-08ab247cdfEXAMPLE" \
```

```
    --iam-role "ExampleRole"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Systems Manager 사용 설명서의 4단계: Systems Manager에 대한 IAM 인스턴스 프로파일 생성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateManagedInstanceRole을 참조하세요.

update-ops-item

다음 코드 예시는 update-ops-item의 사용 방법을 보여 줍니다.

AWS CLI

OpsItem을 업데이트하는 방법

다음 update-ops-item 예시에서는 OpsItem의 설명, 우선순위 및 범주를 업데이트합니다. 또한 이 명령은 이 OpsItem을 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 설명서의 OpsItems 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateOpsItem을 참조하세요.

update-patch-baseline

다음 코드 예시는 update-patch-baseline의 사용 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.

- API 세부 정보는 AWS CLI 명령 참조의 [UpdatePatchBaseline](#)을 참조하세요.

`update-resource-data-sync`

다음 코드 예시는 `update-resource-data-sync`의 사용 방법을 보여 줍니다.

AWS CLI

리소스 데이터 동기화 삭제

다음 `update-resource-data-sync` 예시에서는 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 AWS Systems Manager 사용 설명서의 여러 계정 및 리전에서 데이터를 표시하도록 Systems Manager Explorer 설정 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateResourceDataSync 섹션을 참조하세요.

`update-service-setting`

다음 코드 예시는 `update-service-setting`의 사용 방법을 보여 줍니다.

AWS CLI

Parameter Store 처리량에 대한 서비스 설정을 업데이트하는 방법

다음 `update-service-setting` 예시에서는 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 Parameter Store 처리량 증가 섹션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateServiceSetting 섹션을 참조하세요.

AWS CLI를 사용한 Amazon Textract 예시

다음 코드 예시는 Amazon Textract와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

analyze-document

다음 코드 예시에서는 analyze-document의 사용 방법을 보여줍니다.

AWS CLI

문서 텍스트 분석

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 Amazon Textract로 문서 텍스트 분석을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AnalyzeDocument를 참조하세요.

detect-document-text

다음 코드 예시에서는 `detect-document-text`의 사용 방법을 보여줍니다.

AWS CLI

문서 텍스트 감지

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 Amazon Textract로 문서 텍스트 분석을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DetectDocumentText를 참조하세요.

get-document-analysis

다음 코드 예시에서는 get-document-analysis의 사용 방법을 보여줍니다.

AWS CLI

여러 페이지 문서의 비동기 텍스트 분석 결과 가져오기

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 여러 페이지 문서의 텍스트 감지 및 분석을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetDocumentAnalysis를 참조하세요.

get-document-text-detection

다음 코드 예시에서는 get-document-text-detection의 사용 방법을 보여줍니다.

AWS CLI

여러 페이지 문서의 비동기 텍스트 감지 결과 가져오기

다음 get-document-text-detection 예시에서는 여러 페이지 문서의 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 여러 페이지 문서의 텍스트 감지 및 분석을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [GetDocumentTextDetection](https://docs.aws.amazon.com/goto/aws-cli/textract-2018-06-27/GetDocumentTextDetection)을 참조하세요.

start-document-analysis

다음 코드 예시에서는 `start-document-analysis`의 사용 방법을 보여줍니다.

AWS CLI

여러 페이지 문서의 텍스트 분석 시작

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 여러 페이지 문서의 텍스트 감지 및 분석을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartDocumentAnalysis를 참조하세요.

`start-document-text-detection`

다음 코드 예시에서는 `start-document-text-detection`의 사용 방법을 보여줍니다.

AWS CLI

여러 페이지 문서의 텍스트 감지 시작

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 여러 페이지 문서의 텍스트 감지 및 분석을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartDocumentTextDetection을 참조하세요.

AWS CLI를 사용한 Amazon Transcribe 예시

다음 코드 예시는 Amazon Transcribe와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

create-language-model

다음 코드 예시에서는 create-language-model의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 학습 및 튜닝 데이터를 모두 사용하여 사용자 지정 언어 모델 생성

다음 create-language-model 예시에서는 사용자 지정 언어 모델을 생성합니다. 사용자 지정 언어 모델을 사용하여 법률, 숙박업 서비스, 금융 및 보험과 같은 도메인의 트랜스크립션 성능을 개선할 수 있습니다. 언어 코드에 유효한 언어 코드를 입력합니다. base-model-name의 경우 사용자 지정 언어 모델을 사용하여 전사하려는 오디오의 샘플 속도에 가장 적합한 기본 모델을 지정합니다. 모델 이름에서 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://amzn-s3-demo-bucket/Amazon-S3-Prefix-for-
training-data",TuningDataS3Uri="s3://amzn-s3-demo-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://amzn-s3-demo-bucket/Amazon-S3-Prefix/",
        "TuningDataS3Uri": "s3://amzn-s3-demo-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.

예시 2: 훈련 데이터만 사용하여 사용자 지정 언어 모델 생성

다음 `create-language-model` 예시에서는 오디오 파일을 트랜스크립션합니다. 사용자 지정 언어 모델을 사용하여 법률, 숙박업 서비스, 금융 및 보험과 같은 도메인의 트랜스크립션 성능을 개선할 수 있습니다. 언어 코드에 유효한 언어 코드를 입력합니다. base-model-name의 경우 사용자 지정 언어 모델을 사용하여 전사하려는 오디오의 샘플 속도에 가장 적합한 기본 모델을 지정합니다. 모델 이름에서 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://amzn-s3-demo-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://amzn-s3-demo-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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateLanguageModel을 참조하세요.

create-medical-vocabulary

다음 코드 예시에서는 `create-medical-vocabulary`의 사용 방법을 보여줍니다.

AWS CLI

의료용 사용자 지정 어휘 생성

다음 `create-medical-vocabulary` 예시에서는 사용자 지정 어휘를 생성합니다. 사용자 지정 어휘를 생성하려면 더 정확하게 트랜스크립션하려는 모든 용어가 포함된 텍스트 파일을 만들어야 합니다. vocabulary-file-uri의 경우 해당 텍스트 파일의 Amazon Simple Storage Service(Amazon S3) URI를 지정합니다. language-code에서 사용자 지정 어휘의 언어에 해당하는 언어 코드를 지정합니다. vocabulary-name에서 사용자 지정 어휘를 지칭할 이름을 지정합니다.

```
aws transcribe create-medical-vocabulary \
    --vocabulary-name cli-medical-vocab-example \
    --language-code language-code \
    --vocabulary-file-uri https://amzn-s3-demo-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 어휘](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [CreateMedicalVocabulary](#) 섹션을 참조하세요.

`create-vocabulary-filter`

다음 코드 예시에서는 `create-vocabulary-filter`의 사용 방법을 보여줍니다.

AWS CLI

어휘 필터 생성

다음 `create-vocabulary-filter` 예시에서는 트랜스크립션에 표시하지 않을 단어 목록이 포함된 텍스트 파일을 사용하는 어휘 필터를 생성합니다. language-code의 경우, 어휘 필터의 언어에 해당하는 언어 코드를 지정합니다. vocabulary-filter-file-uri의 경우 텍스트 파일의 Amazon Simple Storage Service(Amazon S3) URI를 지정합니다. vocabulary-filter-name의 경우, 어휘 필터의 이름을 지정합니다.

```
aws transcribe create-vocabulary-filter \
    --language-code language-code \
    --vocabulary-filter-file-uri s3://amzn-s3-demo-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 Filtering Unwanted Words를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateVocabularyFilter 섹션을 참조하세요.

create-vocabulary

다음 코드 예시에서는 `create-vocabulary`의 사용 방법을 보여줍니다.

AWS CLI

사용자 지정 어휘 생성

다음 `create-vocabulary` 예시에서는 사용자 지정 어휘를 생성합니다. 사용자 지정 어휘를 생성하려면 더 정확하게 트랜스크립션하려는 모든 용어가 포함된 텍스트 파일을 만들어야 합니다. vocabulary-file-uri의 경우 해당 텍스트 파일의 Amazon Simple Storage Service(Amazon S3) URI를 지정합니다. language-code에서 사용자 지정 어휘의 언어에 해당하는 언어 코드를 지정합니다. vocabulary-name에서 사용자 지정 어휘를 지칭할 이름을 지정합니다.

```
aws transcribe create-vocabulary \
    --language-code language-code \
    --vocabulary-name cli-vocab-example \
    --vocabulary-file-uri s3://amzn-s3-demo-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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateVocabulary를 참조하세요.

delete-language-model

다음 코드 예시에서는 delete-language-model의 사용 방법을 보여줍니다.

AWS CLI

사용자 지정 언어 모델을 삭제하는 방법

다음 delete-language-model 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteLanguageModel을 참조하세요.

delete-medical-transcription-job

다음 코드 예시에서는 delete-medical-transcription-job의 사용 방법을 보여줍니다.

AWS CLI

의료 트랜스크립션 작업 삭제

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 안내서의 DeleteMedicalTranscriptionJob을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteMedicalTranscriptionJob을 참조하세요.

delete-medical-vocabulary

다음 코드 예시에서는 `delete-medical-vocabulary`의 사용 방법을 보여줍니다.

AWS CLI

의료용 사용자 지정 어휘 삭제

다음 `delete-medical-vocabulary` 예시에서는 의료 사용자 지정 어휘를 삭제합니다. vocabulary-name의 경우 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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteMedicalVocabulary 섹션을 참조하세요.

delete-transcription-job

다음 코드 예시에서는 `delete-transcription-job`의 사용 방법을 보여줍니다.

AWS CLI

트랜스크립션 작업 중 하나 삭제

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 안내서의 DeleteTranscriptionJob을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteTranscriptionJob을 참조하세요.

delete-vocabulary-filter

다음 코드 예시에서는 delete-vocabulary-filter의 사용 방법을 보여줍니다.

AWS CLI

어휘 필터를 삭제하는 방법

다음 delete-vocabulary-filter 예시에서는 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 Filtering Unwanted Words를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteVocabularyFilter 섹션을 참조하세요.

delete-vocabulary

다음 코드 예시에서는 delete-vocabulary의 사용 방법을 보여줍니다.

AWS CLI

사용자 지정 어휘 삭제

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.

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteVocabulary](https://docs.aws.amazon.com/goto/aws-cli/transcribe-2017-10-26/DeleteVocabulary)를 참조하세요.

describe-language-model

다음 코드 예시에서는 `describe-language-model`의 사용 방법을 보여줍니다.

AWS CLI

특정 사용자 지정 언어 모델에 대한 정보를 가져오는 방법

다음 `describe-language-model` 예시에서는 특정 사용자 지정 언어 모델에 대한 정보를 가져옵니다. 예를 들어, BaseModelName에서 모델이 협대역 모델을 사용하여 학습되었는지, 아니면 광대역 모델을 사용하여 학습되었는지 확인할 수 있습니다. NarrowBand 기본 모델을 사용하는 사용자 지정 언어 모델은 샘플 레이트가 16kHz 미만인 오디오를 텍스트로 변환할 수 있습니다. 와이드밴드 기본 모델을 사용하는 언어 모델은 16kHz 이상의 샘플 레이트로 오디오를 텍스트로 변환할 수 있습니다. S3Uri 파라미터는 사용자 지정 언어 모델을 만들기 위해 학습 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://amzn-s3-demo-bucket/Amazon-S3-Prefix/",
            "TuningDataS3Uri": "s3://amzn-s3-demo-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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeLanguageModel 섹션을 참조하세요.

get-medical-transcription-job

다음 코드 예시에서는 `get-medical-transcription-job`의 사용 방법을 보여줍니다.

AWS CLI

특정 의료용 트랜스크립션 작업에 대한 정보 가져오기

다음 `get-medical-transcription-job` 예시에서는 특정 의료 트랜스크립션 작업에 대한 정보를 가져옵니다. 트랜스크립션 결과에 액세스하려면 TranscriptFileUri 파라미터를 사용합니다. 트랜스크립션 작업에 추가 기능을 사용 설정한 경우 설정 객체에서 해당 기능을 확인할 수 있습니다. Specialty 파라미터는 의료 제공자의 전문 분야를 표시합니다. Type 파라미터는 트랜스크립션 작업의 음성이 의료 대화인지 아니면 의료 받아쓰기인지를 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 Batch Transcription을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetMedicalTranscriptionJob을 참조하세요.

get-medical-vocabulary

다음 코드 예시에서는 `get-medical-vocabulary`의 사용 방법을 보여줍니다.

AWS CLI

의료용 사용자 지정 어휘에 대한 정보 가져오기

다음 `get-medical-vocabulary` 예시에서는 의료 사용자 지정 어휘에 대한 정보를 가져옵니다. VocabularyState 파라미터를 사용하여 어휘의 처리 상태를 확인할 수 있습니다. READY인 경우 StartMedicalTranscriptionJob 작업에서 사용할 수 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.

- API 세부 정보는 AWS CLI 명령 참조의 GetMedicalVocabulary 섹션을 참조하세요.

get-transcription-job

다음 코드 예시에서는 get-transcription-job의 사용 방법을 보여줍니다.

AWS CLI

특정 트랜스크립션 작업의 정보 가져오기

다음 get-transcription-job 예시에서는 특정 트랜스크립션 작업의 정보를 가져옵니다. 트랜스크립션 결과에 액세스하려면 TranscriptFileUri 파라미터를 사용합니다. MediaFileUri 파라미터를 사용하면 이 작업에서 어떤 오디오 파일을 트랜스크립션했는지 확인할 수 있습니다. Settings 객체를 사용하면 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://amzn-s3-demo-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 Command Line Interface)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetTranscriptionJob을 참조하세요.

get-vocabulary-filter

다음 코드 예시에서는 get-vocabulary-filter의 사용 방법을 보여줍니다.

AWS CLI

어휘 필터에 대한 정보 가져오기

다음 get-vocabulary-filter 예시에서는 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 Filter Unwanted Words를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetVocabularyFilter 섹션을 참조하세요.

get-vocabulary

다음 코드 예시에서는 get-vocabulary의 사용 방법을 보여줍니다.

AWS CLI

사용자 지정 어휘의 정보 가져오기

다음 `get-vocabulary` 예시에서는 이전에 생성한 사용자 지정 어휘의 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 어휘](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [GetVocabulary](#)를 참조하세요.

`list-language-models`

다음 코드 예시에서는 `list-language-models`의 사용 방법을 보여줍니다.

AWS CLI

사용자 지정 언어 모델을 나열하는 방법

다음 `list-language-models` 예시에서는 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. BaseModelName은 NarrowBand 또는 WideBand 모델을 사용하여 사용자 지정 언어 모델을 생성했는지 여부를 알려줍니다. NarrowBand 기본 모델을 사용하여 사용자 지정 언어 모델을 사용하여 샘플 속도가 16kHz 미만인 오디오를 트랜스크립션할 수 있습니다. WideBand 기본 모델을 사용하는 사용자 지정 언어 모델로 오디오 16kHz 이상을 트랜스크립션할 수 있습니다. `ModelStatus` 파라미터는 트랜스크립션 작업에서 사용자 지정 언어 모델을 사용할 수 있는지 여부를 보여줍니다. 값이 COMPLETED면 트랜스크립션 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://amzn-s3-demo-bucket/clm-training-data/",
                "TuningDataS3Uri": "s3://amzn-s3-demo-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://amzn-s3-demo-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://amzn-s3-demo-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.

- API 세부 정보는 AWS CLI 명령 참조의 ListLanguageModels 섹션을 참조하세요.

list-medical-transcription-jobs

다음 코드 예시에서는 `list-medical-transcription-jobs`의 사용 방법을 보여줍니다.

AWS CLI

의료 트랜스크립션 작업 나열

다음 `list-medical-transcription-jobs` 예시에서는 AWS 계정 및 리전에 연결된 의료 트랜스크립션 작업을 나열합니다. 특정 트랜스크립션 작업에 대한 자세한 정보를 얻으려면 트랜스크립션 출력에서 MedicalTranscriptionJobName 파라미터의 값을 복사하고 `get-medical-transcription-job` 명령의 `MedicalTranscriptionJobName` 옵션에 해당 값을 지정합니다. 더 많은 트랜스크립션 작업을 보려면 NextToken 파라미터의 값을 복사하고 `list-medical-transcription-jobs` 명령을 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.html을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListMedicalTranscriptionJobs를 참조하세요.

list-medical-vocabularies

다음 코드 예시에서는 `list-medical-vocabularies`의 사용 방법을 보여줍니다.

AWS CLI

의료용 사용자 지정 어휘 나열

다음 `list-medical-vocabularies` 예시에서는 AWS 계정 및 리전과 연결된 의료 사용자 지정 어휘를 나열합니다. 특정 트랜스크립션 작업에 대한 자세한 정보를 얻으려면 트랜스크립

션 출력에서 `MedicalTranscriptionJobName` 파라미터의 값을 복사하고 `get-medical-transcription-job` 명령의 `MedicalTranscriptionJobName` 옵션에 해당 값을 지정합니다. 더 많은 트랜스크립션 작업을 보려면 `NextToken` 파라미터의 값을 복사하고 `list-medical-transcription-jobs` 명령을 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 어휘](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ListMedicalVocabularies](#) 섹션을 참조하세요.

`list-transcription-jobs`

다음 코드 예시에서는 `list-transcription-jobs`의 사용 방법을 보여줍니다.

AWS CLI

트랜스크립션 작업 나열

다음 `list-transcription-jobs` 예시에서는 AWS 계정 및 리전에 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 Command Line Interface)]를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ListTranscriptionJobs]를 참조하세요.

`list-vocabularies`

다음 코드 예시에서는 `list-vocabularies`의 사용 방법을 보여줍니다.

AWS CLI

사용자 지정 어휘 나열

다음 `list-vocabularies` 예시에서는 AWS 계정 및 리전에 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListVocabularies를 참조하세요.

`list-vocabulary-filters`

다음 코드 예시에서는 `list-vocabulary-filters`의 사용 방법을 보여줍니다.

AWS CLI

어휘 필터를 나열하는 방법

다음 `list-vocabulary-filters` 예시에서는 AWS 계정 및 리전과 연결된 의료 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 Filtering Unwanted Words를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListVocabularyFilters 섹션을 참조하세요.

start-medical-transcription-job

다음 코드 예시에서는 `start-medical-transcription-job`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 오디오 파일로 저장된 의료 구술 트랜스크립션

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":"amzn-s3-demo-bucket",
    "Media": {
        "MediaFileUri": "s3://amzn-s3-demo-bucket/your-audio-file.extension"
    }
}
```

출력:

```
{
    "MedicalTranscriptionJob": {
        "MedicalTranscriptionJobName": "simple-dictation-medical-transcription-job",
        "TranscriptionJobStatus": "IN_PROGRESS",
        "LanguageCode": "language-code",
        "Media": {
            "MediaFileUri": "s3://amzn-s3-demo-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.

예시 2: 오디오 파일로 저장된 의사와 환자 간 대화 트랜스크립션

다음 `start-medical-transcription-job` 예시에서는 의사와 환자 간 대화가 포함된 오디오 파일을 트랜스크립션합니다. OutputBucketName 파라미터에는 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":"amzn-s3-demo-bucket",
    "Media": {
        "MediaFileUri": "s3://amzn-s3-demo-bucket/your-audio-file.extension"
    }
}
```

출력:

```
{
    "MedicalTranscriptionJob": {
        "MedicalTranscriptionJobName": "simple-conversation-medical-transcription-
job",
        "TranscriptionJobStatus": "IN_PROGRESS",
        "LanguageCode": "language-code",
        "Media": {
            "MediaFileUri": "s3://amzn-s3-demo-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.

예시 3: 의사와 환자 간 대화가 담긴 다중 채널 오디오 파일 트랜스크립션

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":"amzn-s3-demo-bucket",
        "Media": {
          "MediaFileUri": "s3://amzn-s3-demo-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://amzn-s3-demo-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 식별]을 참조하세요.

예시 4: 의사와 환자 간 대화의 오디오 파일을 트랜스크립션하고 트랜스크립션 출력에서 화자 식별

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":"amzn-s3-demo-bucket",
    "Media": {
        "MediaFileUri": "s3://amzn-s3-demo-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://amzn-s3-demo-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.

예시 5: 최대 2개의 대체 트랜스크립션을 사용하여 오디오 파일로 저장된 의료 대화 트랜스크립션

다음 `start-medical-transcription-job` 예시에서는 단일 오디오 파일에서 최대 2개의 대체 트랜스크립션을 생성합니다. 모든 트랜스크립션에는 신뢰도가 있습니다. 기본적으로 Amazon Transcribe은 신뢰도가 가장 높은 트랜스크립션을 반환합니다. Amazon Transcribe가 신뢰도가 낮은 추가 트랜스크립션을 반환하도록 지정할 수도 있습니다. `OutputBucketName` 파라미터에는 트랜스크립션 출력의 위치를 지정합니다.

```
aws transcribe start-medical-transcription-job \
    --cli-input-json file://myfifthfile.json
```

`myfifthfile.json`의 콘텐츠:

```
{
    "MedicalTranscriptionJobName": "alternatives-conversation-medical-transcription-job",
    "LanguageCode": "language-code",
    "Specialty": "PRIMARYCARE",
    "Type": "CONVERSATION",
    "OutputBucketName":"amzn-s3-demo-bucket",
    "Media": {
        "MediaFileUri": "s3://amzn-s3-demo-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://amzn-s3-demo-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 트랜스크립션](#)을 참조하세요.

예시 6: 최대 2개의 대체 트랜스크립션을 사용하여 의료 구술 오디오 파일 트랜스크립션

다음 `start-medical-transcription-job` 예시에서는 오디오 파일을 트랜스크립션하고 어휘 필터를 사용하여 원하지 않는 단어를 마스킹합니다. OutputBucketName 파라미터에는 트랜스크립션 출력의 위치를 지정합니다.

```
aws transcribe start-medical-transcription-job \
    --cli-input-json file://mysixthfile.json
```

`mysixthfile.json`의 콘텐츠:

```
{
    "MedicalTranscriptionJobName": "alternatives-conversation-medical-transcription-
job",
    "LanguageCode": "language-code",
    "Specialty": "PRIMARYCARE",
    "Type": "DICTATION",
    "OutputBucketName":"amzn-s3-demo-bucket",
    "Media": {
```

```
        "MediaFileUri": "s3://amzn-s3-demo-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://amzn-s3-demo-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.

예시 7: 사용자 지정 어휘로 정확도를 높여 의료 구술 오디오 파일 트랜스크립션

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":"amzn-s3-demo-bucket",
    "Media": {
        "MediaFileUri": "s3://amzn-s3-demo-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://amzn-s3-demo-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.

- API 세부 정보는 AWS CLI 명령 참조의 StartMedicalTranscriptionJob을 참조하세요.

`start-transcription-job`

다음 코드 예시에서는 `start-transcription-job`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 오디오 파일 트랜스크립션

다음 `start-transcription-job` 예시에서는 오디오 파일을 트랜스크립션합니다.

```
aws transcribe start-transcription-job \
    --cli-input-json file://myfile.json
```

`myfile.json`의 콘텐츠:

```
{
    "TranscriptionJobName": "cli-simple-transcription-job",
    "LanguageCode": "the-language-of-your-transcription-job",
    "Media": {
        "MediaFileUri": "s3://amzn-s3-demo-bucket/Amazon-S3-prefix/your-media-file-name.file-extension"
    }
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 [시작하기(AWS Command Line Interface)](#)를 참조하세요.

예시 2: 다중 채널 오디오 파일 트랜스크립션

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://amzn-s3-demo-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://amzn-s3-demo-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 트랜스크립션]을 참조하세요.

예시 3: 오디오 파일을 트랜스크립션하고 다양한 화자 식별

다음 `start-transcription-job` 예시에서는 오디오 파일을 트랜스크립션하고 트랜스크립션 출력에서 화자를 식별합니다.

```
aws transcribe start-transcription-job \
    --cli-input-json file://mythirdfile.json
```

`mythirdfile.json`의 콘텐츠:

```
{
    "TranscriptionJobName": "cli-speakerid-job",
    "LanguageCode": "the-language-of-your-transcription-job",
    "Media": {
        "MediaFileUri": "s3://amzn-s3-demo-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://amzn-s3-demo-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.

예시 4: 오디오 파일을 트랜스크립션하고 트랜스크립션 출력에서 원하지 않는 단어 마스킹

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.json`의 콘텐츠:

```
{
    "TranscriptionJobName": "cli-filter-mask-job",
    "LanguageCode": "the-language-of-your-transcription-job",
    "Media": {
          "MediaFileUri": "s3://amzn-s3-demo-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.

예시 5: 오디오 파일을 트랜스크립션하고 트랜스크립션 출력에서 원하지 않는 단어 제거

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.json`의 콘텐츠:

```
{
    "TranscriptionJobName": "cli-filter-remove-job",
    "LanguageCode": "the-language-of-your-transcription-job",
    "Media": {
        "MediaFileUri": "s3://amzn-s3-demo-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://amzn-s3-demo-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.

예시 6: 사용자 지정 어휘로 정확도를 높여 오디오 파일 트랜스크립션

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.json`의 콘텐츠:

```
{
    "TranscriptionJobName": "cli-vocab-job",
    "LanguageCode": "the-language-of-your-transcription-job",
```

```
    "Media": {
        "MediaFileUri": "s3://amzn-s3-demo-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://amzn-s3-demo-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.

예시 7: 오디오 파일의 언어를 식별하고 트랜스크립션

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.json`의 콘텐츠:

```
{
    "TranscriptionJobName": "cli-identify-language-transcription-job",
```

```
    "IdentifyLanguage": true,
    "Media": {
        "MediaFileUri": "s3://amzn-s3-demo-bucket/Amazon-S3-prefix/your-media-file-
name.file-extension"
    }
}
```

출력:

```
{
    "TranscriptionJob": {
        "TranscriptionJobName": "cli-identify-language-transcription-job",
        "TranscriptionJobStatus": "IN_PROGRESS",
        "Media": {
            "MediaFileUri": "s3://amzn-s3-demo-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.

예시 8: 개인 식별 정보를 편집하여 오디오 파일 트랜스크립션

다음 `start-transcription-job` 예시에서는 오디오 파일을 트랜스크립션하고 트랜스크립션 출력에서 개인 식별 정보를 편집합니다.

```
aws transcribe start-transcription-job \
    --cli-input-json file://myeighthfile.json
```

`myeigthfile.json`의 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.

예시 9: 개인 식별 정보(PII)가 편집된 트랜스크립트와 편집되지 않은 트랜스크립트 생성

다음 `start-transcription-job` 예시에서는 오디오 파일의 트랜스크립션 2개를 생성합니다. 하나는 개인 식별 정보를 편집한 것이고 다른 하나는 편집하지 않은 것입니다.

```
aws transcribe start-transcription-job \
    --cli-input-json file://myninthfile.json
```

`myninthfile.json`의 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.

예시 10: 이전에 생성한 사용자 지정 언어 모델을 사용하여 오디오 파일 트랜스크립션

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://amzn-s3-demo-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://amzn-s3-demo-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.

- API 세부 정보는 AWS CLI 명령 참조의 StartTranscriptionJob을 참조하세요.

update-medical-vocabulary

다음 코드 예시에서는 update-medical-vocabulary의 사용 방법을 보여줍니다.

AWS CLI

의료용 사용자 지정 어휘를 새 용어로 업데이트

다음 update-medical-vocabulary 예시에서는 의료 사용자 지정 어휘에 사용되는 용어를 새 어휘로 바꿉니다. 사전 조건: 의료 사용자 지정 어휘의 용어를 바꾸려면 새 용어가 포함된 파일이 필요합니다.

```
aws transcribe update-medical-vocabulary \
    --vocabulary-file-uri s3://amzn-s3-demo-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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateMedicalVocabulary 섹션을 참조하세요.

update-vocabulary-filter

다음 코드 예시에서는 update-vocabulary-filter의 사용 방법을 보여줍니다.

AWS CLI

어휘 필터의 단어를 바꾸려면

다음 update-vocabulary-filter 예시에서는 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://amzn-s3-demo-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 Filtering Unwanted Words를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateVocabularyFilter 섹션을 참조하세요.

update-vocabulary

다음 코드 예시에서는 update-vocabulary의 사용 방법을 보여줍니다.

AWS CLI

사용자 지정 어휘를 새 용어로 업데이트

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://amzn-s3-demo-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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateVocabulary를 참조하세요.

AWS CLI를 사용한 Amazon Translate 예시

다음 코드 예시는 Amazon Translate와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

import-terminology

다음 코드 예시에서는 import-terminology의 사용 방법을 보여줍니다.

AWS CLI

파일에서 사용자 지정 용어 가져오기

다음 import-terminology 예시에서는 test-terminology.csv 파일에서 MyTestTerminology라는 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"
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ImportTerminology를 참조하세요.

AWS CLI를 사용한 Trusted Advisor 예시

다음 코드 예시에서는 Trusted Advisor에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

get-organization-recommendation

다음 코드 예시에서는 get-organization-recommendation의 사용 방법을 보여줍니다.

AWS CLI

조직 권장 사항 가져오기

다음 get-organization-recommendation 예시에서는 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 AWS Trusted Advisor 사용자 안내서의 Trusted Advisor API 시작하기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetOrganizationRecommendation을 참조하세요.

get-recommendation

다음 코드 예시에서는 `get-recommendation`의 사용 방법을 보여줍니다.

AWS CLI

권장 사항 가져오기

다음 `get-recommendation` 예시에서는 해당 식별자로 권장 사항을 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 AWS Trusted Advisor 사용자 안내서의 Trusted Advisor API 시작하기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetRecommendation을 참조하세요.

list-checks

다음 코드 예시에서는 `list-checks`의 사용 방법을 보여줍니다.

AWS CLI

Trusted Advisor 검사 나열

다음 `list-checks` 예시에서는 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 AWS Trusted Advisor 사용자 안내서의 Trusted Advisor API 시작하기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListChecks를 참조하세요.

list-organization-recommendation-accounts

다음 코드 예시에서는 `list-organization-recommendation-accounts`의 사용 방법을 보여줍니다.

AWS CLI

조직 권장 사항 계정 나열

다음 `list-organization-recommendation-accounts` 예시에서는 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 AWS Trusted Advisor 사용자 안내서의 Trusted Advisor API 시작하기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListOrganizationRecommendationAccounts를 참조하세요.

`list-organization-recommendation-resources`

다음 코드 예시에서는 `list-organization-recommendation-resources`의 사용 방법을 보여줍니다.

AWS CLI

조직 권장 사항 리소스 나열

다음 `list-organization-recommendation-resources` 예시에서는 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 AWS Trusted Advisor 사용자 안내서의 Trusted Advisor API 시작하기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListOrganizationRecommendationResources를 참조하세요.

list-organization-recommendations

다음 코드 예시에서는 `list-organization-recommendations`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 조직 권장 사항 나열

다음 `list-organization-recommendations` 예시에서는 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 AWS Trusted Advisor 사용자 안내서의 Trusted Advisor API 시작하기를 참조하세요.

예시 2: 필터를 사용하여 조직 권장 사항 나열

다음 `list-organization-recommendations` 예시에서는 'security' 원칙에 포함된 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 AWS Trusted Advisor 사용자 안내서의 Trusted Advisor API 시작하기를 참조하세요.

예시 3: 페이지 매김 토큰을 사용하여 조직 권장 사항 나열

다음 `list-organization-recommendations` 예시에서는 이전 요청에서 반환된 ‘nextToken’을 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 AWS Trusted Advisor 사용자 안내서의 Trusted Advisor API 시작하기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListOrganizationRecommendations를 참조하세요.

list-recommendation-resources

다음 코드 예시에서는 `list-recommendation-resources`의 사용 방법을 보여줍니다.

AWS CLI

권장 리소스 나열

다음 `list-recommendation-resources` 예시에서는 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 AWS Trusted Advisor 사용자 안내서의 Trusted Advisor API 시작하기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListRecommendationResources를 참조하세요.

list-recommendations

다음 코드 예시에서는 `list-recommendations`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: 권장 사항 나열

다음 `list-recommendations` 예시에서는 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 AWS Trusted Advisor 사용자 안내서의 Trusted Advisor API 시작하기를 참조하세요.

예시 2: 필터를 사용하여 권장 사항 나열

다음 `list-recommendations` 예시에서는 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 AWS Trusted Advisor 사용자 안내서의 [Trusted Advisor API 시작하기](#)를 참조하세요.

예시 3: 페이지 매김 토큰을 사용하여 권장 사항 나열

다음 `list-recommendations` 예시에서는 이전 요청에서 반환된 'nextToken'을 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 AWS Trusted Advisor 사용자 안내서의 Trusted Advisor API 시작하기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListRecommendations를 참조하세요.

update-organization-recommendation-lifecycle

다음 코드 예시에서는 update-organization-recommendation-lifecycle의 사용 방법을 보여줍니다.

AWS CLI

조직 권장 사항 수명 주기 업데이트

다음 update-organization-recommendation-lifecycle 예시에서는 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 AWS Trusted Advisor 사용자 안내서의 Trusted Advisor API 시작하기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateOrganizationRecommendationLifecycle을 참조하세요.

update-recommendation-lifecycle

다음 코드 예시에서는 update-recommendation-lifecycle의 사용 방법을 보여줍니다.

AWS CLI

권장 사항 수명 주기 업데이트

다음 update-recommendation-lifecycle 예시에서는 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 AWS Trusted Advisor 사용자 안내서의 Trusted Advisor API 시작하기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateRecommendationLifecycle을 참조하세요.

AWS CLI를 사용하여 Verified Permissions 예시

다음 코드 예시에서는 Verified Permissions에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`create-identity-source`

다음 코드 예시에서는 `create-identity-source` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

자격 증명 소스를 생성하는 방법

다음 `create-identity-source` 예시에서는 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.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"
}
```

ID 소스에 대한 자세한 내용은 Amazon Verified Permissions 사용 설명서의 Amazon Verified Permissions 정책을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateIdentitySource를 참조하세요.

create-policy-store

다음 코드 예시에서는 create-policy-store 코드를 사용하는 방법을 보여줍니다.

AWS CLI

정책 저장소 생성

다음 create-policy-store 예시에서는 현재 AWS 리전에 정책 저장소를 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 저장소에 대한 자세한 내용은 Amazon Verified Permissions 사용 설명서의 Amazon Verified Permissions 정책을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreatePolicyStore를 참조하세요.

`create-policy-template`

다음 코드 예시는 `create-policy-template`의 사용 방법을 보여 줍니다.

AWS CLI

정책 템플릿을 생성하려면

다음 예에서는 보안 주체에 대한 자리 표시자가 있는 정책 템플릿을 생성합니다.

```
aws verifiedpermissions create-policy-template \
    --statement file://template1.txt \
    --policy-store-id PSEXAMPLEabcdefg111111
```

`template1.txt`의 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 정책을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreatePolicyTemplate을 참조하세요.

create-policy

다음 코드 예시에서는 `create-policy` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

예저 1: 정적 정책 생성

다음 `create-policy` 예시에서는 보안 주체와 리소스를 모두 지정하는 정책 범위가 있는 정적 정책을 만듭니다.

```
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"
    }
}
```

예시 2: 모든 사용자에게 리소스에 대한 액세스 권한을 부여하는 정적 정책 생성

다음 `create-policy` 예시에서는 리소스만 지정하는 정책 범위를 가진 정적 정책을 만듭니다.

```
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"
    }
}
```

예시 3: 지정된 템플릿과 연결된 템플릿 연결 정책 생성

다음 `create-policy` 예시에서는 지정된 정책 템플릿을 사용하여 템플릿 연결 정책을 만들고 사용할 지정된 보안 주체를 새 템플릿 연결 정책과 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.

- API 세부 정보는 AWS CLI 명령 참조의 CreatePolicy를 참조하세요.

delete-identity-source

다음 코드 예시에서는 delete-identity-source 코드를 사용하는 방법을 보여줍니다.

AWS CLI

자격 증명 소스 삭제

다음 delete-identity-source 예시에서는 지정된 ID를 가진 ID 소스를 삭제합니다.

```
aws verifiedpermissions delete-identity-source \
    --identity-source-id ISEXAMPLEabcdefg111111 \
    --policy-store-id PSEXAMPLEabcdefg111111
```

이 명령은 출력을 생성하지 않습니다.

ID 소스에 대한 자세한 내용은 Amazon Verified Permissions 사용 설명서의 Amazon Verified Permissions 정책을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteIdentitySource를 참조하세요.

delete-policy-store

다음 코드 예시에서는 delete-policy-store 코드를 사용하는 방법을 보여줍니다.

AWS CLI

정책 저장소 삭제

다음 delete-policy-store 예시에서는 지정된 ID를 가진 정책 저장소를 삭제합니다.

```
aws verifiedpermissions delete-policy-store \
    --policy-store-id PSEXAMPLEabcdefg111111
```

이 명령은 출력을 생성하지 않습니다.

정책 저장소에 대한 자세한 내용은 Amazon Verified Permissions 사용 설명서의 Amazon Verified Permissions 정책을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeletePolicyStore를 참조하세요.

delete-policy-template

다음 코드 예시에서는 `delete-policy-template` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

정책 작업 템플릿 삭제

다음 `delete-policy-template` 예시에서는 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 정책을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeletePolicyTemplate을 참조하세요.

delete-policy

다음 코드 예시에서는 `delete-policy` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

정적 또는 템플릿 연결 정책 삭제

다음 `delete-policy` 예시에서는 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.

- API 세부 정보는 AWS CLI 명령 참조의 DeletePolicy를 참조하세요.

get-identity-source

다음 코드 예시에서는 `get-identity-source` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

자격 증명 소스에 대한 세부 정보 검색

다음 `get-identity-source` 예시에서는 지정된 ID를 가진 ID 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"
}
```

ID 소스에 대한 자세한 내용은 Amazon Verified Permissions 사용 설명서의 Amazon Verified Permissions 정책을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetIdentitySource를 참조하세요.

get-policy-store

다음 코드 예시에서는 get-policy-store 코드를 사용하는 방법을 보여줍니다.

AWS CLI

정책 저장소 관련 세부 정보 검색

다음 get-policy-store 예시에서는 지정된 ID를 가진 정책 저장소에 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 저장소에 대한 자세한 내용은 Amazon Verified Permissions 사용 설명서의 Amazon Verified Permissions 정책을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetPolicyStore를 참조하세요.

get-policy-template

다음 코드 예시에서는 get-policy-template 코드를 사용하는 방법을 보여줍니다.

AWS CLI

정책 템플릿에 대한 세부 정보를 검색하는 방법

다음 get-policy-template 예시에서는 지정된 ID를 가진 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 정책을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetPolicyTemplate을 참조하세요.

get-policy

다음 코드 예시에서는 `get-policy` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

정책 관련 세부 정보 검색

다음 `get-policy` 예시에서는 지정된 ID를 가진 정책에 대한 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.

- API 세부 정보는 AWS CLI 명령 참조의 GetPolicy를 참조하세요.

get-schema

다음 코드 예시에서는 get-schema 코드를 사용하는 방법을 보여줍니다.

AWS CLI

정책 저장소에서 스키마 검색

다음 get-schema 예시에서는 지정된 정책 저장소에 있는 스키마의 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 저장소 스키마를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetSchema를 참조하세요.

is-authorized-with-token

다음 코드 예시에서는 is-authorized-with-token 코드를 사용하는 방법을 보여줍니다.

AWS CLI

예시 1: 사용자 요청에 대한 권한 부여 결정을 요청하는 방법(허용)

다음 is-authorized-with-token 예시에서는 Amazon Cognito로 인증된 사용자에 대한 권한 부여 결정을 요청합니다. 요청은 액세스 토큰이 아닌 Cognito에서 제공하는 자격 증명 토큰을 사용합니다. 이 예시에서는 지정된 정보 저장소가 보안 주체를 CognitoUser 유형의 엔터티로 반환하도록 구성됩니다.

```
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 저장소에는 지정된 Cognito 사용자 풀 및 애플리케이션 Id의 ID를 허용하는 다음 문이 포함된 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 ID를 사용하는 방법에 대한 자세한 내용은 Amazon Verified Permissions 사용 설명서의 ID 제공업체와 함께 Amazon Verified Permissions 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 IsAuthorizedWithToken을 참조하세요.

is-authorized

다음 코드 예시에서는 is-authorized 코드를 사용하는 방법을 보여줍니다.

AWS CLI

예시 1: 사용자 요청에 대한 권한 부여 결정을 요청하는 방법(허용)

다음 is-authorized 예시는 Photo 유형의 리소스— VacationPhoto94.jpg에 대해 updatePhoto 작업을 수행하려는 User 유형의 보안 주체, Alice에 대한 권한 부여 결정을 요청합니다.

응답은 요청이 하나의 정책에 의해 허용되었음을 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": []
}
```

예시 2: 사용자 요청에 대한 권한 부여 결정을 요청하는 방법(거부)

다음 예시에서는 보안 주체가 User::"Bob"라는 점을 제외하고 이전 예시와 동일합니다. 정책 저장소에 해당 사용자가 Album::"alice_folder"에 액세스할 수 있도록 허용하는 정책이 포함되어 있지 않습니다.

출력은 DeterminingPolicies 목록이 비어 있기 때문에 Deny가 암시적임을 나타냅니다.

```
aws verifiedpermissions create-policy \
    --definition file://definition2.txt \
    --policy-store-id PSEXAMPLEabcdefg111111
```

출력:

```
{
    "decision": "DENY",
    "determiningPolicies": [],
    "errors": []
}
```

자세한 내용은 Amazon Verified Permissions 사용 설명서를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 IsAuthorized를 참조하세요.

list-identity-sources

다음 코드 예시에서는 list-identity-sources 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용 가능한 자격 증명 소스를 나열하는 방법

다음 list-identity-sources 예시에서는 지정된 정책 저장소의 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"
        }
    ]
}
```

ID 소스에 대한 자세한 내용은 Amazon Verified Permissions 사용 설명서의 Amazon Verified Permissions 정책을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListIdentitySources를 참조하세요.

list-policies

다음 코드 예시에서는 `list-policies` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용 가능한 정책을 나열하는 방법

다음 `list-policies` 예시에서는 지정된 정책 저장소의 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.

- API 세부 정보는 AWS CLI 명령 참조의 ListPolicies를 참조하세요.

list-policy-stores

다음 코드 예시에서는 `list-policy-stores` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용 가능한 정책 저장소 나열

다음 `list-policy-stores` 예시에서는 AWS 리전의 모든 정책 저장소를 나열합니다. Verified Permissions에 대한 모든 명령 중 `create-policy-store` 및 `list-policy-stores`를 제외한 모든 명령은 작업하려는 정책 저장소의 ID를 지정해야 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 저장소에 대한 자세한 내용은 Amazon Verified Permissions 사용 설명서의 Amazon Verified Permissions 정책을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListPolicyStores 섹션을 참조하세요.

`list-policy-templates`

다음 코드 예시에서는 `list-policy-templates` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용 가능한 정책 템플릿을 나열하는 방법

다음 `list-policy-templates` 예시에서는 지정된 정책 저장소의 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 정책을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListPolicyTemplates를 참조하세요.

put-schema

다음 코드 예시에서는 `put-schema` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

정책 저장소에 저장

다음 `put-schema` 예시에서는 지정된 정책 저장소의 스키마를 생성하거나 바꿉니다.

입력 파일의 `cedarJson` 파라미터는 JSON 객체를 문자열로 표현합니다. 가장 바깥쪽 따옴표 페어 안에 따옴표(")가 포함되어 있습니다. 이렇게 하려면 포함된 모든 따옴표 앞에 백슬래시 문자(“)를 붙이고 모든 줄을 줄 바꿈 없이 하나의 텍스트 줄로 결합하여 JSON을 문자열로 변환해야 합니다.

가독성을 위해 여기에 여러 줄에 걸쳐 예시 문자열을 표시할 수 있지만, 작업을 수행하려면 파라미터를 단일 줄 문자열로 제출해야 합니다.

aws verifiedpermissions put-schema --definition file://schema.txt --policy-store-id PSEXAMPLEabcdefg111111

`schema.txt`의 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 저장소 스키마를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutSchema를 참조하세요.

`update-identity-source`

다음 코드 예시에서는 `update-identity-source` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

자격 증명 소스를 업데이트하는 방법

다음 `update-identity-source` 예시에서는 새 Cognito 사용자 풀 구성을 제공하고 ID 소스에서 반환한 엔터티 유형을 변경하여 지정된 ID 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.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"
}
```

ID 소스에 대한 자세한 내용은 Amazon Verified Permissions 사용 설명서의 Amazon Verified Permissions 정책을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateIdentitySource를 참조하세요.

`update-policy-store`

다음 코드 예시에서는 `update-policy-store` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

정책 저장소 업데이트

다음 `update-policy-store` 예시에서는 유효성 검사 설정을 변경하여 정책 저장소를 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 저장소에 대한 자세한 내용은 Amazon Verified Permissions 사용 설명서의 Amazon Verified Permissions 정책을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdatePolicyStore를 참조하세요.

update-policy-template

다음 코드 예시에서는 `update-policy-template` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

예시 1: 정책 템플릿 업데이트

다음 `update-policy-template` 예시에서는 템플릿에 연결된 지정된 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 정책을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdatePolicyTemplate을 참조하세요.

update-policy

다음 코드 예시는 update-policy의 사용 방법을 보여 줍니다.

AWS CLI

정적 정책을 생성하려면

다음 update-policy 예제에서는 설명과 문을 업데이트하여 기존 정적 정책을 수정합니다.

```
aws verifiedpermissions update-policy \
    --policy-id SPEXAMPLEabcdefg111111 \
    --definition file://updated-definition.txt \
    --policy-store-id PSEXAMPLEabcdefg111111
```

statement 파라미터는 JSON 객체를 문자열로 표현합니다. 가장 바깥쪽 따옴표 페어 안에 따옴표 (")가 포함되어 있습니다. 이렇게 하려면 포함된 모든 따옴표 앞에 백슬래시 문자(“)를 붙이고 모든 줄을 줄 바꿈 없이 하나의 텍스트 줄로 결합하여 JSON을 문자열로 변환해야 합니다.

가독성을 위해 여러 줄에 걸쳐 예제 문자열을 표시할 수 있지만, 해당 작업을 수행하려면 파라미터를 단일 줄 문자열로 제출해야 합니다.

updated-definition.txt 파일의 콘텐츠:

```
{
    "static": {
        "description": "Updated policy to grant janeFriends UserGroup access to the
 vacationFolder Album with view action only",
        "statement": "permit(principal in UserGroup::\"janeFriends\", action ==
 Action::\"view\", 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"
        }
}
```

정책에 대한 자세한 내용은 Amazon Verified Permissions 사용 설명서의 Amazon Verified Permissions 정책을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdatePolicy를 참조하세요.

AWS CLI를 사용한 VPC Lattice 예시

다음 코드 예시는 VPC Lattice와 함께 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`create-listener`

다음 코드 예시에서는 `create-listener`의 사용 방법을 보여줍니다.

AWS CLI

리스너 생성

다음 `create-listener` 예시에서는 트래픽을 지정된 VPC Lattice 대상 그룹에 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 사용자 안내서의 리스너를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateListener를 참조하세요.

create-resource-configuration

다음 코드 예시는 `create-resource-configuration`의 사용 방법을 보여 줍니다.

AWS CLI

리소스 구성 만들기

다음 `create-resource-configuration` 예제에서는 단일 IPv4 주소를 지정하는 리소스 구성을 만듭니다.

```
aws vpc-lattice create-resource-configuration \
    --name my-resource-config \
    --type SINGLE \
    --resource-gateway-identifier rgw-0bba03f3d56060135 \
    --resource-configuration-definition 'ipResource={ipAddress=10.0.14.85}'
```

출력:

```
{
    "allowAssociationToShareableServiceNetwork": true,
    "arn": "arn:aws:vpc-lattice:us-east-1:123456789012:resourceconfiguration/
rcfg-07129f3acded87625",
    "id": "rcfg-07129f3acded87625",
    "name": "my-resource-config",
    "portRanges": [
        "1-65535"
    ],
    "protocol": "TCP",
    "resourceConfigurationDefinition": {
        "ipResource": {
            "ipAddress": "10.0.14.85"
        }
    },
    "resourceGatewayId": "rgw-0bba03f3d56060135",
    "status": "ACTIVE",
    "type": "SINGLE"
}
```

자세한 내용은 Amazon VPC Lattice 사용 설명서의 [Resource configurations for VPC resources](#) 섹션을 참조하시기 바랍니다.

- API 세부 정보는 AWS CLI 명령 참조의 [CreateResourceConfiguration](#) 섹션을 참조하시기 바랍니다.

create-resource-gateway

다음 코드 예시는 `create-resource-gateway`의 사용 방법을 보여 줍니다.

AWS CLI

리소스 게이트웨이 만들기

다음 `create-resource-gateway` 예제에서는 지정된 서브넷에 대한 리소스 게이트웨이를 만듭니다.

```
aws vpc-lattice create-resource-gateway \
    --name my-resource-gateway \
    --vpc-identifier vpc-0bf4c2739bc05a69 \
    --subnet-ids subnet-08e8943905b63a683
```

출력:

```
{
    "arn": "arn:aws:vpc-lattice:us-east-1:123456789012:resourcegateway/
rgw-0bba03f3d56060135",
    "id": "rgw-0bba03f3d56060135",
    "ipAddressType": "IPV4",
    "name": "my-resource-gateway",
    "securityGroupIds": [
        "sg-087ffd596c5fe962c"
    ],
    "status": "ACTIVE",
    "subnetIds": [
        "subnet-08e8943905b63a683"
    ],
    "vpcIdentifier": "vpc-0bf4c2739bc05a694"
}
```

자세한 내용은 Amazon VPC Lattice 사용 설명서의 [Resource gateways in VPC Lattice](#) 섹션을 참조하시기 바랍니다.

- API 세부 정보는 AWS CLI 명령 참조의 CreateResourceGateway 섹션을 참조하시기 바랍니다.

create-service-network-service-association

다음 코드 예시는 create-service-network-service-association의 사용 방법을 보여 줍니다.

AWS CLI

서비스 연결 생성

다음 create-service-network-service-association 예시에서는 지정된 서비스를 지정된 서비스 네트워크에 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 사용자 안내서의 서비스 연결 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateServiceNetworkServiceAssociation을 참조하세요.

create-service-network-vpc-association

다음 코드 예시에서는 create-service-network-vpc-association의 사용 방법을 보여줍니다.

AWS CLI

VPC 연결 생성

다음 `create-service-network-vpc-association` 예시에서는 지정된 VPC를 지정된 서비스 네트워크에 연결합니다. 지정된 보안 그룹은 서비스 네트워크 및 해당 서비스에 액세스할 수 있는 VPC의 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 사용자 안내서의 VPC 연결 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateServiceNetworkVpcAssociation을 참조하세요.

`create-service-network`

다음 코드 예시에서는 `create-service-network`의 사용 방법을 보여줍니다.

AWS CLI

서비스 네트워크 생성

다음 `create-service-network` 예시에서는 지정된 이름을 사용하여 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 사용자 안내서의 서비스 네트워크를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateServiceNetwork를 참조하세요.

`create-service`

다음 코드 예시에서는 `create-service`의 사용 방법을 보여줍니다.

AWS CLI

서비스 생성

다음 `create-service` 예시에서는 지정된 이름을 사용하여 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 사용자 안내서의 [VPC Lattice 서비스](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [CreateService](#)를 참조하세요.

create-target-group

다음 코드 예시에서는 `create-target-group`의 사용 방법을 보여줍니다.

AWS CLI

예시 1: INSTANCE 유형의 대상 그룹 생성

다음 `create-target-group` 예시에서는 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"
}
```

예시 2: IP 유형의 대상 그룹 생성

다음 `create-target-group` 예시에서는 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"
}
```

예시 3: LAMBDA 유형의 대상 그룹 생성

다음 `create-target-group` 예시에서는 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"
}
```

예시 4: ALB 유형의 대상 그룹 생성

다음 `create-target-group` 예시에서는 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 사용자 안내서의 대상 그룹을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateTargetGroup을 참조하세요.

delete-auth-policy

다음 코드 예시에서는 delete-auth-policy의 사용 방법을 보여줍니다.

AWS CLI

인증 정책 삭제

다음 delete-auth-policy 예시에서는 지정된 서비스의 인증 정책을 삭제합니다.

```
aws vpc-lattice delete-auth-policy \
    --resource-identifier svc-0285b53b2eEXAMPLE
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon VPC Lattice 사용자 안내서의 인증 정책을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteAuthPolicy를 참조하세요.

delete-listener

다음 코드 예시에서는 delete-listener의 사용 방법을 보여줍니다.

AWS CLI

리스너를 삭제하려면

다음 delete-listener 예시에서는 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 사용자 안내서의 리스너를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteListener를 참조하세요.

delete-resource-configuration

다음 코드 예시는 delete-resource-configuration의 사용 방법을 보여 줍니다.

AWS CLI

리소스 구성 삭제

다음 delete-resource-configuration 예제에서는 지정한 리소스 구성을 삭제합니다.

```
aws vpc-lattice delete-resource-configuration \
    --resource-configuration-identifier rcfg-07129f3acded87625
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon VPC Lattice 사용 설명서의 [Resource gateways in VPC Lattice](https://docs.aws.amazon.com/vpc-lattice/latest/ug/resource-gateway.html) 섹션을 참조하시기 바랍니다.

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteResourceConfiguration](https://awscli.amazonaws.com/v2/documentation/api/latest/reference/vpc-lattice/delete-resource-configuration.html) 섹션을 참조하시기 바랍니다.

delete-resource-gateway

다음 코드 예시는 delete-resource-gateway의 사용 방법을 보여 줍니다.

AWS CLI

리소스 게이트웨이 삭제

다음 delete-resource-gateway 예제에서는 지정된 리소스 게이트웨이를 삭제합니다.

```
aws vpc-lattice delete-resource-gateway \
    --resource-gateway-identifier rgw-0bba03f3d56060135
```

출력:

```
{
    "arn": "arn:aws:vpc-lattice:us-east-1:123456789012:resourcegateway/
rgw-0bba03f3d56060135",
    "id": "rgw-0bba03f3d56060135",
    "name": "my-resource-gateway",
    "status": "DELETE_IN_PROGRESS"
```

```
}
```

자세한 내용은 Amazon VPC Lattice 사용 설명서의 Resource gateways in VPC Lattice 섹션을 참조하시기 바랍니다.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteResourceGateway 섹션을 참조하시기 바랍니다.

delete-service-network-service-association

다음 코드 예시는 delete-service-network-service-association의 사용 방법을 보여 줍니다.

AWS CLI

서비스 연결 삭제

다음 delete-service-network-service-association 예시에서는 지정된 서비스 연결을 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 사용자 안내서의 서비스 연결 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteServiceNetworkServiceAssociation을 참조하세요.

delete-service-network-vpc-association

다음 코드 예시에서는 delete-service-network-vpc-association의 사용 방법을 보여줍니다.

AWS CLI

VPC 연결 삭제

다음 delete-service-network-vpc-association 예시에서는 지정된 VPC 연결을 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 사용자 안내서의 VPC 연결 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteServiceNetworkVpcAssociation을 참조하세요.

delete-service-network

다음 코드 예시에서는 delete-service-network의 사용 방법을 보여줍니다.

AWS CLI

서비스 네트워크 삭제

다음 delete-service-network 예시에서는 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 사용자 안내서의 서비스 네트워크를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteServiceNetwork를 참조하세요.

delete-service

다음 코드 예시에서는 delete-service의 사용 방법을 보여줍니다.

AWS CLI

서비스 삭제

다음 `delete-service` 예시에서는 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 사용자 안내서의 VPC Lattice 서비스를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteService를 참조하세요.

delete-target-group

다음 코드 예시에서는 `delete-target-group`의 사용 방법을 보여줍니다.

AWS CLI

대상 그룹 삭제

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 사용자 안내서의 대상 그룹을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteTargetGroup을 참조하세요.

deregister-targets

다음 코드 예시에서는 deregister-targets의 사용 방법을 보여줍니다.

AWS CLI

대상 등록 취소

다음 deregister-targets 예시에서는 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 사용자 안내서의 대상 등록을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeregisterTargets를 참조하세요.

get-auth-policy

다음 코드 예시에서는 get-auth-policy의 사용 방법을 보여줍니다.

AWS CLI

인증 정책 정보 가져오기

다음 `get-auth-policy` 예시에서는 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 사용자 안내서의 인증 정책을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetAuthPolicy를 참조하세요.

`get-listener`

다음 코드 예시에서는 `get-listener`의 사용 방법을 보여줍니다.

AWS CLI

서비스 리스너 정보 가져오기

다음 `get-listener` 예시에서는 지정된 서비스의 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 사용자 안내서의 라우팅 정의를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetListener를 참조하세요.

get-resource-configuration

다음 코드 예시는 `get-resource-configuration`의 사용 방법을 보여 줍니다.

AWS CLI

리소스 구성에 관한 정보 가져오기

다음 `get-resource-configuration` 예제에서는 지정된 구성에 관한 정보를 가져옵니다.

```
aws vpc-lattice get-resource-configuration \
    --resource-configuration-identifier rcfg-07129f3acded87625
```

출력:

```
{
```

```
    "allowAssociationToShareableServiceNetwork": true,
    "amazonManaged": false,
    "arn": "arn:aws:vpc-lattice:us-east-1:123456789012:resourceconfiguration/
rcfg-07129f3acded87625",
    "createdAt": "2025-02-01T00:57:35.871000+00:00",
    "id": "rcfg-07129f3acded87625",
    "lastUpdatedAt": "2025-02-01T00:57:46.874000+00:00",
    "name": "my-resource-config",
    "portRanges": [
        "1-65535"
    ],
    "protocol": "TCP",
    "resourceConfigurationDefinition": {
        "ipResource": {
            "ipAddress": "10.0.14.85"
        }
    },
    "resourceGatewayId": "rgw-0bba03f3d56060135",
    "status": "ACTIVE",
    "type": "SINGLE"
}
```

자세한 내용은 Amazon VPC Lattice 사용 설명서의 Resource gateways in VPC Lattice 섹션을 참조하시기 바랍니다.

- API 세부 정보는 AWS CLI 명령 참조의 GetResourceConfiguration 섹션을 참조하시기 바랍니다.

get-resource-gateway

다음 코드 예시는 `get-resource-gateway`의 사용 방법을 보여 줍니다.

AWS CLI

리소스 게이트웨이에 관한 정보 가져오기

다음 `get-resource-gateway` 예제에서는 지정된 리소스 게이트웨이에 관한 정보를 가져옵니다.

```
aws vpc-lattice get-resource-gateway \
    --resource-gateway-identifier rgw-0bba03f3d56060135
```

출력:

```
{
```

```
    "arn": "arn:aws:vpc-lattice:us-east-1:123456789012:resourcegateway/
rgw-0bba03f3d56060135",
    "createdAt": "2025-02-01T00:57:33.241000+00:00",
    "id": "rgw-0bba03f3d56060135",
    "ipAddressType": "IPV4",
    "lastUpdatedAt": "2025-02-01T00:57:44.351000+00:00",
    "name": "my-resource-gateway",
    "securityGroupIds": [
        "sg-087ffd596c5fe962c"
    ],
    "status": "ACTIVE",
    "subnetIds": [
        "subnet-08e8943905b63a683"
    ],
    "vpcId": "vpc-0bf4c2739bc05a694"
}
```

자세한 내용은 Amazon VPC Lattice 사용 설명서의 Resource gateways in VPC Lattice 섹션을 참조하시기 바랍니다.

- API 세부 정보는 AWS CLI 명령 참조의 GetResourceGateway 섹션을 참조하시기 바랍니다.

get-service-network-service-association

다음 코드 예시는 `get-service-network-service-association`의 사용 방법을 보여 줍니다.

AWS CLI

서비스 연결 정보 가져오기

다음 `get-service-network-service-association` 예시에서는 지정된 서비스 연결의 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 사용자 안내서의 서비스 연결 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetServiceNetworkServiceAssociation을 참조하세요.

get-service-network-vpc-association

다음 코드 예시에서는 get-service-network-vpc-association의 사용 방법을 보여줍니다.

AWS CLI

VPC 연결 정보 가져오기

다음 get-service-network-vpc-association 예시에서는 지정된 VPC 연결의 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 사용자 안내서의 VPC 연결 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetServiceNetworkVpcAssociation을 참조하세요.

`get-service-network`

다음 코드 예시에서는 `get-service-network`의 사용 방법을 보여줍니다.

AWS CLI

서비스 네트워크 정보 가져오기

다음 `get-service-network` 예시에서는 지정된 서비스 네트워크의 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 사용자 안내서의 서비스 네트워크를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetServiceNetwork를 참조하세요.

get-service

다음 코드 예시에서는 get-service의 사용 방법을 보여줍니다.

AWS CLI

서비스 정보 가져오기

다음 get-service 예시에서는 지정된 서비스의 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 사용자 안내서의 서비스를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetService를 참조하세요.

get-target-group

다음 코드 예시에서는 get-target-group의 사용 방법을 보여줍니다.

AWS CLI

대상 그룹 정보 가져오기

다음 get-target-group 예시에서는 대상 유형이 INSTANCE인 지정된 대상 그룹의 정보를 가져옵니다.

```
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 사용자 안내서의 대상 그룹을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetTargetGroup을 참조하세요.

list-listeners

다음 코드 예시에서는 list-listeners의 사용 방법을 보여줍니다.

AWS CLI

서비스 리스너 나열

다음 list-listeners 예시에서는 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 사용자 안내서의 라우팅 정의를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListListeners를 참조하세요.

list-resource-configurations

다음 코드 예시는 list-resource-configurations의 사용 방법을 보여 줍니다.

AWS CLI

리소스 구성 나열

다음 `list-resource-configurations` 예제에서는 리소스 구성을 나열합니다.

```
aws vpc-lattice list-resource-configurations
```

출력:

```
{
    "items": [
        {
            "amazonManaged": false,
            "arn": "arn:aws:vpc-lattice:us-
east-1:123456789012:resourceconfiguration/rcfg-07129f3acded87625",
            "createdAt": "2025-02-01T00:57:35.871000+00:00",
            "id": "rcfg-07129f3acded87625",
            "lastUpdatedAt": "2025-02-01T00:57:46.874000+00:00",
            "name": "my-resource-config",
            "resourceGatewayId": "rgw-0bba03f3d56060135",
            "status": "ACTIVE",
            "type": "SINGLE"
        }
    ]
}
```

자세한 내용은 Amazon VPC Lattice 사용 설명서의 Resource configurations 섹션을 참조하시기 바랍니다.

- API 세부 정보는 AWS CLI 명령 참조의 ListResourceConfigurations 섹션을 참조하시기 바랍니다.

`list-resource-endpoint-associations`

다음 코드 예시는 `list-resource-endpoint-associations`의 사용 방법을 보여 줍니다.

AWS CLI

VPC 엔드포인트 연결 나열

다음 `list-resource-endpoint-associations` 예제에서는 지정된 리소스 구성에 연결된 VPC 엔드포인트를 나열합니다.

```
aws vpc-lattice list-resource-endpoint-associations \
    --resource-configuration-identifier rcfg-07129f3acded87625
```

출력:

```
{
    "items": [
        {
            "arn": "arn:aws:vpc-lattice:us-
east-1:123456789012:resourceendpointassociation/rea-0956a7435baf89326",
            "createdAt": "2025-02-01T00:57:38.998000+00:00",
            "id": "rea-0956a7435baf89326",
            "resourceConfigurationArn": "arn:aws:vpc-lattice:us-
east-1:123456789012:resourceconfiguration/rcfg-07129f3acded87625",
            "resourceConfigurationId": "rcfg-07129f3acded87625",
            "vpcEndpointId": "vpce-019b90d6f16d4f958",
            "vpcEndpointOwner": "123456789012"
        }
    ]
}
```

자세한 내용은 Amazon VPC Lattice 사용 설명서의 Manage associations for a VPC Lattice resource configuration 섹션을 참조하시기 바랍니다.

- API 세부 정보는 AWS CLI 명령 참조의 ListResourceEndpointAssociations 섹션을 참조하시기 바랍니다.

`list-resource-gateways`

다음 코드 예시는 `list-resource-gateways`의 사용 방법을 보여 줍니다.

AWS CLI

리소스 게이트웨이 나열

다음 `list-resource-gateways` 예제에서는 리소스 게이트웨이를 나열합니다.

```
aws vpc-lattice list-resource-gateways
```

출력:

```
{
    "items": [
        {
            "arn": "arn:aws:vpc-lattice:us-east-1:123456789012:resourcegateway/
rgw-0bba03f3d56060135",
            "createdAt": "2025-02-01T00:57:33.241000+00:00",
            "id": "rgw-0bba03f3d56060135",
            "ipAddressType": "IPV4",
            "lastUpdatedAt": "2025-02-01T00:57:44.351000+00:00",
            "name": "my-resource-gateway",
            "seurityGroupIds": [
                "sg-087ffd596c5fe962c"
            ],
            "status": "ACTIVE",
            "subnetIds": [
                "subnet-08e8943905b63a683"
            ],
            "vpcIdentifier": "vpc-0bf4c2739bc05a694"
        }
    ]
}
```

자세한 내용은 Amazon VPC Lattice 사용 설명서의 Resource gateways in VPC Lattice 섹션을 참조하시기 바랍니다.

- API 세부 정보는 AWS CLI 명령 참조의 ListResourceGateways를 참조하시기 바랍니다.

`list-service-network-service-associations`

다음 코드 예시는 `list-service-network-service-associations`의 사용 방법을 보여 줍니다.

AWS CLI

서비스 연결 나열

다음 `list-service-network-service-associations` 예시에서는 지정된 서비스 네트워크의 서비스 연결을 나열합니다. `--query` 옵션은 출력 범위를 서비스 연결의 ID로 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 사용자 안내서의 서비스 연결 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListServiceNetworkServiceAssociations를 참조하세요.

`list-service-network-vpc-associations`

다음 코드 예시에서는 `list-service-network-vpc-associations`의 사용 방법을 보여줍니다.

AWS CLI

VPC 연결 나열

다음 `list-service-network-vpc-associations` 예시에서는 지정된 서비스 네트워크의 VPC 연결을 나열합니다. `--query` 옵션은 출력 범위를 VPC 연결의 ID로 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 사용자 안내서의 VPC 연결 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListServiceNetworkVpcAssociations를 참조하세요.

`list-service-network-vpc-endpoint-associations`

다음 코드 예시는 `list-service-network-vpc-endpoint-associations`의 사용 방법을 보여줍니다.

AWS CLI

VPC 엔드포인트 연결 나열

다음 `list-service-network-vpc-endpoint-associations` 예제에서는 지정된 서비스 네트워크와 연결된 VPC 엔드포인트를 나열합니다.

```
aws vpc-lattice list-service-network-vpc-endpoint-associations \
    --service-network-identifier sn-0808d1748faee0c1e
```

출력:

```
{
    "items": [
        {
            "createdAt": "2025-02-01T01:21:36.667000+00:00",
            "serviceNetworkArn": "arn:aws:vpc-lattice:us-
east-1:123456789012:servicenetwork/sn-0808d1748faee0c1e",
            "state": "ACTIVE",
            "vpcEndpointId": "vpce-0cc199f605eaeace7",
            "vpcEndpointOwnerId": "123456789012"
        }
    ]
}
```

자세한 내용은 Amazon VPC Lattice 사용 설명서의 Manage the associations for a VPC Lattice service network를 참조하시기 바랍니다.

- API 세부 정보는 AWS CLI 명령 참조의 ListServiceNetworkVpcEndpointAssociations를 참조하시기 바랍니다.

`list-service-networks`

다음 코드 예시는 `list-service-networks`의 사용 방법을 보여 줍니다.

AWS CLI

서비스 네트워크 나열

다음 `list-service-networks` 예시에서는 직접 호출하는 계정이 소유하거나 해당 계정에 공유된 서비스 네트워크를 나열합니다. `--query` 옵션은 결과 범위를 서비스 네트워크의 Amazon 리소스 이름(ARN)으로 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 사용자 안내서의 서비스 네트워크를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListServiceNetworks를 참조하세요.

`list-services`

다음 코드 예시에서는 `list-services`의 사용 방법을 보여줍니다.

AWS CLI

서비스 나열

다음 `list-services` 예시에서는 직접 호출하는 계정이 소유하거나 해당 계정에 공유된 서비스를 나열합니다. `--query` 옵션은 결과 범위를 서비스의 Amazon 리소스 이름(ARN)으로 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 사용자 안내서의 서비스를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListServices를 참조하세요.

list-target-groups

다음 코드 예시에서는 `list-target-groups`의 사용 방법을 보여줍니다.

AWS CLI

대상 그룹 나열

다음 `list-target-groups` 예시에서는 대상 유형이 LAMBDA인 대상 그룹을 나열합니다.

```
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 사용자 안내서의 대상 그룹을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTargetGroups를 참조하세요.

list-targets

다음 코드 예시에서는 `list-targets`의 사용 방법을 보여줍니다.

AWS CLI

대상 그룹의 대상 나열

다음 `list-targets` 예시에서는 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 사용자 안내서의 대상 그룹을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTargets를 참조하세요.

put-auth-policy

다음 코드 예시에서는 `put-auth-policy`의 사용 방법을 보여줍니다.

AWS CLI

서비스 인증 정책 생성

다음 `put-auth-policy` 예시에서는 지정된 IAM 역할을 사용하는 인증된 위탁자의 요청에 대한 액세스 권한을 부여합니다. 리소스는 정책이 연결된 서비스의 ARN입니다.

```
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 사용자 안내서의 인증 정책을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutAuthPolicy를 참조하세요.

register-targets

다음 코드 예시에서는 register-targets의 사용 방법을 보여줍니다.

AWS CLI

대상 등록

다음 `register-targets` 예시에서는 지정된 대상 그룹에 지정된 대상을 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 사용자 안내서의 [대상 등록](#)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [RegisterTargets](#)를 참조하세요.

AWS CLI를 사용한 AWS WAF Classic 예시

다음 코드 예시에서는 AWS WAF Classic에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

put-logging-configuration

다음 코드 예시는 put-logging-configuration의 사용 방법을 보여 줍니다.

AWS CLI

지정된 Kinesis Firehose 스트림 ARN을 사용하여 웹 ACL ARN에 대한 로깅 구성을 생성하는 방법

다음 put-logging-configuration 예제에서는 CloudFront를 사용한 WAF에 대한 로깅 구성을 표시합니다.

```
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"
        ]
    }
}
```

- API 세부 정보는 AWS CLI 명령 레퍼런스의 PutLoggingConfiguration을 참조하세요.

update-byte-match-set

다음 코드 예시는 update-byte-match-set의 사용 방법을 보여 줍니다.

AWS CLI

바이트 일치 세트를 업데이트하는 방법

다음 update-byte-match-set 명령은 ByteMatchSet에서 ByteMatchTuple 객체(필터)를 삭제합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 레퍼런스의 UpdateByteMatchSet를 참조하세요.

update-ip-set

다음 코드 예시는 update-ip-set의 사용 방법을 보여 줍니다.

AWS CLI

IP 세트를 업데이트하는 방법

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 수도 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 일치 조건 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 레퍼런스의 UpdateIpSet를 참조하세요.

update-rule

다음 코드 예시는 update-rule의 사용 방법을 보여 줍니다.

AWS CLI

규칙을 업데이트하는 방법

다음 update-rule 명령은 규칙에서 Predicate 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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateRule 섹션을 참조하세요.

update-size-constraint-set

다음 코드 예시는 update-size-constraint-set의 사용 방법을 보여 줍니다.

AWS CLI

크기 제약 조건 세트를 업데이트하는 방법

다음 update-size-constraint-set 명령은 크기 제약 조건 세트에서 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.

- API 세부 정보는 AWS CLI 명령 레퍼런스의 UpdateSizeConstraintSet를 참조하세요.

`update-sql-injection-match-set`

다음 코드 예시는 `update-sql-injection-match-set`의 사용 방법을 보여 줍니다.

AWS CLI

SQL 명령어 삽입 일치 세트를 업데이트하는 방법

다음 명령은 SQL 명령어 삽입 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 명령어 삽입 일치 조건 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 레퍼런스의 UpdateSqlInjectionMatchSet를 참조하세요.

`update-web-acl`

다음 코드 예시는 `update-web-acl`의 사용 방법을 보여 줍니다.

AWS CLI

웹 ACL 업데이트

다음 `update-web-acl` 명령은 WebACL 에서 `ActivatedRule` 객체를 삭제합니다.

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 Firewall Manager, AWS Shield 고급 개발자 안내서의 웹 ACL 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateWebAcl 섹션을 참조하세요.

update-xss-match-set

다음 코드 예시는 update-xss-match-set의 사용 방법을 보여 줍니다.

AWS CLI

XSSMatchSet를 업데이트하는 방법

다음 update-xss-match-set 명령은 XssMatchSet에서 XssMatchTuple 객체(필터)를 삭제합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateXssMatchSet 섹션을 참조하세요.

AWS CLI를 사용한 AWS WAF Classic Regional 예시

다음 코드 예시에서는 AWS WAF Classic Regional에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

associate-web-acl

다음 코드 예시는 associate-web-acl의 사용 방법을 보여 줍니다.

AWS CLI

웹 ACL과 리소스 연결

다음 associate-web-acl 명령은 web-acl-id에서 지정한 웹 ACL을 resource-arn에서 지정한 리소스와 연결합니다. 리소스 ARN은 애플리케이션 로드 밸런서 또는 API 게이트웨이를 참조할 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 ACL 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 레퍼런스에서 AssociateWebAcl을 참조하세요.

put-logging-configuration

다음 코드 예시는 put-logging-configuration의 사용 방법을 보여 줍니다.

AWS CLI

지정된 Kinesis Firehose 스트림 ARN을 사용하여 웹 ACL ARN에 대한 로깅 구성 생성

다음 put-logging-configuration 예제에서는 us-east-1 리전에서 ALB/APIGateway를 사용하는 WAF에 대한 로깅 구성을 표시합니다.

```
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"
        ]
    }
}
```

- API 세부 정보는 AWS CLI 명령 레퍼런스의 PutLoggingConfiguration을 참조하세요.

update-byte-match-set

다음 코드 예시는 update-byte-match-set의 사용 방법을 보여 줍니다.

AWS CLI

바이트 일치 세트 업데이트

다음 update-byte-match-set 명령은 ByteMatchSet의 ByteMatchTuple 객체(필터)를 삭제합니다. updates 값에는 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.

- API 세부 정보는 AWS CLI 명령 레퍼런스의 UpdateByteMatchSet를 참조하세요.

update-ip-set

다음 코드 예시는 update-ip-set의 사용 방법을 보여 줍니다.

AWS CLI

IP 세트 업데이트

다음 update-ip-set 명령은 IPv4 주소로 IPSet를 업데이트하고 IPv6 주소를 삭제합니다. change-token 명령을 실행하여 get-change-token의 값을 가져옵니다. 업데이트 값에는 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 수도 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
```

change.json의 콘텐츠

```
[
    {
        "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.

- API 세부 정보는 AWS CLI 명령 레퍼런스의 UpdateIpSet를 참조하세요.

update-rule

다음 코드 예시는 update-rule의 사용 방법을 보여 줍니다.

AWS CLI

규칙 업데이트

다음 update-rule 명령은 규칙에서 Predicate 객체를 삭제합니다. updates 값에는 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.

- API 세부 정보는 AWS CLI 명령 레퍼런스의 UpdateRule을 참조하세요.

update-size-constraint-set

다음 코드 예시는 update-size-constraint-set의 사용 방법을 보여 줍니다.

AWS CLI

크기 제약 조건 세트 업데이트

다음 update-size-constraint-set 명령은 크기 제약 조건 세트에서 SizeConstraint` 객체(필터)를 삭제합니다. updates 값에는 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.

- API 세부 정보는 AWS CLI 명령 레퍼런스의 UpdateSizeConstraintSet를 참조하세요.

update-sql-injection-match-set

다음 코드 예시는 update-sql-injection-match-set의 사용 방법을 보여 줍니다.

AWS CLI

SQL 명령어 삽입 일치 세트 업데이트

다음 update-sql-injection-match-set 명령은 SQL 명령어 삽입 일치 세트의 SqlInjectionMatchTuple 객체(필터)를 삭제합니다. updates 값에는 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"},TextTransformatio

자세한 내용은 AWS WAF 개발자 안내서의 SQL 명령어 삽입 일치 조건 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 레퍼런스의 UpdateSqlInjectionMatchSet를 참조하세요.

update-web-acl

다음 코드 예시는 update-web-acl의 사용 방법을 보여 줍니다.

AWS CLI

웹 ACL 업데이트

다음 update-web-acl 명령은 WebACL 에서 ActivatedRule 객체를 삭제합니다. updates 값에는 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 ACL 작업을 참조하세요.

- API 세부 정보는 AWS CLI 명령 레퍼런스의 UpdateWebAcl을 참조하세요.

`update-xss-match-set`

다음 코드 예시는 `update-xss-match-set`의 사용 방법을 보여 줍니다.

AWS CLI

XSSMatchSet 업데이트

다음 `update-xss-match-set` 명령은 `XssMatchSet`에서 `XssMatchTuple` 객체(필터)를 삭제합니다. `updates` 값에는 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.

- API 세부 정보는 AWS CLI 명령 레퍼런스의 UpdateXssMatchSet를 참조하세요.

AWS CLI를 사용한 AWS WAFV2 예시

다음 코드 예시에서는 AWS WAFV2에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- [작업](#)

작업

associate-web-acl

다음 코드 예시에서는 associate-web-acl 코드를 사용하는 방법을 보여줍니다.

AWS CLI

웹 ACL과 리전 AWS 리소스 연결

다음 associate-web-acl 예시에서는 지정된 웹 ACL을 Application Load Balancer와 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 Firewall Manager, AWS Shield 고급 개발자 안내서의 웹 ACL을 AWS 리소스와 연결 또는 연결 해제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 AssociateWebAcl 섹션을 참조하세요.

check-capacity

다음 코드 예시에서는 check-capacity 코드를 사용하는 방법을 보여줍니다.

AWS CLI

규칙 집합에서 사용하는 용량을 얻으려면

다음 check-capacity는 속도 기반 규칙 문과 중첩된 규칙을 포함하는 AND 규칙 문이 포함된 규칙 세트의 용량 요구 사항을 검색합니다.

```
aws wafv2 check-capacity \
    --scope REGIONAL \
    --rules file://waf-rule-list.json \
    --region us-west-2
```

file://waf-rule-list.json의 콘텐츠:

```
[
    {
        "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 Firewall Manager, AWS Shield 고급 개발자 안내서의 AWS WAF 웹 ACL 용량 단위(WCU)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CheckCapacity 섹션을 참조하세요.

`create-ip-set`

다음 코드 예시에서는 `create-ip-set` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

웹 ACLs 및 규칙 그룹에서 사용할 IP 세트를 생성하는 방법

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 Firewall Manager, AWS Shield 고급 개발자 안내서의 IP 집합 및 정규식 패턴 집합을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateIpSet 섹션을 참조하세요.

`create-regex-pattern-set`

다음 코드 예시에서는 `create-regex-pattern-set` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

웹 ACLs 및 규칙 그룹에서 사용할 정규식 패턴 세트를 생성하는 방법

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 Firewall Manager, AWS Shield 고급 개발자 안내서의 IP 집합 및 정규식 패턴 집합을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateRegexPatternSet 섹션을 참조하세요.

create-rule-group

다음 코드 예시에서는 create-rule-group 코드를 사용하는 방법을 보여줍니다.

AWS CLI

웹 ACLs에 사용할 사용자 지정 규칙 그룹을 생성하는 방법

다음 create-rule-group 명령은 리전에서 사용할 사용자 지정 규칙 그룹을 생성합니다. 그룹의 규칙 문은 JSON 형식 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 콘텐츠:

```
[
    {
        "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 Firewall Manager, AWS Shield 고급 개발자 안내서의 자체 규칙 그룹 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateRuleGroup 섹션을 참조하세요.

`create-web-acl`

다음 코드 예시에서는 `create-web-acl` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

웹 ACL 생성

다음 `create-web-acl` 명령은 리전에서 사용할 웹 ACL을 생성합니다. 웹 ACL에 대한 규칙 문은 JSON 형식의 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 콘텐츠:

```
[
    {
        "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 AWS WAF, AWS Firewall Manager, AWS Shield 고급 개발자 안내서의 웹 액세스 제어 목록(웹 ACL) 관리 및 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateWebAcl 섹션을 참조하세요.

`delete-ip-set`

다음 코드 예시에서는 `delete-ip-set` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

IP 세트 삭제

다음 delete-ip-set는 지정된 IP 세트를 삭제합니다. 이 호출에는 ID가 필요하며, 이는 list-ip-sets 호출에서 얻을 수 있고 잠금 토큰은 list-ip-sets 호출 및 get-ip-set 호출에서 얻을 수 있습니다.

```
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 Firewall Manager, AWS Shield 고급 개발자 안내서의 IP 집합 및 정규식 패턴 집합을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteIpSet 섹션을 참조하세요.

delete-logging-configuration

다음 코드 예시에서는 delete-logging-configuration 코드를 사용하는 방법을 보여줍니다.

AWS CLI

웹 ACL에 대한 로깅 비활성화

다음 delete-logging-configuration은 지정된 웹 ACL에서 모든 로깅 구성을 제거합니다.

```
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 Firewall Manager, AWS Shield 고급 개발자 안내서의 웹 ACL 트래픽 정보 로깅을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteLoggingConfiguration 섹션을 참조하세요.

delete-regex-pattern-set

다음 코드 예시에서는 delete-regex-pattern-set 코드를 사용하는 방법을 보여줍니다.

AWS CLI

정규식 패턴 세트 삭제

다음 delete-regex-pattern-set는 지정된 정규식 패턴 세트의 설정을 업데이트합니다. 이 호출에는 ID가 필요하며, 이는 list-regex-pattern-sets 호출에서 얻을 수 있고 잠금 토큰은 list-regex-pattern-sets 호출 또는 get-regex-pattern-set 호출에서 얻을 수 있습니다.

```
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 Firewall Manager, AWS Shield 고급 개발자 안내서의 IP 집합 및 정규식 패턴 집합을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteRegexPatternSet 섹션을 참조하세요.

delete-rule-group

다음 코드 예시에서는 delete-rule-group 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용자 지정 규칙 그룹 삭제

다음 delete-rule-group은 지정된 사용자 지정 규칙 그룹을 삭제합니다. 이 호출에는 ID가 필요하며, 이는 list-rule-groups 호출에서 얻을 수 있고 잠금 토큰은 list-rule-groups 호출 또는 get-rule-group 호출에서 얻을 수 있습니다.

```
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 Firewall Manager, AWS Shield 고급 개발자 안내서의 자체 규칙 그룹 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteRuleGroup 섹션을 참조하세요.

delete-web-acl

다음 코드 예시에서는 delete-web-acl 코드를 사용하는 방법을 보여줍니다.

AWS CLI

웹 ACL 삭제

다음 delete-web-acl은 계정에서 지정된 웹 ACL을 삭제합니다. 웹 ACL은 리소스와 연결되어 있지 않은 경우에만 삭제할 수 있습니다. 이 호출에는 ID가 필요하며, 이는 list-web-acls 호출에서 얻을 수 있고 잠금 토큰은 list-web-acls 호출 또는 get-web-acl 호출에서 얻을 수 있습니다.

```
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 AWS WAF, AWS Firewall Manager, AWS Shield 고급 개발자 안내서의 웹 액세스 제어 목록(웹 ACL) 관리 및 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteWebAcl 섹션을 참조하세요.

describe-managed-rule-group

다음 코드 예시에서는 describe-managed-rule-group 코드를 사용하는 방법을 보여줍니다.

AWS CLI

관리형 규칙 그룹에 대한 설명을 검색하는 방법

다음 describe-managed-rule-group은 AWS 관리형 규칙 그룹에 대한 설명을 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 Firewall Manager, AWS Shield 고급 개발자 안내서의 [관리형 규칙 그룹](#)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeManagedRuleGroup](#) 섹션을 참조하세요.

disassociate-web-acl

다음 코드 예시에서는 disassociate-web-acl 코드를 사용하는 방법을 보여줍니다.

AWS CLI

리전 AWS 리소스에서 웹 ACL의 연결 해제

다음 disassociate-web-acl 예시에서는 지정된 Application Load Balancer에서 기존 웹 ACL 연결을 모두 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 Firewall Manager, AWS Shield 고급 개발자 안내서의 [웹 ACL을 AWS 리소스와 연결 또는 연결 해제](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DisassociateWebAcl](#) 섹션을 참조하세요.

get-ip-set

다음 코드 예시에서는 get-ip-set 코드를 사용하는 방법을 보여줍니다.

AWS CLI

특정 IP 세트를 검색하는 방법

다음 get-ip-set는 이름, 범위, ID가 지정된 IP 세트를 검색합니다. create-ip-set 및 list-ip-sets 명령에서 IP 세트의 ID를 가져올 수 있습니다.

```
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 Firewall Manager, AWS Shield 고급 개발자 안내서의 IP 집합 및 정규식 패턴 집합을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetIpSet 섹션을 참조하세요.

get-logging-configuration

다음 코드 예시에서는 get-logging-configuration 코드를 사용하는 방법을 보여줍니다.

AWS CLI

웹 ACL에 대한 로깅 구성을 검색하는 방법

다음 `get-logging-configuration`은 지정된 웹 ACL에 대한 로깅 구성을 검색합니다.

```
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 Firewall Manager, AWS Shield 고급 개발자 안내서의 웹 ACL 트래픽 정보 로깅을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetLoggingConfiguration 섹션을 참조하세요.

get-rate-based-statement-managed-keys

다음 코드 예시에서는 `get-rate-based-statement-managed-keys` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

속도 기반 규칙에 따라 차단된 IP 주소 목록 검색

다음 `get-rate-based-statement-managed-keys`는 리전 애플리케이션에 사용 중인 속도 기반 규칙에 따라 차단 중인 IP 주소를 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, AWS Shield 고급 개발자 안내서의 속도 기반 규칙 문을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetRateBasedStatementManagedKeys 섹션을 참조하세요.

`get-regex-pattern-set`

다음 코드 예시에서는 `get-regex-pattern-set` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

특정 정규식 패턴 세트를 검색하는 방법

다음 `get-regex-pattern-set`는 지정된 이름, 범위, 리전 및 ID를 가진 정규식 패턴 세트를 검색합니다. `create-regex-pattern-set` 및 `list-regex-pattern-sets` 명령에서 정규식 패턴 세트의 ID를 가져올 수 있습니다.

```
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 Firewall Manager, AWS Shield 고급 개발자 안내서의 IP 집합 및 정규식 패턴 집합을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetRegexPatternSet 섹션을 참조하세요.

get-rule-group

다음 코드 예시에서는 get-rule-group 코드를 사용하는 방법을 보여줍니다.

AWS CLI

특정 사용자 지정 규칙 그룹을 검색하는 방법

다음 get-rule-group은 이름, 범위, ID가 지정된 사용자 지정 규칙 그룹을 검색합니다. create-rule-group 및 list-rule-groups 명령에서 규칙 그룹의 ID를 가져올 수 있습니다.

```
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 Firewall Manager, AWS Shield 고급 개발자 안내서의 자체 규칙 그룹 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetRuleGroup 섹션을 참조하세요.

get-sampled-requests

다음 코드 예시에서는 `get-sampled-requests` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

웹 ACL에 대한 웹 요청 샘플을 검색하는 방법

다음 `get-sampled-requests`는 지정된 웹 ACL, 규칙 지표 및 기간별로 샘플링된 웹 요청을 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 Firewall Manager, AWS Shield 고급 개발자 안내서의 웹 요청 샘플 보기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetSampledRequests 섹션을 참조하세요.

`get-web-acl-for-resource`

다음 코드 예시에서는 `get-web-acl-for-resource` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS 리소스와 연결된 웹 ACL 검색

다음 `get-web-acl-for-resource`는 지정된 리소스와 연결된 웹 ACL의 JSON을 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 Firewall Manager, AWS Shield 고급 개발자 안내서의 웹 ACL을 AWS 리소스와 연결 또는 연결 해제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetWebAclForResource 섹션을 참조하세요.

get-web-acl

다음 코드 예시에서는 get-web-acl 코드를 사용하는 방법을 보여줍니다.

AWS CLI

웹 ACL을 검색하는 방법

다음 get-web-acl은 이름, 범위, ID가 지정된 웹 ACL을 검색합니다. create-web-acl 및 list-web-acls 명령에서 웹 ACL의 ID를 가져올 수 있습니다.

```
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 AWS WAF, AWS Firewall Manager, AWS Shield 고급 개발자 안내서의 웹 액세스 제어 목록(웹 ACL) 관리 및 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetWebAcl 섹션을 참조하세요.

`list-available-managed-rule-groups`

다음 코드 예시에서는 `list-available-managed-rule-groups` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

관리형 규칙 그룹 검색

다음 `list-available-managed-rule-groups`는 현재 웹 ACL에서 사용할 수 있는 모든 관리형 규칙 그룹 목록을 반환합니다.

```
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 Firewall Manager, AWS Shield 고급 개발자 안내서의 관리형 규칙 그룹을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListAvailableManagedRuleGroups 섹션을 참조하세요.

list-ip-sets

다음 코드 예시에서는 `list-ip-sets` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

IP 세트 목록을 검색하는 방법

다음 `list-ip-sets`는 리전 범위가 있는 계정의 모든 IP 세트를 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 Firewall Manager, AWS Shield 고급 개발자 안내서의 IP 집합 및 정규식 패턴 집합을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListIpSets 섹션을 참조하세요.

`list-logging-configurations`

다음 코드 예시에서는 `list-logging-configurations` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

리전의 모든 로깅 구성 목록을 검색하는 방법

다음 `list-logging-configurations`는 `us-west-2` 리전 내 리전 사용 범위가 지정된 웹 ACL에 대한 모든 로깅 구성을 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 Firewall Manager, AWS Shield 고급 개발자 안내서의 웹 ACL 트래픽 정보 로깅을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListLoggingConfigurations 섹션을 참조하세요.

`list-regex-pattern-sets`

다음 코드 예시에서는 `list-regex-pattern-sets` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

정규식 패턴 세트 목록을 검색하는 방법

다음 `list-regex-pattern-sets`는 `us-west-2` 리전에 정의된 계정의 모든 정규식 패턴 세트를 검색합니다.

```
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 Firewall Manager, AWS Shield 고급 개발자 안내서의 [IP 집합 및 정규식 패턴 집합]을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ListRegexPatternSets] 섹션을 참조하세요.

`list-resources-for-web-acl`

다음 코드 예시에서는 `list-resources-for-web-acl` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

웹 ACL과 연결된 리소스를 검색하는 방법

다음 `list-resources-for-web-acl`은 `us-west-2` 리전에서 지정된 웹 ACL과 연결된 API 게이트웨이 REST API 리소스를 검색합니다.

```
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 Firewall Manager, AWS Shield 고급 개발자 안내서의 웹 ACL을 AWS 리소스와 연결 또는 연결 해제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListResourcesForWebAcl 섹션을 참조하세요.

`list-rule-groups`

다음 코드 예시에서는 `list-rule-groups` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용자 지정 규칙 그룹 목록 검색

다음 `list-rule-groups`는 지정된 범위 및 리전 위치에 대해 계정에 정의된 모든 사용자 지정 규칙 그룹을 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 Firewall Manager, AWS Shield 고급 개발자 안내서의 자체 규칙 그룹 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListRuleGroups 섹션을 참조하세요.

`list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS WAF 리소스의 모든 태그를 검색하는 방법

다음 `list-tags-for-resource`는 지정된 웹 ACL의 모든 태그 키, 값 페어 목록을 검색합니다.

```
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 Firewall Manager, AWS Shield 고급 개발자 안내서의 AWS WAF 시작하기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource 섹션을 참조하세요..

list-web-acls

다음 코드 예시에서는 `list-web-acls` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

범위에 대한 웹 ACLs을 검색하는 방법

다음 `list-web-acls`는 지정된 범위의 계정에 정의된 모든 웹 ACL을 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 AWS WAF, AWS Firewall Manager, AWS Shield 고급 개발자 안내서의 웹 액세스 제어 목록(웹 ACL) 관리 및 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListWebAcls 섹션을 참조하세요.

put-logging-configuration

다음 코드 예시에서는 put-logging-configuration 코드를 사용하는 방법을 보여줍니다.

AWS CLI

웹 ACL에 로깅 구성을 추가하는 방법

다음 put-logging-configuration은 로그에서 삭제된 필드 없이 Amazon Kinesis Data Firehose 로깅 구성 aws-waf-logs-custom-transformation을 지정된 웹 ACL에 추가합니다.

```
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 Firewall Manager, AWS Shield 고급 개발자 안내서의 웹 ACL 트래픽 정보 로깅을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutLoggingConfiguration 섹션을 참조하세요.

tag-resource

다음 코드 예시에서는 tag-resource 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS WAF 리소스에 태그 추가

다음 tag-resource 예시에서는 Name 키가 있고 값이 AWSWAF로 설정된 태그를 지정된 웹 ACL에 추가합니다.

```
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 Firewall Manager, AWS Shield 고급 개발자 안내서의 AWS WAF 시작하기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource 섹션을 참조하세요..

untag-resource

다음 코드 예시에서는 untag-resource 코드를 사용하는 방법을 보여줍니다.

AWS CLI

AWS WAF 리소스에서 태그 제거

다음 untag-resource 예시는 지정된 키가 있는 태그를 KeyName 객체에서 삭제합니다.

```
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 Firewall Manager, AWS Shield 고급 개발자 안내서의 AWS WAF 시작하기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource 섹션을 참조하세요..

update-ip-set

다음 코드 예시에서는 update-ip-set 코드를 사용하는 방법을 보여줍니다.

AWS CLI

기존 IP 세트의 설정을 수정하는 방법

다음 update-ip-set는 지정된 IP 세트의 설정을 업데이트합니다. 이 호출에는 ID가 필요하며, 이는 list-ip-sets 호출에서 얻을 수 있고 잠금 토큰은 list-ip-sets 호출 및 get-ip-set 호출에서 얻을 수 있습니다. 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 Firewall Manager, AWS Shield 고급 개발자 안내서의 IP 집합 및 정규식 패턴 집합을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateIpSet 섹션을 참조하세요.

update-regex-pattern-set

다음 코드 예시에서는 update-regex-pattern-set 코드를 사용하는 방법을 보여줍니다.

AWS CLI

기존 정규식 패턴 세트의 설정을 수정하는 방법

다음 update-regex-pattern-set는 지정된 정규식 패턴 세트의 설정을 업데이트합니다. 이 호출에는 ID가 필요하며, 이는 list-regex-pattern-sets 호출에서 얻을 수 있고 잠금 토큰은 list-regex-pattern-sets 호출 및 get-regex-pattern-set 호출에서 얻을 수 있습니다. 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 Firewall Manager, AWS Shield 고급 개발자 안내서의 IP 집합 및 정규식 패턴 집합을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateRegexPatternSet 섹션을 참조하세요.

update-rule-group

다음 코드 예시에서는 update-rule-group 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용자 지정 규칙 그룹 업데이트

다음 update-rule-group는 호출하면 기존 사용자 지정 규칙 그룹의 표시 여부 구성이 변경됩니다. 이 호출에는 ID가 필요하며, 이는 list-rule-groups 호출에서 얻을 수 있고 잠금 토큰은 list-rule-groups 호출 및 get-rule-group 호출에서 얻을 수 있습니다. 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 Firewall Manager, AWS Shield 고급 개발자 안내서의 자체 규칙 그룹 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateRuleGroup 섹션을 참조하세요.

update-web-acl

다음 코드 예시에서는 update-web-acl 코드를 사용하는 방법을 보여줍니다.

AWS CLI

웹 ACL 업데이트

다음 update-web-acl은 기존 웹 ACL에 대한 설정을 변경합니다. 이 호출에는 ID가 필요하며, 이는 list-web-acls 호출에서 얻을 수 있고 잠금 토큰은 및 기타 설정은 get-web-acl 호출에서 얻을 수 있습니다. 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 AWS WAF, AWS Firewall Manager, AWS Shield 고급 개발자 안내서의 웹 액세스 제어 목록(웹 ACL) 관리 및 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [UpdateWebAcl] 섹션을 참조하세요.

AWS CLI를 사용한 WorkDocs 예제

다음 코드 예제에서는 WorkDocs에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여 줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- [작업]

작업

`abort-document-version-upload`

다음 코드 예시는 `abort-document-version-upload`의 사용 방법을 보여 줍니다.

AWS CLI

문서 버전 업로드를 중지하는 방법

이 예제에서는 이전에 시작된 문서 버전 업로드를 중지합니다.

명령:

```
aws workdocs abort-document-version-upload --document-id feaba64d4efdf271c2521b60a2a44a8f057e84beaabbe22f01267313209835f2 --version-id 1536773972914-ddb67663e782e7ce8455ebc962217cf9f9e47b5a9a702e5c84dcccd417da9313
```

출력:

```
None
```

- API 세부 정보는 AWS CLI 명령 참조의 [AbortDocumentVersionUpload] 섹션을 참조하세요.

`activate-user`

다음 코드 예시는 `activate-user`의 사용 방법을 보여 줍니다.

AWS CLI

사용자 활성화

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"
          }
      }
  }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ActivateUser 섹션을 참조하세요.

add-resource-permissions

다음 코드 예시는 add-resource-permissions의 사용 방법을 보여 줍니다.

AWS CLI

리소스에 대한 권한을 추가하는 방법

이 예제에서는 지정된 보안 주체에 대한 리소스에 권한을 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": ""
      }
  ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 RespondToAuthChallenge 섹션을 참조하세요.

create-comment

다음 코드 예시는 create-comment의 사용 방법을 보여 줍니다.

AWS CLI

새 주석을 추가하는 방법

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"
  }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [CreateOpsItem](https://docs.aws.amazon.com/goto/aws-cli/workdocs-2016-05-01/CreateComment) 섹션을 참조하세요.

`create-custom-metadata`

다음 코드 예시는 `create-custom-metadata`의 사용 방법을 보여 줍니다.

AWS CLI

사용자 지정 메타데이터 업데이트

이 예제에서는 지정된 문서에 대한 사용자 지정 메타데이터를 만듭니다.

명령:

```
aws workdocs create-custom-metadata --resource-
id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65 --custom-
metadata KeyName1=example,KeyName2=example2
```

출력:

```
None
```

- API 세부 정보는 AWS CLI 명령 참조의 [CreateSecurityGroup](#) 섹션을 참조하세요.

`create-folder`

다음 코드 예시는 `create-folder`의 사용 방법을 보여 줍니다.

AWS CLI

폴더 생성

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
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreatePolicy 섹션을 참조하세요.

create-labels

다음 코드 예시는 create-labels의 사용 방법을 보여 줍니다.

AWS CLI

레이블을 생성하는 방법

이 예제에서는 문서에 대한 일련의 레이블을 생성합니다.

명령:

```
aws workdocs create-labels --resource-
id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65 --
labels "documents" "examples" "my_documents"
```

출력:

```
None
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateLabels 섹션을 참조하세요.

create-notification-subscription

다음 코드 예시는 create-notification-subscription의 사용 방법을 보여 줍니다.

AWS CLI

이벤트 알림 구독 생성

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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateNotificationSubscription 섹션을 참조하세요.

create-user

다음 코드 예시는 create-user의 사용 방법을 보여 줍니다.

AWS CLI

새로운 사용자를 생성하는 방법

이 예제에서는 Simple AD 또는 Microsoft AD 디렉터리에 새 사용자를 만듭니다.

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"
            }
        }
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateUser를 참조하세요.

deactivate-user

다음 코드 예시는 deactivate-user의 사용 방법을 보여 줍니다.

AWS CLI

사용자를 비활성화하는 방법

이 예제에서는 활성 사용자를 비활성화합니다.

명령:

```
aws workdocs deactivate-user --user-
id "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c"
```

출력:

```
None
```

- API 세부 정보는 AWS CLI 명령 참조의 [DeactivateUser](#) 섹션을 참조하세요.

delete-comment

다음 코드 예시는 `delete-comment`의 사용 방법을 보여 줍니다.

AWS CLI

문서 버전에서 지정된 주석 삭제

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
```

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteArchive](#) 섹션을 참조하세요.

delete-custom-metadata

다음 코드 예시는 `delete-custom-metadata`의 사용 방법을 보여 줍니다.

AWS CLI

리소스에서 사용자 지정 메타데이터 삭제

이 예제에서는 지정된 리소스에서 모든 사용자 지정 메타데이터를 삭제합니다.

명령:

```
aws workdocs delete-custom-metadata --resource-
id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65 --delete-all
```

출력:

```
None
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteCustomMetadata 섹션을 참조하세요.

delete-document

다음 코드 예시는 delete-document의 사용 방법을 보여 줍니다.

AWS CLI

문서 삭제

이 예시에서는 지정된 문서를 삭제합니다.

명령:

```
aws workdocs delete-document --document-
id b83ed5e5b167b65ef69de9d597627ff1a0d4f07a45e67f1fab7d26b54427de0a
```

출력:

```
None
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteDocument 섹션을 참조하세요.

delete-folder-contents

다음 코드 예시는 delete-folder-contents의 사용 방법을 보여 줍니다.

AWS CLI

폴더의 콘텐츠 삭제

이 예제에서는 지정된 폴더의 콘텐츠를 삭제합니다.

명령:

```
aws workdocs delete-folder-contents --folder-
id 26fa8aa4ba2071447c194f7b150b07149dbdb9e1c8a301872dcd93a4735ce65d
```

출력:

```
None
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteFolderContents 섹션을 참조하세요.

delete-folder

다음 코드 예시는 delete-folder의 사용 방법을 보여 줍니다.

AWS CLI

폴더를 삭제하는 방법

이 예제에서는 지정된 폴더를 삭제합니다.

명령:

```
aws workdocs delete-folder --folder-
id 26fa8aa4ba2071447c194f7b150b07149dbdb9e1c8a301872dcd93a4735ce65d
```

출력:

```
None
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteFolder 섹션을 참조하세요.

delete-labels

다음 코드 예시는 delete-labels의 사용 방법을 보여 줍니다.

AWS CLI

레이블 삭제

이 예제에서는 문서에서 지정된 레이블을 삭제합니다.

명령:

```
aws workdocs delete-labels --resource-
id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65 --
labels "documents" "examples"
```

출력:

```
None
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteLabels 섹션을 참조하세요.

delete-notification-subscription

다음 코드 예시는 delete-notification-subscription의 사용 방법을 보여 줍니다.

AWS CLI

이벤트 알림 구독 삭제

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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteNotificationSubscription 섹션을 참조하세요.

delete-user

다음 코드 예시는 delete-user의 사용 방법을 보여 줍니다.

AWS CLI

사용자 삭제

이 예제에서는 사용자를 삭제합니다.

명령:

```
aws workdocs delete-user --user-
id "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c"
```

출력:

```
None
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteUser를 참조하세요.

describe-activities

다음 코드 예시는 describe-activities의 사용 방법을 보여 줍니다.

AWS CLI

사용자 활동 목록 가져오기

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":
 "DnF1ZXJ5VGhlbkZldGNoAgAAAAAAAS7FmlTaU1OdlFTU1h1UU00VVFIbDlRWHcAAAAAAAAJTRY3bWh5eUgzaVF1ZX
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeActivities](#) 섹션을 참조하세요.

`describe-comments`

다음 코드 예시는 `describe-comments`의 사용 방법을 보여 줍니다.

AWS CLI

지정된 문서 버전에 대한 모든 주석 나열

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"
      }
  ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeComments 섹션을 참조하세요.

describe-document-versions

다음 코드 예시는 describe-document-versions의 사용 방법을 보여 줍니다.

AWS CLI

문서의 버전 검색

이 예제에서는 초기화된 버전과 소스 문서의 URL을 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"
            }
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeDocumentVersions 섹션을 참조하세요.

describe-folder-contents

다음 코드 예시는 describe-folder-contents의 사용 방법을 보여 줍니다.

AWS CLI

폴더의 콘텐츠 설명

이 예제에서는 문서 및 하위 폴더를 포함하여 지정된 폴더의 모든 활성 콘텐츠를 날짜별로 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeFolderContents 섹션을 참조하세요.

describe-groups

다음 코드 예시는 describe-groups의 사용 방법을 보여 줍니다.

AWS CLI

그룹 목록 검색

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 WorkDocs 관리자 안내서의 Amazon WorkDocs로 시작하기를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeGroups 섹션을 참조하세요.

describe-notification-subscriptions

다음 코드 예시는 describe-notification-subscriptions의 사용 방법을 보여 줍니다.

AWS CLI

알림 구독 목록 검색

다음 describe-notification-subscriptions 예제에서는 지정된 Amazon WorkDocs 조직에 대한 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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeNotificationSubscriptions 섹션을 참조하세요.

describe-resource-permissions

다음 코드 예시는 describe-resource-permissions의 사용 방법을 보여 줍니다.

AWS CLI

리소스에 대한 권한 목록 가져오기

이 예제에서는 지정된 리소스(문서 또는 폴더)에 대한 사용 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"
                }
            ]
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeResourcePermissions](#) 섹션을 참조하세요.

describe-users

다음 코드 예시는 describe-users의 사용 방법을 보여 줍니다.

AWS CLI

지정된 사용자의 세부 정보 검색

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
      }
  ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeUsers 섹션 섹션을 참조하세요.

get-document-path

다음 코드 예시는 get-document-path의 사용 방법을 보여 줍니다.

AWS CLI

문서의 경로 정보 검색

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"
          }
      ]
  }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetDocumentPath 섹션을 참조하세요.

get-document-version

다음 코드 예시는 `get-document-version`의 사용 방법을 보여 줍니다.

AWS CLI

지정된 문서의 버전 메타데이터 검색

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"
      }
  }
```

```
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetDocumentVersion 섹션을 참조하세요.

get-document

다음 코드 예시는 get-document의 사용 방법을 보여 줍니다.

AWS CLI

문서 세부 정보 검색

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"
```

```
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetDocument 섹션을 참조하세요.

get-folder-path

다음 코드 예시는 get-folder-path의 사용 방법을 보여 줍니다.

AWS CLI

폴더의 경로 정보 검색

이 예제에서는 지정된 폴더의 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"
          }
      ]
  }
```

```
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetFolderPath 섹션을 참조하세요.

get-folder

다음 코드 예시는 get-folder의 사용 방법을 보여 줍니다.

AWS CLI

폴더의 메타데이터 검색

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
  }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetFolder 섹션을 참조하세요.

get-resources

다음 코드 예시는 get-resources의 사용 방법을 보여 줍니다.

AWS CLI

공유 리소스 검색

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.

- API 세부 정보는 AWS CLI 명령 참조의 GetResources를 참조하세요.

initiate-document-version-upload

다음 코드 예시는 `initiate-document-version-upload`의 사용 방법을 보여 줍니다.

AWS CLI

문서 버전 업로드 시작

다음 `initiate-document-upload` 예제에서는 새 문서 객체와 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"
        }
    }
}
```

- API 세부 정보는 AWS CLI 명령 참조의 InitiateDocumentVersionUpload 섹션을 참조하세요.

`remove-all-resource-permissions`

다음 코드 예시는 `remove-all-resource-permissions`의 사용 방법을 보여 줍니다.

AWS CLI

지정된 리소스에서 모든 권한 제거

이 예제에서는 지정된 리소스에서 모든 권한을 제거합니다.

명령:

```
aws workdocs remove-all-resource-permissions --resource-
id 1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678
```

출력:

```
None
```

- API 세부 정보는 AWS CLI 명령 참조의 RemoveAllResourcePermissions 섹션을 참조하세요.

remove-resource-permission

다음 코드 예시는 `remove-resource-permission`의 사용 방법을 보여 줍니다.

AWS CLI

리소스에서 권한 제거

이 예제에서는 지정된 주체에 대한 리소스에서 권한을 제거합니다.

명령:

```
aws workdocs remove-resource-permission --resource-
id 1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678 --principal-
id anonymous
```

출력:

```
None
```

- API 세부 정보는 AWS CLI 명령 참조의 RemoveResourcePermission 섹션을 참조하세요.

update-document-version

다음 코드 예시는 `update-document-version`의 사용 방법을 보여 줍니다.

AWS CLI

문서 버전 상태를 활성으로 변경

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
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateDocumentVersion 섹션을 참조하세요.

update-document

다음 코드 예시는 update-document의 사용 방법을 보여 줍니다.

AWS CLI

문서 업데이트

이 예제에서는 문서의 이름과 상위 폴더를 업데이트합니다.

명령:

```
aws workdocs update-document --document-
id 15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3 --
name updatedDoc --parent-folder-
id 50893c0af679524d1a0e0651130ed6d073e1a05f95bd12c42dcde5d35634ed08
```

출력:

```
None
```

- API 세부 정보는 AWS CLI 명령 참조의 [UpdateDocument](https://docs.aws.amazon.com/goto/aws-cli/workdocs-2016-05-01/UpdateDocument) 섹션을 참조하세요.

update-folder

다음 코드 예시는 update-folder의 사용 방법을 보여 줍니다.

AWS CLI

폴더 업데이트

이 예제에서는 폴더의 이름과 상위 폴더를 업데이트합니다.

명령:

```
aws workdocs update-folder --folder-
id 50893c0af679524d1a0e0651130ed6d073e1a05f95bd12c42dcde5d35634ed08 --
name exampleFolder1 --parent-folder-
id 1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678
```

출력:

```
None
```

- API 세부 정보는 AWS CLI 명령 참조의 [UpdateFolder](https://docs.aws.amazon.com/goto/aws-cli/workdocs-2016-05-01/UpdateFolder) 섹션을 참조하세요.

update-user

다음 코드 예시는 update-user의 사용 방법을 보여 줍니다.

AWS CLI

사용자 업데이트

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"
          }
      }
  }
}
```


- API 세부 정보는 AWS CLI 명령 참조의 UpdateUser 섹션을 참조하세요.

AWS CLI를 사용한 Amazon WorkMail 예시

다음 코드 예시에서는 Amazon WorkMail에서 AWS Command Line Interface를 사용하여 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`associate-delegate-to-resource`

다음 코드 예시에서는 `associate-delegate-to-resource` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

리소스에 대리인 추가

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.

- API 세부 정보는 AWS CLI 명령 참조의 AssociateDelegateToResource 섹션을 참조하세요.

`associate-member-to-group`

다음 코드 예시에서는 `associate-member-to-group` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

그룹에 멤버 추가

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.

- API 세부 정보는 AWS CLI 명령 참조의 AssociateMemberToGroup 섹션을 참조하세요.

`create-alias`

다음 코드 예시에서는 `create-alias` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

별칭 생성

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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateAlias 섹션을 참조하세요.

create-group

다음 코드 예시에서는 create-group 코드를 사용하는 방법을 보여줍니다.

AWS CLI

새 그룹 생성

다음 create-group 명령은 지정된 조직에 대한 새 그룹을 만듭니다.

```
aws workmail create-group \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --name exampleGroup1
```

출력:

```
{
    "GroupId": "S-1-1-11-1122222222-2222233333-3333334444-4444"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateGroup을 참조하세요.

create-resource

다음 코드 예시에서는 create-resource 코드를 사용하는 방법을 보여줍니다.

AWS CLI

새 리소스 생성

다음 `create-resource` 명령은 지정된 조직에 대한 새 리소스(회의실)를 만듭니다.

```
aws workmail create-resource \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --name exampleRoom1 \
    --type ROOM
```

출력:

```
{
    "ResourceId": "r-7afe0efbade843a58cdc10251fce992c"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateResource를 참조하세요.

`create-user`

다음 코드 예시에서는 `create-user` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

새로운 사용자 생성

다음 `create-user` 명령은 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [CreateUser](#)를 참조하세요.

delete-access-control-rule

다음 코드 예시에서는 `delete-access-control-rule` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

액세스 제어 규칙 삭제

다음 `delete-access-control-rule` 예시에서는 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 [Working with Access Control Rules](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteAccessControlRule](#) 섹션을 참조하세요.

delete-alias

다음 코드 예시에서는 `delete-alias` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

별칭 삭제

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.

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteAlias](#)를 참조하세요.

`delete-group`

다음 코드 예시에서는 `delete-group` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

기존 그룹 삭제

다음 `delete-group` 명령은 Amazon WorkMail 에서 기존 그룹을 삭제합니다.

```
aws workmail delete-group \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --group-id S-1-1-11-1122222222-2222233333-3333334444-4444
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteGroup을 참조하세요.

`delete-mailbox-permissions`

다음 코드 예시에서는 `delete-mailbox-permissions` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사서함 권한을 삭제하는 방법

다음 `delete-mailbox-permissions` 명령은 이전에 사용자 또는 그룹에 부여된 사서함 권한을 삭제합니다. 엔터티는 사서함을 소유한 사용자를 나타내며, 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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteMailboxPermissions 섹션을 참조하세요.

`delete-resource`

다음 코드 예시에서는 `delete-resource` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

기존 리소스를 삭제하는 방법

다음 delete-resource 명령은 Amazon WorkMail에서 기존 리소스를 삭제합니다.

```
aws workmail delete-resource \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --resource-id r-7afe0efbade843a58cdc10251fce992c
```

이 명령은 출력을 생성하지 않습니다.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteResource 섹션을 참조하세요.

delete-user

다음 코드 예시에서는 delete-user 코드를 사용하는 방법을 보여줍니다.

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.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteUser를 참조하세요.

deregister-from-work-mail

다음 코드 예시에서는 deregister-from-work-mail 코드를 사용하는 방법을 보여줍니다.

AWS CLI

기존 엔터티를 비활성화하는 방법

다음 deregister-from-work-mail 명령은 기존 엔터티(사용자, 그룹 또는 리소스)가 Amazon WorkMail을 사용할 수 없도록 설정합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 [DeregisterFromWorkMail](#) 섹션을 참조하세요.

`describe-group`

다음 코드 예시에서는 `describe-group` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

그룹에 대한 정보를 검색하는 방법

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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeGroup](#) 섹션을 참조하세요.

`describe-organization`

다음 코드 예시에서는 `describe-organization` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

조직의 정보 검색

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 Working with Organizations를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeOrganization 섹션을 참조하세요.

describe-resource

다음 코드 예시에서는 describe-resource 코드를 사용하는 방법을 보여줍니다.

AWS CLI

리소스에 대한 정보를 검색하는 방법

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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeResource 섹션을 참조하세요.

describe-user

다음 코드 예시에서는 describe-user 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용자 정보를 검색하는 방법

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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeUser 섹션을 참조하세요.

disassociate-delegate-from-resource

다음 코드 예시에서는 disassociate-delegate-from-resource 코드를 사용하는 방법을 보여줍니다.

AWS CLI

리소스에서 태그 제거

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.

- API 세부 정보는 AWS CLI 명령 참조의 [DisassociateDelegateFromResource](https://) 섹션을 참조하세요.

`disassociate-member-from-group`

다음 코드 예시에서는 `disassociate-member-from-group` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

그룹에서 멤버 제거

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.

- API 세부 정보는 AWS CLI 명령 참조의 [DisassociateMemberFromGroup](https://) 섹션을 참조하세요.

`get-access-control-effect`

다음 코드 예시에서는 `get-access-control-effect` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

액세스 제어 규칙의 효과를 받는 방법

다음 get-access-control-effect 예시에서는 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 Working with Access Control Rules를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetAccessControlEffect 섹션을 참조하세요.

get-mailbox-details

다음 코드 예시에서는 get-mailbox-details 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용자의 사서함 세부 정보를 가져오는 방법

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 Managing User Accounts를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetMailboxDetails 섹션을 참조하세요.

list-access-control-rules

다음 코드 예시에서는 `list-access-control-rules` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

액세스 제어 규칙을 나열하는 방법

다음 `list-access-control-rules` 예시에서는 지정된 Amazon WorkMail 조직의 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 Working with Access Control Rules를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListAccessControlRules 섹션을 참조하세요.

list-aliases

다음 코드 예시에서는 `list-aliases` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

멤버의 별칭을 나열하는 방법

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"
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListAliases를 참조하세요.

list-group-members

다음 코드 예시에서는 `list-group-members` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

그룹 멤버를 나열하는 방법

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
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListGroupMembers 섹션을 참조하세요.

list-groups

다음 코드 예시에서는 `list-groups` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

그룹 목록 검색

다음 `list-groups` 명령은 지정된 조직에 있는 그룹의 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListGroups를 참조하세요.

`list-mailbox-permissions`

다음 코드 예시에서는 `list-mailbox-permissions` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사서함 권한 검색

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"
            ]
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [ListMailboxPermissions](https://docs.aws.amazon.com/goto/aws-cli/workmail-2017-10-01/ListMailboxPermissions) 섹션을 참조하세요.

`list-organizations`

다음 코드 예시에서는 `list-organizations` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

조직 목록을 검색하는 방법

다음 `list-organizations` 명령은 고객 조직의 요약을 검색합니다.

```
aws workmail list-organizations
```

출력:

```
{
    "OrganizationSummaries": [
        {
            "OrganizationId": "m-d281d0a2fd824be5b6cd3d3ce909fd27",
            "Alias": "exampleAlias",
            "State": "Active"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [ListOrganizations](https://docs.aws.amazon.com/goto/aws-cli/workmail-2017-10-01/ListOrganizations) 섹션을 참조하세요.

`list-resource-delegates`

다음 코드 예시에서는 `list-resource-delegates` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

리소스의 대리인을 나열하는 방법

다음 `list-resource-delegates` 명령은 지정된 리소스와 연결된 대리인을 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [ListResourceDelegates](#) 섹션을 참조하세요.

list-resources

다음 코드 예시에서는 `list-resources` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

리소스 목록을 검색하는 방법

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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [ListResources](#) 섹션을 참조하세요.

`list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

리소스에 대한 태그 나열

다음 `list-tags-for-resource` 예시에서는 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 Tagging an Organization을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ListTagsForResource를 참조하세요.

`list-users`

다음 코드 예시에서는 `list-users` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용자 목록 검색

다음 `list-users` 명령은 지정된 조직에 있는 사용자의 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"
        }
    ]
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListUsers를 참조하세요.

put-access-control-rule

다음 코드 예시에서는 `put-access-control-rule` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

새 액세스 제어 규칙 설정

다음 `put-access-control-rule` 예시에서는 지정된 사용자가 지정된 Amazon WorkMail 조직에 액세스하는 것을 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 Working with Access Control Rules를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutAccessControlRule 섹션을 참조하세요.

put-mailbox-permissions

다음 코드 예시에서는 put-mailbox-permissions 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사서함 권한 설정

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.

- API 세부 정보는 AWS CLI 명령 참조의 PutMailboxPermissions 섹션을 참조하세요.

register-to-work-mail

다음 코드 예시에서는 register-to-work-mail 코드를 사용하는 방법을 보여줍니다.

AWS CLI

기존 또는 비활성화된 엔터티를 등록하는 방법

다음 register-to-work-mail 명령은 지정된 기존 엔터티(사용자, 그룹 또는 리소스)가 Amazon WorkMail을 사용할 수 있도록 합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 RegisterToWorkMail 섹션을 참조하세요.

reset-password

다음 코드 예시에서는 `reset-password` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용자 암호를 재설정하려면

다음 `reset-password` 명령은 지정된 사용자의 암호를 초기화합니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 ResetPassword 섹션을 참조하세요.

tag-resource

다음 코드 예시에서는 `tag-resource` 코드를 사용하는 방법을 보여줍니다.

AWS CLI

리소스에 태그 적용

다음 `tag-resource` 예시는 키 'priority'와 값 '1'을 가진 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 Tagging an Organization을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TagResource를 참조하세요.

untag-resource

다음 코드 예시에서는 untag-resource 코드를 사용하는 방법을 보여줍니다.

AWS CLI

리소스에서 태그 제거

다음 untag-resource 예시에서는 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 Tagging an Organization을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UntagResource를 참조하세요.

update-mailbox-quota

다음 코드 예시에서는 update-mailbox-quota 코드를 사용하는 방법을 보여줍니다.

AWS CLI

사용자의 사서함 할당량을 업데이트하는 방법

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 Managing User Accounts를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateMailboxQuota 섹션을 참조하세요.

update-primary-email-address

다음 코드 예시에서는 update-primary-email-address 코드를 사용하는 방법을 보여줍니다.

AWS CLI

기본 이메일 주소를 업데이트하는 방법

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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdatePrimaryEmailAddress 섹션을 참조하세요.

update-resource

다음 코드 예시에서는 update-resource 코드를 사용하는 방법을 보여줍니다.

AWS CLI

리소스 업데이트

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.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateResource 섹션을 참조하세요.

AWS CLI를 사용한 Amazon WorkMail Message Flow 예제

다음 코드 예제에서는 Amazon WorkMail Message Flow에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여 줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

get-raw-message-content

다음 코드 예시는 `get-raw-message-content`의 사용 방법을 보여 줍니다.

AWS CLI

이메일 메시지의 원시 콘텐츠 가져오기

다음 `get-raw-message-content` 예제에서는 전송 중인 이메일 메시지의 원시 콘텐츠를 가져와 `test`라는 이름의 텍스트 파일로 보냅니다.

```
aws workmailmessageflow get-raw-message-content \
    --message-id a1b2cd34-ef5g-6h7j-kl8m-npq9012345rs \
    test
```

명령 실행 후 파일 `test` 콘텐츠입니다.

```
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 관리 안내서의 AWS Lambda를 통해 메시지 콘텐츠 검색을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetRawMessageContent 섹션을 참조하세요.

AWS CLI를 사용한 WorkSpaces 예제

다음 코드 예제에서는 WorkSpaces에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여 줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

create-tags

다음 코드 예시는 `create-tags`의 사용 방법을 보여 줍니다.

AWS CLI

WorkSpace에 태그를 추가하는 방법

다음 `create-tags` 예제에서는 지정된 태그를 지정된 WorkSpace.에 추가합니다.

```
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 태그를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateTags 섹션을 참조하세요.

create-workspaces

다음 코드 예시는 `create-workspaces`의 사용 방법을 보여 줍니다.

AWS CLI

예제 1: AlwaysOn WorkSpace를 생성하는 방법

다음 `create-workspaces` 예제에서는 지정된 디렉터리와 번들을 사용하여 지정된 사용자에 대한 AlwaysOn WorkSpace를 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: AutoStop WorkSpace를 생성하는 방법

다음 create-workspaces 예제에서는 지정된 디렉터리와 번들을 사용하여 지정된 사용자에 대한 AutoStop WorkSpace를 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 WorkSpace를 생성하는 방법

다음 `create-workspaces` 예제에서는 사용자 이름을 `[UNDEFINED]`로 설정하고 워크스페이스 이름, 디렉터리 ID 및 번들 ID를 지정하여 사용자 분리된 워크스페이스를 만듭니다.

```
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.

- API 세부 정보는 AWS CLI 명령 참조의 CreateWorkspaces 섹션을 참조하세요.

delete-tags

다음 코드 예시는 `delete-tags`의 사용 방법을 보여 줍니다.

AWS CLI

WorkSpace에서 태그를 삭제하는 방법

다음 `delete-tags` 예제에서는 지정된 WorkSpace에서 지정된 태그를 삭제합니다.

```
aws workspaces delete-tags \
    --resource-id ws-dk1xzr417 \
    --tag-keys Department
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon WorkSpaces 관리 안내서의 WorkSpaces 리소스 태그를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteTags 섹션을 참조하세요.

deregister-workspace-directory

다음 코드 예시는 deregister-workspace-directory의 사용 방법을 보여 줍니다.

AWS CLI

디렉터리를 등록 해제하는 방법

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 WorkSpaces로 디렉터리 등록을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeregisterWorkspaceDirectory 섹션을 참조하세요.

describe-tags

다음 코드 예시는 describe-tags의 사용 방법을 보여 줍니다.

AWS CLI

WorkSpace에 대한 태그 설명

다음 describe-tags 예제에서는 지정된 WorkSpace에 대한 태그를 설명합니다.

```
aws workspaces describe-tags \
    --resource-id ws-dk1xzr417
```

출력:

```
{
    "TagList": [
        {
```

```
                "Key": "Department",
                "Value": "Finance"
            }
        ]
    }
```

자세한 내용은 Amazon WorkSpaces 관리 안내서의 WorkSpaces 리소스 태그를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeTags 섹션을 참조하세요.

describe-workspace-bundles

다음 코드 예시는 describe-workspace-bundles의 사용 방법을 보여 줍니다.

AWS CLI

Amazon에서 제공하는 번들 나열

다음 describe-workspace-bundles 예제에서는 Amazon에서 제공하는 번들의 이름과 ID를 테이블 형식으로 나열하고 이름별로 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.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeWorkspaceBundles 섹션을 참조하세요.

describe-workspace-directories

다음 코드 예시는 `describe-workspace-directories`의 사용 방법을 보여 줍니다.

AWS CLI

등록된 디렉터리 설명

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": {
                "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 WorkSpaces의 디렉터리 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeMaintenanceWindows 섹션을 참조하세요.

describe-workspaces-connection-status

다음 코드 예시는 describe-workspaces-connection-status의 사용 방법을 보여 줍니다.

AWS CLI

WorkSpace의 연결 상태 설명

다음 `describe-workspaces-connection-status` 예제에서는 지정된 WorkSpace의 연결 상태를 설명합니다.

```
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 WorkSpaces로 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeWorkspacesConnectionStatus 섹션을 참조하세요.

`describe-workspaces`

다음 코드 예시는 `describe-workspaces`의 사용 방법을 보여 줍니다.

AWS CLI

WorkSpace 설명

다음 `describe-workspaces` 예제에서는 지정된 WorkSpace를 설명합니다.

```
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 WorkSpaces로 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DescribeWorkspaces 섹션을 참조하세요.

migrate-workspace

다음 코드 예시는 migrate-workspace의 사용 방법을 보여 줍니다.

AWS CLI

WorkSpace를 마이그레이션하는 방법

다음 migrate-workspace 예제에서는 지정된 WorkSpace를 지정된 번들로 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 WorkSpace 마이그레이션을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 MigrateWorkspace 섹션을 참조하세요.

modify-workspace-creation-properties

다음 코드 예시는 modify-workspace-creation-properties의 사용 방법을 보여 줍니다.

AWS CLI

디렉터리의 WorkSpace 생성 속성 수정

다음 modify-workspace-creation-properties 예제에서는 지정된 디렉터리에 대한 EnableInternetAccess 속성을 활성화합니다. 이렇게 하면 디렉터리에 대해 생성된 워크스페이스에 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 WorkSpaces의 디렉터리 업데이트를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyWorkspaceCreationProperties 섹션을 참조하세요.

modify-workspace-properties

다음 코드 예시는 modify-workspace-properties의 사용 방법을 보여 줍니다.

AWS CLI

WorkSpaces 실행 모드를 수정하는 방법

다음 modify-workspace-properties 예제에서는 지정된 WorkSpace의 실행 모드를 AUTO_STOP로 설정합니다.

```
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 WorkSpaces 수정을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyWorkspaceProperties 섹션을 참조하세요.

modify-workspace-state

다음 코드 예시는 modify-workspace-state의 사용 방법을 보여 줍니다.

AWS CLI

WorkSpace의 상태 수정

다음 modify-workspace-state 예제에서는 지정된 WorkSpace의 상태를 ADMIN_MAINTENANCE로 설정합니다.

```
aws workspaces modify-workspace-state \
    --workspace-id ws-dk1xzr417 \
    --workspace-state ADMIN_MAINTENANCE
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon WorkSpaces 관리 안내서의 WorkSpaces 유지 관리를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 ModifyWorkspaceState 섹션을 참조하세요.

reboot-workspaces

다음 코드 예시는 reboot-workspaces의 사용 방법을 보여 줍니다.

AWS CLI

WorkSpace를 재부팅하는 방법

다음 reboot-workspaces 예제에서는 지정된 WorkSpace를 재부팅합니다.

```
aws workspaces reboot-workspaces \
    --reboot-workspace-requests ws-dk1xzr417
```

출력:

```
{
    "FailedRequests": []
}
```

자세한 내용은 Amazon WorkSpaces 관리 안내서의 WorkSpaces 리부팅을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RebootWorkspaces 섹션을 참조하세요.

`rebuild-workspaces`

다음 코드 예시는 `rebuild-workspaces`의 사용 방법을 보여 줍니다.

AWS CLI

WorkSpace를 재구축하는 방법

다음 `rebuild-workspaces` 예제에서는 지정된 WorkSpace 를 재구축합니다.

```
aws workspaces rebuild-workspaces \
    --rebuild-workspace-requests ws-dk1xzr417
```

출력:

```
{
    "FailedRequests": []
}
```

자세한 내용은 Amazon WorkSpaces 관리 안내서의 WorkSpaces 재구축을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RebuildWorkspaces 섹션을 참조하세요.

`register-workspace-directory`

다음 코드 예시는 `register-workspace-directory`의 사용 방법을 보여 줍니다.

AWS CLI

디렉터리를 등록하는 방법

다음 `register-workspace-directory` 예제에서는 Amazon WorkSpaces에 사용할 지정된 디렉터리를 등록합니다.

```
aws workspaces register-workspace-directory \
    --directory-id d-926722edaf
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon WorkSpaces 관리 안내서의 WorkSpaces Personal로 기존 AWS Directory Service 디렉터리 등록을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RegisterDefaultPatchBaseline 섹션을 참조하세요.

`restore-workspace`

다음 코드 예시는 `restore-workspace`의 사용 방법을 보여 줍니다.

AWS CLI

WorkSpace를 복원하는 방법

다음 `restore-workspace` 예제에서는 지정된 WorkSpace를 복원합니다.

```
aws workspaces restore-workspace \
    --workspace-id ws-dk1xzr417
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon WorkSpaces 관리 안내서의 WorkSpaces 복원을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 RestoreWorkspace 섹션을 참조하세요.

`start-workspaces`

다음 코드 예시는 `start-workspaces`의 사용 방법을 보여 줍니다.

AWS CLI

AutoStop WorkSpaces를 시작하는 방법

다음 `start-workspaces` 예제에서는 지정된 WorkSpace를 시작합니다. WorkSpace의 실행 모드가 `AutoStop`이어야 합니다.

```
aws workspaces start-workspaces \
```

```
    --start-workspace-requests WorkspaceId=ws-dk1xzr417
```

출력:

```
{
    "FailedRequests": []
}
```

자세한 내용은 Amazon WorkSpaces 관리 안내서의 AutoStop WorkSpace 중지 및 시작을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StartWorkspaces 섹션을 참조하세요.

`stop-workspaces`

다음 코드 예시는 `stop-workspaces`의 사용 방법을 보여 줍니다.

AWS CLI

AutoStop WorkSpaces를 중지하는 방법

다음 `stop-workspaces` 예제에서는 지정된 WorkSpace를 중지합니다. WorkSpace의 실행 모드가 `AutoStop`이어야 합니다.

```
aws workspaces stop-workspaces \
    --stop-workspace-requests WorkspaceId=ws-dk1xzr417
```

출력:

```
{
    "FailedRequests": []
}
```

자세한 내용은 Amazon WorkSpaces 관리 안내서의 AutoStop WorkSpace 중지 및 시작을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 StopWorkspaces 섹션을 참조하세요.

`terminate-workspaces`

다음 코드 예시는 `terminate-workspaces`의 사용 방법을 보여 줍니다.

AWS CLI

WorkSpace 종료

다음 `terminate-workspaces` 예제에서는 지정된 WorkSpace를 종료합니다.

```
aws workspaces terminate-workspaces \
    --terminate-workspace-requests ws-dk1xzr417
```

출력:

```
{
    "FailedRequests": []
}
```

자세한 내용은 Amazon WorkSpaces 관리 안내서의 WorkSpace 삭제를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 TerminateWorkspaces 섹션을 참조하세요.

AWS CLI를 사용한 X-Ray 예제

다음 코드 예제에서는 X-Ray에서 AWS Command Line Interface를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여 줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`batch-traces-get`

다음 코드 예시는 `batch-traces-get`의 사용 방법을 보여 줍니다.

AWS CLI

추적 목록 가져오기

다음 `batch-get-traces` 예제에서는 ID로 지정된 추적 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 AWS CLI에서 AWS X-Ray API 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 BatchTracesGet 섹션을 참조하세요.

create-group

다음 코드 예시는 `create-group`의 사용 방법을 보여 줍니다.

AWS CLI

그룹 생성

다음 예제에서는 `create-group`이라는 그룹 리소스를 생성합니다. 그룹은 장애 또는 오류를 일으키는 특정 서비스와 관련된 세그먼트로 그룹의 기준을 정의하는 필터 식을 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 안내서에서 AWS X-Ray API를 통해 샘플링, 그룹 및 암호화 설정 구성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 CreateGroup을 참조하세요.

create-sampling-rule

다음 코드 예시는 create-sampling-rule의 사용 방법을 보여 줍니다.

AWS CLI

샘플링 규칙 생성

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 안내서에서 AWS X-Ray API를 통해 샘플링, 그룹 및 암호화 설정 구성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의CreateSamplingRule 섹션을 참조하세요.

delete-group

다음 코드 예시는 delete-group의 사용 방법을 보여 줍니다.

AWS CLI

그룹 삭제

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 안내서에서 AWS X-Ray API를 통해 샘플링, 그룹 및 암호화 설정 구성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteGroup을 참조하세요.

`delete-sampling-rule`

다음 코드 예시는 `delete-sampling-rule`의 사용 방법을 보여 줍니다.

AWS CLI

샘플링 규칙 삭제

다음 `delete-sampling-rule` 예제에서는 지정된 샘플링 규칙을 삭제합니다. 그룹 이름 또는 그룹 ARN을 사용하여 그룹을 지정할 수 있습니다.

```
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 안내서에서 AWS X-Ray API를 통해 샘플링, 그룹 및 암호화 설정 구성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 DeleteSamplingRule 섹션을 참조하세요.

`get-encryption-config`

다음 코드 예시는 `get-encryption-config`의 사용 방법을 보여 줍니다.

AWS CLI

암호화 구성 검색

다음 `get-encryption-config` 예제에서는 AWS X-Ray 데이터의 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 안내서에서 AWS X-Ray API를 통해 샘플링, 그룹 및 암호화 설정 구성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetEncryptionConfig 섹션을 참조하세요.

`get-group`

다음 코드 예시는 `get-group`의 사용 방법을 보여 줍니다.

AWS CLI

그룹 검색

다음 `get-group` 예제에서는 지정된 그룹 리소스에 대한 세부 정보를 표시합니다. 세부 정보에는 그룹 이름, 그룹 ARN 및 해당 그룹에 대한 기준을 정의하는 필터 표현식이 포함됩니다. 그룹은 ARN으로도 검색할 수 있습니다.

```
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 안내서에서 AWS X-Ray API를 통해 샘플링, 그룹 및 암호화 설정 구성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetGroup을 참조하세요.

get-groups

다음 코드 예시는 `get-groups`의 사용 방법을 보여 줍니다.

AWS CLI

모든 그룹 검색

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 안내서에서 AWS X-Ray API를 통해 샘플링, 그룹 및 암호화 설정 구성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetGroup 섹션을 참조하세요.

`get-sampling-rules`

다음 코드 예시는 `get-sampling-rules`의 사용 방법을 보여 줍니다.

AWS CLI

모든 샘플링 규칙 검색

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 AWS X-Ray 개발자 안내서에서 X-Ray API로 샘플링 규칙 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetBucketWebsite 섹션을 참조하세요.

get-sampling-targets

다음 코드 예시는 get-sampling-targets의 사용 방법을 보여 줍니다.

AWS CLI

샘플링 할당량 요청

다음 get-sampling-targets 예제에서는 서비스가 요청을 샘플링하기 위해 사용하는 규칙에 대한 샘플링 할당량을 요청합니다. AWS X-Ray의 응답에는 저장소에서 빌리는 대신 사용할 수 있는 할당량이 포함되어 있습니다.

```
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 AWS X-Ray 개발자 안내서에서 X-Ray API로 샘플링 규칙 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetSamplingTargets 섹션을 참조하세요.

get-service-graph

다음 코드 예시는 get-service-graph의 사용 방법을 보여 줍니다.

AWS CLI

서비스 그래프 가져오기

다음 예제에서는 수신 요청을 처리하는 서비스와 그 결과로 호출하는 다운스트림 서비스를 설명하는 지정된 기간 내의 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 AWS CLI에서 AWS X-Ray API 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetServiceGraph 섹션을 참조하세요.

get-trace-summaries

다음 코드 예시는 `get-trace-summaries`의 사용 방법을 보여 줍니다.

AWS CLI

추적 요약 가져오기

다음 `get-trace-summaries` 예제에서는 지정된 기간 내에 사용 가능한 추적에 대한 ID와 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 AWS CLI에서 AWS X-Ray API 사용을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 GetTraceSummaries 섹션을 참조하세요.

put-encryption-config

다음 코드 예시는 put-encryption-config의 사용 방법을 보여 줍니다.

AWS CLI

암호화 구성 업데이트

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 안내서에서 AWS X-Ray API를 통해 샘플링, 그룹 및 암호화 설정 구성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutEncryptionConfig 섹션을 참조하세요.

put-trace-segments

다음 코드 예시는 put-trace-segments의 사용 방법을 보여 줍니다.

AWS CLI

세그먼트 업로드

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 AWS X-Ray로 트레이스 데이터 전송을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 PutTraceSegments 섹션을 참조하세요.

update-group

다음 코드 예시는 update-group의 사용 방법을 보여 줍니다.

AWS CLI

그룹 업데이트

다음 update-group 예제에서는 AdminGroup이라는 그룹으로 트레이스를 수락할 기준을 업데이트합니다. 그룹 이름 또는 그룹 ARN을 사용하여 원하는 그룹을 지정할 수 있습니다.

```
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 안내서에서 AWS X-Ray API를 통해 샘플링, 그룹 및 암호화 설정 구성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateGroup을 참조하세요.

update-sampling-rule

다음 코드 예시는 update-sampling-rule의 사용 방법을 보여 줍니다.

AWS CLI

샘플링 규칙 업데이트

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 안내서에서 AWS X-Ray API를 통해 샘플링, 그룹 및 암호화 설정 구성을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 UpdateSamplingRuler 섹션을 참조하세요.

Bash 스크립트와 함께 AWS CLI 사용 예제

이 주제의 코드 예제에서는 AWS에서 Bash 스크립트와 함께 AWS Command Line Interface를 사용하는 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.

시나리오는 동일한 서비스 내에서 또는 다른 AWS 서비스와 결합된 상태에서 여러 함수를 호출하여 특정 태스크를 수행하는 방법을 보여주는 코드 예제입니다.

일부 서비스에는 서비스와 관련된 라이브러리 또는 함수를 활용하는 방법을 보여주는 추가 예제 범주가 포함되어 있습니다.

서비스

- Bash 스크립트와 함께 AWS CLI를 사용하는 DynamoDB 예제
- Bash 스크립트와 함께 AWS CLI를 사용하는 Amazon EC2 예제
- Bash 스크립트와 함께 AWS CLI를 사용하는 HealthImaging 예제
- Bash 스크립트와 함께 AWS CLI를 사용하는 IAM 예제
- Bash 스크립트와 함께 AWS CLI를 사용하는 AWS KMS 예제
- Bash 스크립트와 함께 AWS CLI를 사용하는 Amazon S3 예제
- Bash 스크립트와 함께 AWS CLI를 사용하는 AWS STS 예제

Bash 스크립트와 함께 AWS CLI를 사용하는 DynamoDB 예제

다음 코드 예제에서는 DynamoDB에서 Bash 스크립트와 함께 AWS Command Line Interface를 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.

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

- 기본 사항
- 작업
- 시나리오

기본 사항

기본 사항 알아보기

다음 코드 예제는 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.

Bash 스크립트와 함께 AWS CLI사용

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. [AWS 코드 예제 리포지토리](https://github.com/awsdocs/aws-doc-sdk-examples)에서 전체 예제를 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

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" 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.
#
#  Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function dynamodb_create_table() {
  local table_name attribute_definitions key_schema 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 with on-demand billing."
    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 ""
  }

  # Retrieve the calling parameters.
  while getopts "n:a:k:h" option; do
    case "${option}" in
      n) table_name="${OPTARG}" ;;
      a) attribute_definitions="${OPTARG}" ;;
      k) key_schema="${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

  iecho "Parameters:\n"
  iecho "    table_name:   $table_name"
  iecho "    attribute_definitions:   $attribute_definitions"
  iecho "    key_schema:   $key_schema"
  iecho ""

  response=$(aws dynamodb create-table \
    --table-name "$table_name" \
    --attribute-definitions file://"$attribute_definitions" \
    --billing-mode PAY_PER_REQUEST \
    --key-schema file://"$key_schema" )

  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 세부 정보는 AWS CLI 명령 참조의 다음 토픽을 참조하세요.
  - BatchWriteItem
  - CreateTable
  - DeleteItem
  - DeleteTable
  - DescribeTable
  - GetItem
  - PutItem
  - Query
  - Scan
  - UpdateItem

작업

`BatchGetItem`

다음 코드 예제에서는 `BatchGetItem` 코드를 사용하는 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예제 리포지토리에서 전체 예제를 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 BatchGetItem을 참조하세요.

BatchWriteItem

다음 코드 예제에서는 BatchWriteItem 코드를 사용하는 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예제 리포지토리에서 전체 예제를 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [BatchWriteItem](https://docs.aws.amazon.com/goto/aws-cli/dynamodb-2012-08-10/BatchWriteItem)을 참조하세요.

`CreateTable`

다음 코드 예제에서는 `CreateTable` 코드를 사용하는 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. [AWS 코드 예제 리포지토리](https://github.com/awsdocs/aws-doc-sdk-examples)에서 전체 예제를 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.
#
#  Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function dynamodb_create_table() {
  local table_name attribute_definitions key_schema 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 with on-demand billing."
    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 ""
```

```
  }

  # Retrieve the calling parameters.
  while getopts "n:a:k:h" option; do
    case "${option}" in
      n) table_name="${OPTARG}" ;;
      a) attribute_definitions="${OPTARG}" ;;
      k) key_schema="${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

  iecho "Parameters:\n"
  iecho "    table_name:   $table_name"
  iecho "    attribute_definitions:   $attribute_definitions"
  iecho "    key_schema:   $key_schema"
```

```
  iecho ""

  response=$(aws dynamodb create-table \
    --table-name "$table_name" \
    --attribute-definitions file://"$attribute_definitions" \
    --billing-mode PAY_PER_REQUEST \
    --key-schema file://"$key_schema" )

  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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [CreateTable](#)을 참조하세요.

DeleteItem

다음 코드 예제에서는 `DeleteItem` 코드를 사용하는 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

Note

GitHub에 더 많은 내용이 있습니다. AWS 코드 예제 리포지토리에서 전체 예제를 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteItem을 참조하세요.

DeleteTable

다음 코드 예제에서는 `DeleteTable` 코드를 사용하는 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예제 리포지토리에서 전체 예제를 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteTable을 참조하세요.

DescribeTable

다음 코드 예제에서는 DescribeTable 코드를 사용하는 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeTable을 참조하세요.

GetItem

다음 코드 예제에서는 GetItem 코드를 사용하는 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

> Note
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예제 리포지토리에서 전체 예제를 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetItem을 참조하세요.

ListTables

다음 코드 예시는 ListTables의 사용 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예제 리포지토리에서 전체 예제를 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [ListTables](https://docs.aws.amazon.com/goto/aws-cli/dynamodb-2012-08-10/ListTables)를 참조하세요.

PutItem

다음 코드 예제에서는 PutItem 코드를 사용하는 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예제 리포지토리에서 전체 예제를 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 PutItem을 참조하세요.

Query

다음 코드 예시는 `Query`의 사용 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예제 리포지토리에서 전체 예제를 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 Query를 참조하세요.

Scan

다음 코드 예제에서는 Scan 코드를 사용하는 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예제 리포지토리에서 전체 예제를 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 Scan을 참조하세요.

UpdateItem

다음 코드 예제에서는 UpdateItem 코드를 사용하는 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateItem을 참조하세요.

시나리오

고급 글로벌 보조 인덱스 시나리오

다음 코드 예제는 고급 글로벌 보조 인덱스 구성으로 작업하는 방법을 보여줍니다.

- 여러 GSI가 있는 테이블을 생성합니다.
- 온디맨드 용량과 GSI를 사용하여 테이블을 생성합니다.
- 여러 GSI가 있는 테이블에 항목을 넣습니다.
- 다양한 조건을 사용하여 여러 GSI를 쿼리합니다.

Bash 스크립트와 함께 AWS CLI사용

여러 GSI가 있는 테이블을 생성합니다.

```
# Create a table with multiple GSIs
aws dynamodb create-table \
    --table-name MusicLibrary \
    --attribute-definitions \
        AttributeName=Artist,AttributeType=S \
        AttributeName=SongTitle,AttributeType=S \
        AttributeName=AlbumTitle,AttributeType=S \
        AttributeName=Genre,AttributeType=S \
        AttributeName=Year,AttributeType=N \
    --key-schema \
        AttributeName=Artist,KeyType=HASH \
        AttributeName=SongTitle,KeyType=RANGE \
    --billing-mode PAY_PER_REQUEST \
    --global-secondary-indexes \
        "[
            {
                \"IndexName\": \"AlbumIndex\",
                \"KeySchema\": [{\"AttributeName\":\"AlbumTitle\",\"KeyType\":\"HASH
\"}],
                \"Projection\": {\"ProjectionType\":\"ALL\"}
            },
            {
                \"IndexName\": \"GenreYearIndex\",
```

```
                \"KeySchema\": [
                    {\"AttributeName\":\"Genre\",\"KeyType\":\"HASH\"},
                    {\"AttributeName\":\"Year\",\"KeyType\":\"RANGE\"}
                ],
                \"Projection\": {\"ProjectionType\":\"INCLUDE\",\"NonKeyAttributes
\":[\"Artist\",\"SongTitle\"]}
            }
        ]"
```

온디맨드 용량과 GSI를 사용하여 테이블을 생성합니다.

```
# Create a table with on-demand capacity and GSI
aws dynamodb create-table \
    --table-name MusicOnDemand \
    --attribute-definitions \
        AttributeName=Artist,AttributeType=S \
        AttributeName=SongTitle,AttributeType=S \
        AttributeName=Genre,AttributeType=S \
    --key-schema \
        AttributeName=Artist,KeyType=HASH \
        AttributeName=SongTitle,KeyType=RANGE \
    --billing-mode PAY_PER_REQUEST \
    --global-secondary-indexes \
        "[
            {
                \"IndexName\": \"GenreIndex\",
                \"KeySchema\": [{\"AttributeName\":\"Genre\",\"KeyType\":\"HASH\"}],
                \"Projection\": {\"ProjectionType\":\"ALL\"}
            }
        ]"
```

여러 GSI가 있는 테이블에 항목을 넣습니다.

```
# Add items to MusicLibrary table
aws dynamodb put-item \
    --table-name MusicLibrary \
    --item '{
        "Artist": {"S": "The Beatles"},
        "SongTitle": {"S": "Hey Jude"},
        "AlbumTitle": {"S": "Past Masters"},
        "Genre": {"S": "Rock"},
```

```
        "Year": {"N": "1968"}
    }'

aws dynamodb put-item \
    --table-name MusicLibrary \
    --item '{
        "Artist": {"S": "Miles Davis"},
        "SongTitle": {"S": "So What"},
        "AlbumTitle": {"S": "Kind of Blue"},
        "Genre": {"S": "Jazz"},
        "Year": {"N": "1959"}
    }'
```

여러 GSI가 있는 테이블에서 항목을 쿼리합니다.

```
# Query the AlbumIndex GSI
echo "Querying AlbumIndex GSI:"
aws dynamodb query \
    --table-name MusicLibrary \
    --index-name AlbumIndex \
    --key-condition-expression "AlbumTitle = :album" \
    --expression-attribute-values '{":album":{"S":"Kind of Blue"}}'

# Query the GenreYearIndex GSI with a range condition
echo "Querying GenreYearIndex GSI with range condition:"
aws dynamodb query \
    --table-name MusicLibrary \
    --index-name GenreYearIndex \
    --key-condition-expression "Genre = :genre AND #yr > :year" \
    --expression-attribute-names '{"#yr": "Year"}' \
    --expression-attribute-values '{":genre":{"S":"Rock"},":year":{"N":"1965"}}'
```

- API 세부 정보는 AWS CLI 명령 참조의 다음 토픽을 참조하세요.
  - [CreateTable]
  - [PutItem]
  - [Query]

MRSC 글로벌 테이블 만들기 및 관리

다음 코드 예제에서는 다중 리전 강력한 일관성(MRSC)을 사용하여 DynamoDB 글로벌 테이블을 만들고 관리하는 방법을 보여줍니다.

- 다중 리전 강력한 일관성이 있는 테이블을 만듭니다.
- MRSC 구성 및 복제본 상태를 확인합니다.
- 즉각적인 읽기를 통해 리전 간 강력한 일관성을 테스트합니다.
- MRSC 보장을 사용하여 조건부 쓰기를 수행합니다.
- MRSC 글로벌 테이블 리소스를 정리합니다.

Bash 스크립트와 함께 AWS CLI사용

다중 리전 강력한 일관성이 있는 테이블을 만듭니다.

```
# Step 1: Create a new table in us-east-2 (primary region for MRSC)
# Note: Table must be empty when enabling MRSC
aws dynamodb create-table \
    --table-name MusicTable \
    --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 \
    --region us-east-2

# Wait for table to become active
aws dynamodb wait table-exists --table-name MusicTable --region us-east-2

# Step 2: Add replica and witness with Multi-Region Strong Consistency
# MRSC requires exactly three replicas in supported regions
aws dynamodb update-table \
    --table-name MusicTable \
    --replica-updates '[{"Create": {"RegionName": "us-east-1"}}]' \
    --global-table-witness-updates '[{"Create": {"RegionName": "us-west-2"}}]' \
    --multi-region-consistency STRONG \
    --region us-east-2
```

MRSC 구성 및 복제본 상태를 확인합니다.

```
# Verify the global table configuration and MRSC setting
aws dynamodb describe-table \
    --table-name MusicTable \
    --region us-east-2 \
    --query 'Table.
{TableName:TableName,TableStatus:TableStatus,MultiRegionConsistency:MultiRegionConsistency,R
{Region:RegionName,Status:ReplicaStatus}}'
```

리전 간 즉각적인 읽기를 통해 강력한 일관성을 테스트합니다.

```
# Write an item to the primary region
aws dynamodb put-item \
    --table-name MusicTable \
    --item '{"Artist": {"S":"The Beatles"},"SongTitle": {"S":"Hey Jude"},"Album":
 {"S":"The Beatles 1967-1970"},"Year": {"N":"1968"}}' \
    --region us-east-2

# Read the item from replica region to verify strong consistency (cannot read or
 write to witness)
# No wait time needed - MRSC provides immediate consistency
echo "Reading from us-east-1 (immediate consistency):"
aws dynamodb get-item \
    --table-name MusicTable \
    --key '{"Artist": {"S":"The Beatles"},"SongTitle": {"S":"Hey Jude"}}' \
    --consistent-read \
    --region us-east-1
```

MRSC 보장을 사용하여 조건부 쓰기를 수행합니다.

```
# Perform a conditional update from a different region
# This demonstrates that conditions work consistently across all regions
aws dynamodb update-item \
    --table-name MusicTable \
    --key '{"Artist": {"S":"The Beatles"},"SongTitle": {"S":"Hey Jude"}}' \
    --update-expression "SET #rating = :rating" \
    --condition-expression "attribute_exists(Artist)" \
    --expression-attribute-names '{"#rating": "Rating"}' \
    --expression-attribute-values '{":rating": {"N":"5"}}' \
```

```
    --region us-east-1
```

MRSC 글로벌 테이블 리소스를 정리합니다.

```
# Remove replica tables (must be done before deleting the primary table)
aws dynamodb update-table \
    --table-name MusicTable \
    --replica-updates '[{"Delete": {"RegionName": "us-east-1"}}]' \
    --global-table-witness-updates '[{"Delete": {"RegionName": "us-west-2"}}]' \
    --region us-east-2

# Wait for replicas to be deleted
echo "Waiting for replicas to be deleted..."
sleep 30

# Delete the primary table
aws dynamodb delete-table \
    --table-name MusicTable \
    --region us-east-2
```

- API 세부 정보는 AWS CLI 명령 참조의 다음 토픽을 참조하세요.
  - CreateTable
  - DeleteTable
  - DescribeTable
  - GetItem
  - PutItem
  - UpdateItem
  - UpdateTable

글로벌 보조 인덱스 관리

다음 코드 예제에서는 글로벌 보조 인덱스의 전체 수명 주기를 관리하는 방법을 보여줍니다.

- 글로벌 보조 인덱스가 있는 테이블을 생성합니다.
- 기존 테이블에 새 GSI를 추가합니다.
- GSI 웜 처리량을 업데이트(증가)합니다.

- GSI를 사용하여 데이터를 쿼리합니다.
- GSI를 삭제합니다.

Bash 스크립트와 함께 AWS CLI사용

글로벌 보조 인덱스가 있는 테이블을 생성합니다.

```
# Create a table with a GSI
aws dynamodb create-table \
    --table-name MusicCollection \
    --attribute-definitions \
        AttributeName=Artist,AttributeType=S \
        AttributeName=SongTitle,AttributeType=S \
        AttributeName=AlbumTitle,AttributeType=S \
    --key-schema \
        AttributeName=Artist,KeyType=HASH \
        AttributeName=SongTitle,KeyType=RANGE \
    --billing-mode PAY_PER_REQUEST \
    --global-secondary-indexes \
        "IndexName=AlbumIndex,\
        KeySchema=[{AttributeName=AlbumTitle,KeyType=HASH}],\
        Projection={ProjectionType=ALL}"
```

기존 테이블에 새 (온디맨드) GSI를 추가합니다.

```
# Add a new GSI to an existing table
aws dynamodb update-table \
    --table-name MusicCollection \
    --attribute-definitions \
        AttributeName=Genre,AttributeType=S \
    --global-secondary-index-updates \
        "[{\"Create\":{\"IndexName\":\"GenreIndex\",\
        \"KeySchema\":[{\"AttributeName\":\"Genre\",\"KeyType\":\"HASH\"}],\
        \"Projection\":{\"ProjectionType\":\"ALL\"}}}]"
```

GSI 웜 처리량을 업데이트(증가)합니다.

```
# Increase the warm throughput of a GSI (default values are 12k reads, 4k writes)
aws dynamodb update-table \
```

```
    --table-name MusicCollection \
    --global-secondary-index-updates \
        "[{\"Update\":{\"IndexName\":\"AlbumIndex\",\
        \"WarmThroughput\":{\"ReadUnitsPerSecond\":15000,\"WriteUnitsPerSecond
\":6000}}}]"
```

GSI를 사용하여 데이터를 쿼리합니다.

```
# Query the AlbumIndex GSI
aws dynamodb query \
    --table-name MusicCollection \
    --index-name AlbumIndex \
    --key-condition-expression "AlbumTitle = :album" \
    --expression-attribute-values '{":album":{"S":"Let It Be"}}'

# Query the GenreIndex GSI
aws dynamodb query \
    --table-name MusicCollection \
    --index-name GenreIndex \
    --key-condition-expression "Genre = :genre" \
    --expression-attribute-values '{":genre":{"S":"Jazz"}}'
```

GSI를 삭제합니다.

```
# Delete a GSI from a table
aws dynamodb update-table \
    --table-name MusicCollection \
    --global-secondary-index-updates \
        "[{\"Delete\":{\"IndexName\":\"GenreIndex\"}}]"
```

- API 세부 정보는 AWS CLI 명령 참조의 다음 토픽을 참조하세요.
  - CreateTable
  - DeleteTable
  - Query
  - UpdateTable

리소스 기반 정책 관리

다음 코드 예제에서는 DynamoDB 테이블에 대한 리소스 기반 정책의 전체 수명 주기를 관리하는 방법을 보여줍니다.

- 리소스 정책을 포함한 테이블을 생성합니다.
- 리소스 정책을 가져옵니다.
- 리소스 정책을 업데이트합니다.
- 리소스 정책을 삭제합니다.

Bash 스크립트와 함께 AWS CLI사용

리소스 정책을 포함한 테이블을 생성합니다.

```
# Step 1: Create a DynamoDB table
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

# Step 2: Create a resource-based policy document
cat > policy.json << 'EOF'
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:role/DynamoDBReadOnly"
      },
      "Action": [
        "dynamodb:GetItem",
        "dynamodb:BatchGetItem",
        "dynamodb:Query",
        "dynamodb:Scan"
      ],
```

```
      "Resource": "arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection"
    }
  ]
}
EOF

# Step 3: Attach the resource-based policy to the table
aws dynamodb put-resource-policy \
    --resource-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection \
    --policy file://policy.json
```

리소스 정책을 가져옵니다.

```
# Get the resource-based policy attached to a table
aws dynamodb get-resource-policy \
    --resource-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection
```

리소스 정책을 업데이트합니다.

```
# Step 1: Create an updated policy document
cat > updated-policy.json << 'EOF'
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::123456789012:role/DynamoDBReadOnly",
          "arn:aws:iam::123456789012:role/DynamoDBAnalytics"
        ]
      },
      "Action": [
        "dynamodb:GetItem",
        "dynamodb:BatchGetItem",
        "dynamodb:Query",
        "dynamodb:Scan"
      ],
      "Resource": "arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection"
    }
  ]
```

```
}
EOF

# Step 2: Update the resource-based policy on the table
aws dynamodb put-resource-policy \
    --resource-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection \
    --policy file://updated-policy.json
```

리소스 정책을 삭제합니다.

```
# Delete the resource-based policy from a table
aws dynamodb delete-resource-policy \
    --resource-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection
```

- API 세부 정보는 AWS CLI 명령 참조의 다음 토픽을 참조하세요.
  - CreateTable
  - DeleteResourcePolicy
  - GetResourcePolicy
  - PutResourcePolicy

속성 기반 액세스 제어 설정

다음 코드 예제에서는 DynamoDB에 대한 속성 기반 액세스 제어(ABAC)를 구현하는 방법을 보여줍니다.

- ABAC를 위한 IAM 정책을 생성합니다.
- 부서별 태그가 있는 테이블을 생성합니다.
- 태그를 기반으로 테이블을 나열하고 필터링합니다.

Bash 스크립트와 함께 AWS CLI사용

ABAC를 위한 IAM 정책을 생성합니다.

```
# Step 1: Create a policy document for ABAC
cat > abac-policy.json << 'EOF'
{
  "Version": "2012-10-17",
```

```
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "dynamodb:GetItem",
        "dynamodb:BatchGetItem",
        "dynamodb:Query",
        "dynamodb:Scan"
      ],
      "Resource": "arn:aws:dynamodb:*:*:table/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Department": "${aws:PrincipalTag/Department}"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "dynamodb:PutItem",
        "dynamodb:UpdateItem",
        "dynamodb:DeleteItem",
        "dynamodb:BatchWriteItem"
      ],
      "Resource": "arn:aws:dynamodb:*:*:table/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Department": "${aws:PrincipalTag/Department}",
          "aws:ResourceTag/Environment": "Development"
        }
      }
    }
  ]
}
EOF

# Step 2: Create the IAM policy
aws iam create-policy \
    --policy-name DynamoDBDepartmentBasedAccess \
    --policy-document file://abac-policy.json
```

부서별 태그가 있는 테이블을 생성합니다.

```
# Create a DynamoDB table with tags for ABAC
aws dynamodb create-table \
    --table-name FinanceData \
    --attribute-definitions \
        AttributeName=RecordID,AttributeType=S \
    --key-schema \
        AttributeName=RecordID,KeyType=HASH \
    --billing-mode PAY_PER_REQUEST \
    --tags \
        Key=Department,Value=Finance \
        Key=Environment,Value=Development

# Create another table with different tags
aws dynamodb create-table \
    --table-name MarketingData \
    --attribute-definitions \
        AttributeName=RecordID,AttributeType=S \
    --key-schema \
        AttributeName=RecordID,KeyType=HASH \
    --billing-mode PAY_PER_REQUEST \
    --tags \
        Key=Department,Value=Marketing \
        Key=Environment,Value=Production
```

태그를 기반으로 테이블을 나열하고 필터링합니다.

```
# List all DynamoDB tables
echo "Listing all tables:"
aws dynamodb list-tables

# Get ARNs for all tables
echo -e "\nGetting ARNs for all tables:"
TABLE_ARNS=$(aws dynamodb list-tables --query "TableNames[*]" --output text | xargs
 -I {} aws dynamodb describe-table --table-name {} --query "Table.TableArn" --output
 text)

# For each table ARN, list its tags
echo -e "\nListing tags for each table:"
for ARN in $TABLE_ARNS; do
    TABLE_NAME=$(echo $ARN | awk -F/ '{print $2}')
    echo -e "\nTags for table: $TABLE_NAME"
    aws dynamodb list-tags-of-resource --resource-arn $ARN
```

```
done

# Example: Find tables with a specific tag
echo -e "\nFinding tables with Environment=Production tag:"
for ARN in $TABLE_ARNS; do
    TABLE_NAME=$(echo $ARN | awk -F/ '{print $2}')
    TAGS=$(aws dynamodb list-tags-of-resource --resource-arn $ARN --query "Tags[?
Key=='Environment' && Value=='Production']" --output text)
    if [ ! -z "$TAGS" ]; then
        echo "Table with Production tag: $TABLE_NAME"
    fi
done
```

- API 세부 정보는 AWS CLI 명령 참조의 다음 토픽을 참조하세요.
  - CreatePolicy
  - CreateTable
  - ListTables

로컬 보조 인덱스로 작업

다음 코드 예제에서는 로컬 보조 인덱스가 있는 테이블을 생성하고 쿼리하는 방법을 보여줍니다.

- 로컬 보조 인덱스(LSI)가 있는 테이블을 생성합니다.
- 다양한 프로젝션 유형의 여러 LSI가 있는 테이블을 생성합니다.
- LSI를 사용하여 데이터를 쿼리합니다.

Bash 스크립트와 함께 AWS CLI사용

로컬 보조 인덱스가 있는 테이블을 생성합니다.

```
# Create a table with a Local Secondary Index
aws dynamodb create-table \
    --table-name CustomerOrders \
    --attribute-definitions \
        AttributeName=CustomerID,AttributeType=S \
        AttributeName=OrderID,AttributeType=S \
        AttributeName=OrderDate,AttributeType=S \
    --key-schema \
        AttributeName=CustomerID,KeyType=HASH \
```

```
        AttributeName=OrderID,KeyType=RANGE \
    --local-secondary-indexes \
        "IndexName=OrderDateIndex,\
        KeySchema=[{AttributeName=CustomerID,KeyType=HASH},
{AttributeName=OrderDate,KeyType=RANGE}],\
        Projection={ProjectionType=ALL}" \
    --billing-mode PAY_PER_REQUEST
```

여러 LSI가 있는 테이블을 생성합니다.

```
# Create a table with multiple Local Secondary Indexes
aws dynamodb create-table \
    --table-name CustomerDetails \
    --attribute-definitions \
        AttributeName=CustomerID,AttributeType=S \
        AttributeName=Name,AttributeType=S \
        AttributeName=Email,AttributeType=S \
        AttributeName=RegistrationDate,AttributeType=S \
    --key-schema \
        AttributeName=CustomerID,KeyType=HASH \
        AttributeName=Name,KeyType=RANGE \
    --local-secondary-indexes \
        "[
            {
                \"IndexName\": \"EmailIndex\",
                \"KeySchema\": [
                    {\"AttributeName\":\"CustomerID\",\"KeyType\":\"HASH\"},
                    {\"AttributeName\":\"Email\",\"KeyType\":\"RANGE\"}
                ],
                \"Projection\": {\"ProjectionType\":\"INCLUDE\",\"NonKeyAttributes
\":[\"Address\",\"Phone\"]}
            },
            {
                \"IndexName\": \"RegistrationIndex\",
                \"KeySchema\": [
                    {\"AttributeName\":\"CustomerID\",\"KeyType\":\"HASH\"},
                    {\"AttributeName\":\"RegistrationDate\",\"KeyType\":\"RANGE\"}
                ],
                \"Projection\": {\"ProjectionType\":\"KEYS_ONLY\"}
            }
        ]" \
    --billing-mode PAY_PER_REQUEST
```

LSI를 사용하여 데이터를 쿼리합니다.

```
# Query the OrderDateIndex LSI
aws dynamodb query \
    --table-name CustomerOrders \
    --index-name OrderDateIndex \
    --key-condition-expression "CustomerID = :custId AND OrderDate BETWEEN :date1
 AND :date2" \
    --expression-attribute-values '{
        ":custId": {"S": "C1"},
        ":date1": {"S": "2023-01-01"},
        ":date2": {"S": "2023-02-01"}
    }'

# Query with a filter expression
aws dynamodb query \
    --table-name CustomerOrders \
    --index-name OrderDateIndex \
    --key-condition-expression "CustomerID = :custId" \
    --filter-expression "Amount > :amount" \
    --expression-attribute-values '{
        ":custId": {"S": "C1"},
        ":amount": {"N": "150"}
    }'
```

- API 세부 정보는 AWS CLI 명령 참조의 다음 토픽을 참조하세요.
  - CreateTable
  - Query

Streams 및 Time-to-Live 작업

다음 코드 예제에서는 DynamoDB Streams 및 Time-to-Live 기능을 관리하는 방법을 보여줍니다.

- Streams를 활성화하여 테이블을 생성합니다.
- Streams를 설명합니다.
- Streams 처리를 위한 Lambda 함수를 생성합니다.
- 테이블에서 TTL을 활성화합니다.

- TTL 속성이 있는 항목을 추가합니다.
- TTL 설정을 설명합니다.

Bash 스크립트와 함께 AWS CLI사용

Streams를 활성화하여 테이블을 생성합니다.

```
# Create a table with DynamoDB Streams enabled
aws dynamodb create-table \
    --table-name StreamsDemo \
    --attribute-definitions \
        AttributeName=ID,AttributeType=S \
    --key-schema \
        AttributeName=ID,KeyType=HASH \
    --billing-mode PAY_PER_REQUEST \
    --stream-specification StreamEnabled=true,StreamViewType=NEW_AND_OLD_IMAGES
```

Streams를 설명합니다.

```
# Get information about the stream
aws dynamodb describe-table \
    --table-name StreamsDemo \
    --query "Table.StreamSpecification"

# Get the stream ARN
STREAM_ARN=$(aws dynamodb describe-table \
    --table-name StreamsDemo \
    --query "Table.LatestStreamArn" \
    --output text)

echo "Stream ARN: $STREAM_ARN"

# Describe the stream
aws dynamodbstreams describe-stream \
    --stream-arn $STREAM_ARN
```

Streams 관련 Lambda 함수를 생성합니다.

```
# Step 1: Create an IAM role for the Lambda function
```

```
cat > trust-policy.json << 'EOF'
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "lambda.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF

aws iam create-role \
    --role-name DynamoDBStreamsLambdaRole \
    --assume-role-policy-document file://trust-policy.json

# Step 2: Attach permissions to the role
aws iam attach-role-policy \
    --role-name DynamoDBStreamsLambdaRole \
    --policy-arn arn:aws:iam::aws:policy/service-role/AWSLambdaDynamoDBExecutionRole

# Step 3: Create a Lambda function (code would be in a separate file)
echo "Lambda function creation would be done separately with appropriate code"

# Step 4: Create an event source mapping
echo "Example command to create event source mapping:"
echo "aws lambda create-event-source-mapping \\"
echo "    --function-name ProcessDynamoDBRecords \\"
echo "    --event-source $STREAM_ARN \\"
echo "    --batch-size 100 \\"
echo "    --starting-position LATEST"
```

테이블에서 TTL을 활성화합니다.

```
# Create a table for TTL demonstration
aws dynamodb create-table \
    --table-name TTLDemo \
    --attribute-definitions \
        AttributeName=ID,AttributeType=S \
```

```
    --key-schema \
        AttributeName=ID,KeyType=HASH \
    --billing-mode PAY_PER_REQUEST

# Wait for table to become active
aws dynamodb wait table-exists --table-name TTLDemo

# Enable TTL on the table
aws dynamodb update-time-to-live \
    --table-name TTLDemo \
    --time-to-live-specification "Enabled=true, AttributeName=ExpirationTime"
```

TTL 속성이 있는 항목을 추가합니다.

```
# Calculate expiration time (current time + 1 day in seconds)
EXPIRATION_TIME=$(date -d "+1 day" +%s)

# Add an item with TTL attribute
aws dynamodb put-item \
    --table-name TTLDemo \
    --item '{
        "ID": {"S": "item1"},
        "Data": {"S": "This item will expire in 1 day"},
        "ExpirationTime": {"N": "'$EXPIRATION_TIME'"}
    }'

# Add an item that expires in 1 hour
EXPIRATION_TIME_HOUR=$(date -d "+1 hour" +%s)
aws dynamodb put-item \
    --table-name TTLDemo \
    --item '{
        "ID": {"S": "item2"},
        "Data": {"S": "This item will expire in 1 hour"},
        "ExpirationTime": {"N": "'$EXPIRATION_TIME_HOUR'"}
    }'
```

TTL 설정을 설명합니다.

```
# Describe TTL settings for a table
aws dynamodb describe-time-to-live \
    --table-name TTLDemo
```

- API 세부 정보는 AWS CLI 명령 참조의 다음 토픽을 참조하세요.
  - AttachRolePolicy
  - CreateRole
  - CreateTable
  - DescribeTable
  - DescribeTimeToLive
  - PutItem
  - UpdateTimeToLive

글로벌 테이블 및 다중 리전 복제 최종 일관성(MREC) 작업

다음 코드 예제에서는 최종 일관성이 있는 다중 리전 복제(MREC)를 사용하여 DynamoDB 글로벌 테이블을 관리하는 방법을 보여줍니다.

- 다중 리전 복제(MREC)를 사용하여 테이블을 만듭니다.
- 복제본 테이블에서 항목을 넣고 가져옵니다.
- 복제본을 하나씩 제거합니다.
- 테이블을 삭제하여 정리합니다.

Bash 스크립트와 함께 AWS CLI사용

다중 리전 복제를 사용하여 테이블을 생성합니다.

```
# Step 1: Create a new table (MusicTable) in US East (Ohio), with DynamoDB Streams
 enabled (NEW_AND_OLD_IMAGES)
aws dynamodb create-table \
    --table-name MusicTable \
    --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 \
    --stream-specification StreamEnabled=true,StreamViewType=NEW_AND_OLD_IMAGES \
```

```
    --region us-east-2

# Step 2: Create an identical MusicTable table in US East (N. Virginia)
aws dynamodb update-table --table-name MusicTable --cli-input-json \
'{
  "ReplicaUpdates":
  [
    {
      "Create": {
        "RegionName": "us-east-1"
      }
    }
  ]
}' \
--region us-east-2

# Step 3: Create a table in Europe (Ireland)
aws dynamodb update-table --table-name MusicTable --cli-input-json \
'{
  "ReplicaUpdates":
  [
    {
      "Create": {
        "RegionName": "eu-west-1"
      }
    }
  ]
}' \
--region us-east-2
```

다중 리전 테이블을 설명합니다.

```
# Step 4: View the list of replicas created using describe-table
aws dynamodb describe-table \
    --table-name MusicTable \
    --region us-east-2 \
    --query 'Table.
{TableName:TableName,TableStatus:TableStatus,MultiRegionConsistency:MultiRegionConsistency,R
{Region:RegionName,Status:ReplicaStatus}}'
```

복제본 테이블에 항목을 넣습니다.

```
# Step 5: To verify that replication is working, add a new item to the Music table
 in US East (Ohio)
aws dynamodb put-item \
    --table-name MusicTable \
    --item '{"Artist": {"S":"item_1"},"SongTitle": {"S":"Song Value 1"}}' \
    --region us-east-2
```

복제본 테이블에서 항목을 가져옵니다.

```
# Step 6: Wait for a few seconds, and then check to see whether the item has been
# successfully replicated to US East (N. Virginia) and Europe (Ireland)
aws dynamodb get-item \
    --table-name MusicTable \
    --key '{"Artist": {"S":"item_1"},"SongTitle": {"S":"Song Value 1"}}' \
    --region us-east-1

aws dynamodb get-item \
    --table-name MusicTable \
    --key '{"Artist": {"S":"item_1"},"SongTitle": {"S":"Song Value 1"}}' \
    --region eu-west-1
```

복제본을 제거합니다.

```
# Step 7: Delete the replica table in Europe (Ireland) Region
aws dynamodb update-table --table-name MusicTable --cli-input-json \
'{
  "ReplicaUpdates":
  [
    {
      "Delete": {
        "RegionName": "eu-west-1"
      }
    }
  ]
}' \
--region us-east-2

# Delete the replica table in US East (N. Virginia) Region
aws dynamodb update-table --table-name MusicTable --cli-input-json \
'{
```

```
  "ReplicaUpdates":
  [
    {
      "Delete": {
        "RegionName": "us-east-1"
      }
    }
  ]
}' \
--region us-east-2
```

테이블을 삭제하여 정리합니다.

```
# Clean up: Delete the primary table
aws dynamodb delete-table --table-name MusicTable --region us-east-2

echo "Global table demonstration complete."
```

- API 세부 정보는 AWS CLI 명령 참조의 다음 토픽을 참조하세요.
  - CreateTable
  - DeleteTable
  - DescribeTable
  - GetItem
  - PutItem
  - UpdateTable

리소스 태그 지정 작업

다음 코드 예제에서는 DynamoDB 리소스에 대한 태그를 관리하는 방법을 보여줍니다.

- 태그가 있는 테이블을 생성합니다.
- 리소스에 대한 태그를 나열합니다.
- 리소스에 태그를 추가합니다.
- 리소스에서 태그를 제거합니다.
- 태그를 기준으로 테이블을 필터링합니다.

Bash 스크립트와 함께 AWS CLI사용

태그가 있는 테이블을 생성합니다.

```
# Create a table with tags
aws dynamodb create-table \
    --table-name TaggedTable \
    --attribute-definitions \
        AttributeName=ID,AttributeType=S \
    --key-schema \
        AttributeName=ID,KeyType=HASH \
    --billing-mode PAY_PER_REQUEST \
    --tags \
        Key=Environment,Value=Production \
        Key=Project,Value=Analytics \
        Key=Owner,Value=DataTeam
```

리소스에 대한 태그를 나열합니다.

```
# Get the table ARN
TABLE_ARN=$(aws dynamodb describe-table \
    --table-name TaggedTable \
    --query "Table.TableArn" \
    --output text)

# List tags for the table
aws dynamodb list-tags-of-resource \
    --resource-arn $TABLE_ARN
```

리소스에 태그를 추가합니다.

```
# Add tags to an existing table
aws dynamodb tag-resource \
    --resource-arn $TABLE_ARN \
    --tags \
        Key=CostCenter,Value=12345 \
        Key=BackupSchedule,Value=Daily
```

리소스에서 태그를 제거합니다.

```
# Remove tags from a table
aws dynamodb untag-resource \
    --resource-arn $TABLE_ARN \
    --tag-keys Owner BackupSchedule
```

태그를 기준으로 테이블을 필터링합니다.

```
# Create another table with different tags
aws dynamodb create-table \
    --table-name AnotherTaggedTable \
    --attribute-definitions \
        AttributeName=ID,AttributeType=S \
    --key-schema \
        AttributeName=ID,KeyType=HASH \
    --billing-mode PAY_PER_REQUEST \
    --tags \
        Key=Environment,Value=Development \
        Key=Project,Value=Testing

# Wait for table to become active
aws dynamodb wait table-exists --table-name AnotherTaggedTable

# List all tables
echo "All tables:"
aws dynamodb list-tables

# Get ARNs for all tables
echo -e "\nFiltering tables by Environment=Production tag:"
TABLE_ARNS=$(aws dynamodb list-tables --query "TableNames[*]" --output text | xargs
 -I {} aws dynamodb describe-table --table-name {} --query "Table.TableArn" --output
 text)

# Find tables with specific tag
for ARN in $TABLE_ARNS; do
    TABLE_NAME=$(echo $ARN | awk -F/ '{print $2}')
    TAGS=$(aws dynamodb list-tags-of-resource --resource-arn $ARN --query "Tags[?
Key=='Environment' && Value=='Production']" --output text)
    if [ ! -z "$TAGS" ]; then
        echo "Table with Production tag: $TABLE_NAME"
    fi
done
```

- API 세부 정보는 AWS CLI 명령 참조의 다음 토픽을 참조하세요.
  - CreateTable
  - ListTagsOfResource
  - TagResource
  - UntagResource

테이블 암호화 작업

다음 코드 예제에서는 DynamoDB 테이블의 암호화 옵션을 관리하는 방법을 보여줍니다.

- 기본 암호화를 사용하여 테이블을 생성합니다.
- 고객 관리형 CMK를 사용하여 테이블을 생성합니다.
- 테이블 암호화 설정을 업데이트합니다.
- 테이블 암호화를 설명합니다.

Bash 스크립트와 함께 AWS CLI사용

기본 암호화를 사용하여 테이블을 생성합니다.

```
# Create a table with default encryption (AWS owned key)
aws dynamodb create-table \
    --table-name CustomerData \
    --attribute-definitions \
        AttributeName=CustomerID,AttributeType=S \
    --key-schema \
        AttributeName=CustomerID,KeyType=HASH \
    --billing-mode PAY_PER_REQUEST \
    --sse-specification Enabled=true,SSEType=KMS
```

고객 관리형 CMK를 사용하여 테이블을 생성합니다.

```
# Step 1: Create a customer managed key in KMS
aws kms create-key \
    --description "Key for DynamoDB table encryption" \
```

```
    --key-usage ENCRYPT_DECRYPT \
    --customer-master-key-spec SYMMETRIC_DEFAULT

# Store the key ID for later use
KEY_ID=$(aws kms list-keys --query "Keys[?contains(KeyArn, 'Key for
 DynamoDB')].KeyId" --output text)

# Step 2: Create a table with the customer managed key
aws dynamodb create-table \
    --table-name SensitiveData \
    --attribute-definitions \
        AttributeName=RecordID,AttributeType=S \
    --key-schema \
        AttributeName=RecordID,KeyType=HASH \
    --billing-mode PAY_PER_REQUEST \
    --sse-specification Enabled=true,SSEType=KMS,KMSMasterKeyId=$KEY_ID
```

테이블 암호화를 업데이트합니다.

```
# Update a table to use a different KMS key
aws dynamodb update-table \
    --table-name CustomerData \
    --sse-specification Enabled=true,SSEType=KMS,KMSMasterKeyId=$KEY_ID
```

테이블 암호화를 설명합니다.

```
# Describe the table to see encryption settings
aws dynamodb describe-table \
    --table-name CustomerData \
    --query "Table.SSEDescription"
```

- API 세부 정보는 AWS CLI 명령 참조의 다음 토픽을 참조하세요.
  - CreateKey
  - CreateTable
  - DescribeTable
  - UpdateTable

Bash 스크립트와 함께 AWS CLI를 사용하는 Amazon EC2 예제

다음 코드 예제에서는 Amazon EC2에서 Bash 스크립트와 함께 AWS Command Line Interface 코드를 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 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

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 예제는 다음과 같은 작업을 수행하는 방법을 보여줍니다.

- 키 페어 및 보안 그룹을 생성합니다.
- Amazon Machine Image(AMI) 및 호환되는 인스턴스 유형을 선택한 다음 인스턴스를 생성합니다.
- 인스턴스를 중지한 후 다시 시작합니다.
- 인스턴스와 탄력적 IP 주소 연결.
- SSH로 인스턴스에 연결한 다음 리소스를 정리합니다.

Bash 스크립트와 함께 AWS CLI사용

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS코드 예 리포지토리에서 전체 예를 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 세부 정보는 AWS CLI 명령 참조의 다음 토픽을 참조하세요.
  - AllocateAddress
  - AssociateAddress
  - AuthorizeSecurityGroupIngress
  - CreateKeyPair
  - CreateSecurityGroup
  - DeleteKeyPair
  - DeleteSecurityGroup
  - DescribeImages
  - DescribeInstanceTypes
  - DescribeInstances
  - DescribeKeyPairs
  - DescribeSecurityGroups
  - DisassociateAddress
  - ReleaseAddress
  - RunInstances
  - StartInstances
  - StopInstances
  - TerminateInstances
  - UnmonitorInstances

작업

`AllocateAddress`

다음 코드 예시는 `AllocateAddress`의 사용 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. [AWS코드 예 리포지토리]에서 전체 예를 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 AllocateAddress를 참조하세요.

AssociateAddress

다음 코드 예시는 `AssociateAddress`의 사용 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS코드 예 리포지토리에서 전체 예를 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [AssociateAddress](#)를 참조하세요.

`AuthorizeSecurityGroupIngress`

다음 코드 예시는 `AuthorizeSecurityGroupIngress`의 사용 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. [AWS코드 예 리포지토리](#)에서 전체 예를 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 AuthorizeSecurityGroupIngress를 참조하세요.

CreateKeyPair

다음 코드 예시는 CreateKeyPair의 사용 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS코드 예 리포지토리에서 전체 예를 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateKeyPair를 참조하세요.

CreateSecurityGroup

다음 코드 예제에서는 `CreateSecurityGroup` 코드를 사용하는 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS코드 예 리포지토리에서 전체 예를 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateSecurityGroup을 참조하세요.

`DeleteKeyPair`

다음 코드 예시는 `DeleteKeyPair`의 사용 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

> Note
>
> GitHub에 더 많은 내용이 있습니다. AWS코드 예 리포지토리에서 전체 예를 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteKeyPair를 참조하세요.

`DeleteSecurityGroup`

다음 코드 예제에서는 `DeleteSecurityGroup` 코드를 사용하는 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS코드 예 리포지토리에서 전체 예를 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteSecurityGroup을 참조하세요.

DescribeImages

다음 코드 예시는 DescribeImages의 사용 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS코드 예 리포지토리에서 전체 예를 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeImages를 참조하세요.

DescribeInstanceTypes

다음 코드 예시는 DescribeInstanceTypes의 사용 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

Note

GitHub에 더 많은 내용이 있습니다. AWS코드 예 리포지토리에서 전체 예를 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [DescribeInstanceTypes](#)를 참조하세요.

`DescribeInstances`

다음 코드 예시는 `DescribeInstances`의 사용 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

Note

GitHub에 더 많은 내용이 있습니다. AWS코드 예 리포지토리에서 전체 예를 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeInstances를 참조하세요.

DescribeKeyPairs

다음 코드 예시는 `DescribeKeyPairs`의 사용 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS코드 예 리포지토리에서 전체 예를 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeKeyPairs를 참조하세요.

DescribeSecurityGroups

다음 코드 예시는 `DescribeSecurityGroups`의 사용 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

Note

GitHub에 더 많은 내용이 있습니다. AWS코드 예 리포지토리에서 전체 예를 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DescribeSecurityGroups를 참조하세요.

DisassociateAddress

다음 코드 예시는 `DisassociateAddress`의 사용 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. [AWS코드 예 리포지토리]에서 전체 예를 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DisassociateAddress를 참조하세요.

`ReleaseAddress`

다음 코드 예시는 `ReleaseAddress`의 사용 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

Note

GitHub에 더 많은 내용이 있습니다. AWS코드 예 리포지토리에서 전체 예를 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ReleaseAddress를 참조하세요.

RunInstances

다음 코드 예시는 RunInstances의 사용 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. [AWS코드 예 리포지토리]에서 전체 예를 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 RunInstances를 참조하세요.

StartInstances

다음 코드 예시는 StartInstances의 사용 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS코드 예 리포지토리에서 전체 예를 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 StartInstances를 참조하세요.

StopInstances

다음 코드 예시는 `StopInstances`의 사용 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

> Note
>
> GitHub에 더 많은 내용이 있습니다. AWS코드 예 리포지토리에서 전체 예를 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 StopInstances를 참조하세요.

`TerminateInstances`

다음 코드 예시는 `TerminateInstances`의 사용 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS코드 예 리포지토리에서 전체 예를 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 TerminateInstances를 참조하세요.

Bash 스크립트와 함께 AWS CLI를 사용하는 HealthImaging 예제

다음 코드 예제에서는 HealthImaging에서 Bash 스크립트와 함께 AWS Command Line Interface 코드를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

`CreateDatastore`

다음 코드 예시는 `CreateDatastore`의 사용 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

```
###############################################################################
# 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateDatastore를 참조하세요.

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

DeleteDatastore

다음 코드 예시는 `DeleteDatastore`의 사용 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

```
###############################################################################
# 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteDatastore를 참조하세요.

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

GetDatastore

다음 코드 예시는 GetDatastore의 사용 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

```
###############################################################################
# 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetDatastore를 참조하세요.

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. [AWS코드 예 리포지토리](https://github.com/awsdocs/aws-doc-sdk-examples)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

ListDatastores

다음 코드 예시는 ListDatastores의 사용 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

```
###############################################################################
# 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListDatastores를 참조하세요.

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS코드 예 리포지토리에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

Bash 스크립트와 함께 AWS CLI를 사용하는 IAM 예제

다음 코드 예제에서는 IAM에서 Bash 스크립트와 함께 AWS Command Line Interface를 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.

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

- 기본 사항
- 작업
- 시나리오

기본 사항

기본 사항 알아보기

다음 코드 예제에서는 사용자를 생성하고 역할을 수임하는 방법을 보여줍니다.

> **Warning**
>
> 보안 위험을 방지하려면 목적별 소프트웨어를 개발하거나 실제 데이터로 작업할 때 IAM 사용자를 인증에 사용하지 마세요. 대신 AWS IAM Identity Center과 같은 자격 증명 공급자를 통한 페더레이션을 사용하세요.

- 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.

Bash 스크립트와 함께 AWS CLI사용

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. [AWS코드 예 리포지토리]에서 전체 예를 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 시나리오에 사용된 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 AWS 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 AWS 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 AWS 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 AWS 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 AWS 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 세부 정보는 AWS CLI 명령 참조의 다음 토픽을 참조하세요.
  - AttachRolePolicy
  - CreateAccessKey
  - CreatePolicy
  - CreateRole
  - CreateUser
  - DeleteAccessKey
  - DeletePolicy
  - DeleteRole
  - DeleteUser

- DeleteUserPolicy
- DetachRolePolicy
- PutUserPolicy

작업

AttachRolePolicy

다음 코드 예시는 AttachRolePolicy의 사용 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예제 리포지토리에서 전체 예제를 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 AttachRolePolicy를 참조하세요.

`CreateAccessKey`

다음 코드 예시는 `CreateAccessKey`의 사용 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예제 리포지토리에서 전체 예제를 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateAccessKey를 참조하세요.

CreatePolicy

다음 코드 예시는 CreatePolicy의 사용 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예제 리포지토리]에서 전체 예제를 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"
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreatePolicy를 참조하세요.

CreateRole

다음 코드 예제에서는 CreateRole 코드를 사용하는 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예제 리포지토리에서 전체 예제를 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateRole을 참조하세요.

CreateUser

다음 코드 예시는 `CreateUser`의 사용 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예제 리포지토리에서 전체 예제를 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 AWS 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [CreateUser](#)를 참조하세요.

DeleteAccessKey

다음 코드 예시는 `DeleteAccessKey`의 사용 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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 AWS Identity and Access Management (IAM) access key for the
 specified IAM user"
    echo "  -u user_name    The name of the user."
    echo "  -k access_key   The access key to delete."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "u:k:h" option; do
    case "${option}" in
      u) user_name="${OPTARG}" ;;
      k) access_key="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$user_name" ]]; then
    errecho "ERROR: You must provide a username with the -u parameter."
    usage
    return 1
  fi

  if [[ -z "$access_key" ]]; then
    errecho "ERROR: You must provide an access key with the -k parameter."
    usage
    return 1
  fi

  iecho "Parameters:\n"
```

```
  iecho "    Username:   $user_name"
  iecho "    Access key:   $access_key"
  iecho ""

  response=$(aws iam delete-access-key \
    --user-name "$user_name" \
    --access-key-id "$access_key")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-access-key operation failed.\n$response"
    return 1
  fi

  iecho "delete-access-key response:$response"
  iecho

  return 0
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteAccessKey를 참조하세요.

DeletePolicy

다음 코드 예시는 DeletePolicy의 사용 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예제 리포지토리에서 전체 예제를 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 AWS 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeletePolicy를 참조하세요.

`DeleteRole`

다음 코드 예제에서는 `DeleteRole` 코드를 사용하는 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예제 리포지토리](https://github.com/awsdocs/aws-doc-sdk-examples)에서 전체 예제를 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 AWS Identity and Access Management (IAM) role"
    echo "  -n role_name -- The name of the IAM role."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:h" option; do
    case "${option}" in
      n) role_name="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  echo "role_name:$role_name"
  if [[ -z "$role_name" ]]; then
    errecho "ERROR: You must provide a role name with the -n parameter."
    usage
    return 1
  fi

  iecho "Parameters:\n"
  iecho "    Role name:  $role_name"
  iecho ""

  response=$(aws iam delete-role \
    --role-name "$role_name")
```

```
  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-role operation failed.\n$response"
    return 1
  fi

  iecho "delete-role response:$response"
  iecho

  return 0
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteRole을 참조하세요.

DeleteUser

다음 코드 예시는 DeleteUser의 사용 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예제 리포지토리에서 전체 예제를 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 AWS 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 세부 정보는 AWS CLI 명령 참조의 DeleteUser를 참조하세요.

`DetachRolePolicy`

다음 코드 예시는 `DetachRolePolicy`의 사용 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [DetachRolePolicy](#)를 참조하세요.

GetUser

다음 코드 예시는 GetUser의 사용 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetUser를 참조하세요.

ListAccessKeys

다음 코드 예시는 ListAccessKeys의 사용 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예제 리포지토리에서 전체 예제를 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 ListAccessKeys를 참조하세요.

ListUsers

다음 코드 예시는 ListUsers의 사용 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. [AWS 코드 예제 리포지토리]에서 전체 예제를 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
    }
```

- API 세부 정보는 AWS CLI 명령 참조의 ListUsers를 참조하세요.

UpdateAccessKey

다음 코드 예시는 UpdateAccessKey의 사용 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. [AWS코드 예 리포지토리]에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
###############################################################################
# function iam_update_access_key
#
# This function can activate or deactivate 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 update.
#       -a            -- Activate the selected access key.
#       -d            -- Deactivate the selected access key.
#
# Example:
#       # To deactivate the selected access key for IAM user Bob
#       iam_update_access_key -u Bob -k AKIAIOSFODNN7EXAMPLE -d
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_update_access_key() {
  local user_name access_key status response
  local option OPTARG # Required to use getopts command in a function.
  local activate_flag=false deactivate_flag=false

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_update_access_key"
    echo "Updates the status of an AWS 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 update."
    echo "  -a              Activate the access key."
    echo "  -d              Deactivate the access key."
```

```
    echo ""
  }

  # Retrieve the calling parameters.
    while getopts "u:k:adh" option; do
      case "${option}" in
        u) user_name="${OPTARG}" ;;
        k) access_key="${OPTARG}" ;;
        a) activate_flag=true ;;
        d) deactivate_flag=true ;;
        h)
          usage
          return 0
          ;;
        \?)
          echo "Invalid parameter"
          usage
          return 1
          ;;
      esac
    done
    export OPTIND=1

   # Validate input parameters
    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

    # Ensure that only -a or -d is specified
    if [[ "$activate_flag" == true && "$deactivate_flag" == true ]]; then
      errecho "ERROR: You cannot specify both -a (activate) and -d (deactivate) at
 the same time."
      usage
      return 1
    fi
```

```
    # If neither -a nor -d is provided, return an error
    if [[ "$activate_flag" == false && "$deactivate_flag" == false ]]; then
      errecho "ERROR: You must specify either -a (activate) or -d (deactivate)."
      usage
      return 1
    fi

    # Determine the status based on the flag
    if [[ "$activate_flag" == true ]]; then
      status="Active"
    elif [[ "$deactivate_flag" == true ]]; then
      status="Inactive"
    fi

    iecho "Parameters:\n"
    iecho "    Username:   $user_name"
    iecho "    Access key: $access_key"
    iecho "    New status: $status"
    iecho ""

    # Update the access key status
    response=$(aws iam update-access-key \
      --user-name "$user_name" \
      --access-key-id "$access_key" \
      --status "$status" 2>&1)

    local error_code=${?}

    if [[ $error_code -ne 0 ]]; then
      aws_cli_error_log $error_code
      errecho "ERROR: AWS reports update-access-key operation failed.\n$response"
      return 1
    fi

    iecho "update-access-key response: $response"
    iecho

    return 0
}
```

- API 세부 정보는 AWS CLI 명령 참조의 UpdateAccessKey를 참조하세요.

시나리오

속성 기반 액세스 제어 설정

다음 코드 예제에서는 DynamoDB에 대한 속성 기반 액세스 제어(ABAC)를 구현하는 방법을 보여줍니다.

- ABAC를 위한 IAM 정책을 생성합니다.
- 부서별 태그가 있는 테이블을 생성합니다.
- 태그를 기반으로 테이블을 나열하고 필터링합니다.

Bash 스크립트와 함께 AWS CLI사용

ABAC를 위한 IAM 정책을 생성합니다.

```
# Step 1: Create a policy document for ABAC
cat > abac-policy.json << 'EOF'
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "dynamodb:GetItem",
        "dynamodb:BatchGetItem",
        "dynamodb:Query",
        "dynamodb:Scan"
      ],
      "Resource": "arn:aws:dynamodb:*:*:table/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Department": "${aws:PrincipalTag/Department}"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "dynamodb:PutItem",
        "dynamodb:UpdateItem",
        "dynamodb:DeleteItem",
        "dynamodb:BatchWriteItem"
```

```
      ],
      "Resource": "arn:aws:dynamodb:*:*:table/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Department": "${aws:PrincipalTag/Department}",
          "aws:ResourceTag/Environment": "Development"
        }
      }
    }
  ]
}
EOF

# Step 2: Create the IAM policy
aws iam create-policy \
    --policy-name DynamoDBDepartmentBasedAccess \
    --policy-document file://abac-policy.json
```

부서별 태그가 있는 테이블을 생성합니다.

```
# Create a DynamoDB table with tags for ABAC
aws dynamodb create-table \
    --table-name FinanceData \
    --attribute-definitions \
        AttributeName=RecordID,AttributeType=S \
    --key-schema \
        AttributeName=RecordID,KeyType=HASH \
    --billing-mode PAY_PER_REQUEST \
    --tags \
        Key=Department,Value=Finance \
        Key=Environment,Value=Development

# Create another table with different tags
aws dynamodb create-table \
    --table-name MarketingData \
    --attribute-definitions \
        AttributeName=RecordID,AttributeType=S \
    --key-schema \
        AttributeName=RecordID,KeyType=HASH \
    --billing-mode PAY_PER_REQUEST \
    --tags \
        Key=Department,Value=Marketing \
```

```
        Key=Environment,Value=Production
```

태그를 기반으로 테이블을 나열하고 필터링합니다.

```
# List all DynamoDB tables
echo "Listing all tables:"
aws dynamodb list-tables

# Get ARNs for all tables
echo -e "\nGetting ARNs for all tables:"
TABLE_ARNS=$(aws dynamodb list-tables --query "TableNames[*]" --output text | xargs
 -I {} aws dynamodb describe-table --table-name {} --query "Table.TableArn" --output
 text)

# For each table ARN, list its tags
echo -e "\nListing tags for each table:"
for ARN in $TABLE_ARNS; do
    TABLE_NAME=$(echo $ARN | awk -F/ '{print $2}')
    echo -e "\nTags for table: $TABLE_NAME"
    aws dynamodb list-tags-of-resource --resource-arn $ARN
done

# Example: Find tables with a specific tag
echo -e "\nFinding tables with Environment=Production tag:"
for ARN in $TABLE_ARNS; do
    TABLE_NAME=$(echo $ARN | awk -F/ '{print $2}')
    TAGS=$(aws dynamodb list-tags-of-resource --resource-arn $ARN --query "Tags[?
Key=='Environment' && Value=='Production']" --output text)
    if [ ! -z "$TAGS" ]; then
        echo "Table with Production tag: $TABLE_NAME"
    fi
done
```

- API 세부 정보는 AWS CLI 명령 참조의 다음 토픽을 참조하세요.
  - CreatePolicy
  - CreateTable
  - ListTables

Streams 및 Time-to-Live 작업

다음 코드 예제에서는 DynamoDB Streams 및 Time-to-Live 기능을 관리하는 방법을 보여줍니다.

- Streams를 활성화하여 테이블을 생성합니다.
- Streams를 설명합니다.
- Streams 처리를 위한 Lambda 함수를 생성합니다.
- 테이블에서 TTL을 활성화합니다.
- TTL 속성이 있는 항목을 추가합니다.
- TTL 설정을 설명합니다.

Bash 스크립트와 함께 AWS CLI사용

Streams를 활성화하여 테이블을 생성합니다.

```
# Create a table with DynamoDB Streams enabled
aws dynamodb create-table \
    --table-name StreamsDemo \
    --attribute-definitions \
        AttributeName=ID,AttributeType=S \
    --key-schema \
        AttributeName=ID,KeyType=HASH \
    --billing-mode PAY_PER_REQUEST \
    --stream-specification StreamEnabled=true,StreamViewType=NEW_AND_OLD_IMAGES
```

Streams를 설명합니다.

```
# Get information about the stream
aws dynamodb describe-table \
    --table-name StreamsDemo \
    --query "Table.StreamSpecification"

# Get the stream ARN
STREAM_ARN=$(aws dynamodb describe-table \
    --table-name StreamsDemo \
    --query "Table.LatestStreamArn" \
    --output text)

echo "Stream ARN: $STREAM_ARN"
```

```
# Describe the stream
aws dynamodbstreams describe-stream \
    --stream-arn $STREAM_ARN
```

Streams 관련 Lambda 함수를 생성합니다.

```
# Step 1: Create an IAM role for the Lambda function
cat > trust-policy.json << 'EOF'
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "lambda.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF

aws iam create-role \
    --role-name DynamoDBStreamsLambdaRole \
    --assume-role-policy-document file://trust-policy.json

# Step 2: Attach permissions to the role
aws iam attach-role-policy \
    --role-name DynamoDBStreamsLambdaRole \
    --policy-arn arn:aws:iam::aws:policy/service-role/AWSLambdaDynamoDBExecutionRole

# Step 3: Create a Lambda function (code would be in a separate file)
echo "Lambda function creation would be done separately with appropriate code"

# Step 4: Create an event source mapping
echo "Example command to create event source mapping:"
echo "aws lambda create-event-source-mapping \\"
echo "    --function-name ProcessDynamoDBRecords \\"
echo "    --event-source $STREAM_ARN \\"
echo "    --batch-size 100 \\"
echo "    --starting-position LATEST"
```

테이블에서 TTL을 활성화합니다.

```
# Create a table for TTL demonstration
aws dynamodb create-table \
    --table-name TTLDemo \
    --attribute-definitions \
        AttributeName=ID,AttributeType=S \
    --key-schema \
        AttributeName=ID,KeyType=HASH \
    --billing-mode PAY_PER_REQUEST

# Wait for table to become active
aws dynamodb wait table-exists --table-name TTLDemo

# Enable TTL on the table
aws dynamodb update-time-to-live \
    --table-name TTLDemo \
    --time-to-live-specification "Enabled=true, AttributeName=ExpirationTime"
```

TTL 속성이 있는 항목을 추가합니다.

```
# Calculate expiration time (current time + 1 day in seconds)
EXPIRATION_TIME=$(date -d "+1 day" +%s)

# Add an item with TTL attribute
aws dynamodb put-item \
    --table-name TTLDemo \
    --item '{
        "ID": {"S": "item1"},
        "Data": {"S": "This item will expire in 1 day"},
        "ExpirationTime": {"N": "'$EXPIRATION_TIME'"}
    }'

# Add an item that expires in 1 hour
EXPIRATION_TIME_HOUR=$(date -d "+1 hour" +%s)
aws dynamodb put-item \
    --table-name TTLDemo \
    --item '{
        "ID": {"S": "item2"},
        "Data": {"S": "This item will expire in 1 hour"},
```

```
        "ExpirationTime": {"N": "'$EXPIRATION_TIME_HOUR'"}
    }'
```

TTL 설정을 설명합니다.

```
# Describe TTL settings for a table
aws dynamodb describe-time-to-live \
    --table-name TTLDemo
```

- API 세부 정보는 AWS CLI 명령 참조의 다음 토픽을 참조하세요.
  - AttachRolePolicy
  - CreateRole
  - CreateTable
  - DescribeTable
  - DescribeTimeToLive
  - PutItem
  - UpdateTimeToLive

Bash 스크립트와 함께 AWS CLI를 사용하는 AWS KMS 예제

다음 코드 예제에서는 AWS KMS에서 Bash 스크립트와 함께 AWS Command Line Interface 코드를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

시나리오는 동일한 서비스 내에서 또는 다른 AWS 서비스와 결합된 상태에서 여러 함수를 호출하여 특정 태스크를 수행하는 방법을 보여주는 코드 예제입니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 시나리오

시나리오

테이블 암호화 작업

다음 코드 예제에서는 DynamoDB 테이블의 암호화 옵션을 관리하는 방법을 보여줍니다.

- 기본 암호화를 사용하여 테이블을 생성합니다.
- 고객 관리형 CMK를 사용하여 테이블을 생성합니다.
- 테이블 암호화 설정을 업데이트합니다.
- 테이블 암호화를 설명합니다.

Bash 스크립트와 함께 AWS CLI사용

기본 암호화를 사용하여 테이블을 생성합니다.

```
# Create a table with default encryption (AWS owned key)
aws dynamodb create-table \
    --table-name CustomerData \
    --attribute-definitions \
        AttributeName=CustomerID,AttributeType=S \
    --key-schema \
        AttributeName=CustomerID,KeyType=HASH \
    --billing-mode PAY_PER_REQUEST \
    --sse-specification Enabled=true,SSEType=KMS
```

고객 관리형 CMK를 사용하여 테이블을 생성합니다.

```
# Step 1: Create a customer managed key in KMS
aws kms create-key \
    --description "Key for DynamoDB table encryption" \
    --key-usage ENCRYPT_DECRYPT \
    --customer-master-key-spec SYMMETRIC_DEFAULT

# Store the key ID for later use
KEY_ID=$(aws kms list-keys --query "Keys[?contains(KeyArn, 'Key for
 DynamoDB')].KeyId" --output text)

# Step 2: Create a table with the customer managed key
aws dynamodb create-table \
    --table-name SensitiveData \
```

```
    --attribute-definitions \
        AttributeName=RecordID,AttributeType=S \
    --key-schema \
        AttributeName=RecordID,KeyType=HASH \
    --billing-mode PAY_PER_REQUEST \
    --sse-specification Enabled=true,SSEType=KMS,KMSMasterKeyId=$KEY_ID
```

테이블 암호화를 업데이트합니다.

```
# Update a table to use a different KMS key
aws dynamodb update-table \
    --table-name CustomerData \
    --sse-specification Enabled=true,SSEType=KMS,KMSMasterKeyId=$KEY_ID
```

테이블 암호화를 설명합니다.

```
# Describe the table to see encryption settings
aws dynamodb describe-table \
    --table-name CustomerData \
    --query "Table.SSEDescription"
```

- API 세부 정보는 AWS CLI 명령 참조의 다음 토픽을 참조하세요.
  - CreateKey
  - CreateTable
  - DescribeTable
  - UpdateTable

Bash 스크립트와 함께 AWS CLI를 사용하는 Amazon S3 예제

다음 코드 예제에서는 Amazon S3에서 Bash 스크립트와 함께 AWS Command Line Interface 코드를 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 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

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 예제는 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.

Bash 스크립트와 함께 AWS CLI사용

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS코드 예 리포지토리에서 전체 예를 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 세부 정보는 AWS CLI 명령 참조의 다음 토픽을 참조하세요.
  - CopyObject
  - CreateBucket
  - DeleteBucket

- DeleteObjects
- GetObject
- ListObjectsV2
- PutObject

작업

CopyObject

다음 코드 예시는 CopyObject의 사용 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예제 리포지토리에서 전체 예제를 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CopyObject를 참조하세요.

CreateBucket

다음 코드 예제에서는 CreateBucket 코드를 사용하는 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예제 리포지토리에서 전체 예제를 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 CreateBucket을 참조하세요.

DeleteBucket

다음 코드 예제에서는 `DeleteBucket` 코드를 사용하는 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예제 리포지토리에서 전체 예제를 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 DeleteBucket을 참조하세요.

DeleteObject

다음 코드 예시는 DeleteObject의 사용 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예제 리포지토리에서 전체 예제를 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteBucket](#)을 참조하세요.

`DeleteObjects`

다음 코드 예시는 `DeleteObjects`의 사용 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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
}
```

• API 세부 정보는 AWS CLI 명령 참조의 DeleteObjects를 참조하세요.

GetObject

다음 코드 예시는 GetObject의 사용 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

> Note
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예제 리포지토리에서 전체 예제를 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 GetObject를 참조하세요.

HeadBucket

다음 코드 예제에서는 HeadBucket 코드를 사용하는 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예제 리포지토리에서 전체 예제를 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 HeadBucket을 참조하세요.

ListObjectsV2

다음 코드 예시는 `ListObjectsV2`의 사용 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예제 리포지토리에서 전체 예제를 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 세부 정보는 AWS CLI 명령 참조의 ListObjectsV2를 참조하세요.

PutObject

다음 코드 예시는 PutObject의 사용 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. [AWS 코드 예제 리포지토리]에서 전체 예제를 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
}
```

- API 세부 정보는 AWS CLI 명령 참조의 PutObject를 참조하세요.

Bash 스크립트와 함께 AWS CLI를 사용하는 AWS STS 예제

다음 코드 예제에서는 AWS STS에서 Bash 스크립트와 함께 AWS Command Line Interface 코드를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 관련 시나리오의 컨텍스트에 따라 표시되며, 개별 서비스 함수를 직접적으로 호출하는 방법을 보여줍니다.

각 예시에는 전체 소스 코드에 대한 링크가 포함되어 있으며, 여기에서 컨텍스트에 맞춰 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있습니다.

주제

- 작업

작업

AssumeRole

다음 코드 예제에서는 AssumeRole 코드를 사용하는 방법을 보여 줍니다.

Bash 스크립트와 함께 AWS CLI사용

> **Note**
>
> GitHub에 더 많은 내용이 있습니다. AWS 코드 예제 리포지토리에서 전체 예제를 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
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done

  response=$(aws sts assume-role \
    --role-session-name "$role_session_name" \
    --role-arn "$role_arn" \
    --output text \
    --query "Credentials.[AccessKeyId, SecretAccessKey, SessionToken]")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-role operation failed.\n$response"
    return 1
  fi

  echo "$response"

  return 0
}
```

- API 세부 정보는 AWS CLI 명령 참조의 AssumeRole을 참조하세요.

AWS CLI의 보안

AWS는 클라우드 보안을 가장 중요하게 생각합니다. AWS 고객으로서 여러분은 가장 높은 보안 요구 사항을 충족하기 위해 설계된 데이터 센터 및 네트워크 아키텍처의 혜택을 받게 됩니다.

보안은 AWS와 귀하의 공동 책임입니다. 공동 책임 모델 에서는 이를 클라우드 자체의 보안과 클라우드 내부의 보안으로 설명합니다.

- 클라우드의 보안 - AWS는 AWS 클라우드에서 AWS 서비스를 실행하는 인프라를 보호합니다. AWS는 또한 안전하게 사용할 수 있는 서비스를 제공합니다. 서드 파티 감사자는 AWS 규정 준수 프로그램의 일환으로 보안 효과를 정기적으로 테스트하고 검증합니다. AWS Command Line Interface에 적용되는 규정 준수 프로그램에 대한 자세한 내용은 규정 준수 프로그램의 범위에 속하는 AWS 서비스를 참조하십시오.
- 클라우드의 보안 – 귀하의 책임은 귀하가 사용하는 AWS 서비스에 의해 결정됩니다. 또한 귀하는 데이터의 민감도, 회사 요구 사항, 관련 법률 및 규정을 비롯한 기타 요소에 대해서도 책임이 있습니다.

이 설명서는 AWS Command Line Interface(AWS CLI)를 사용할 때 공동 책임 모델을 적용하는 방법을 이해하는 데 도움이 됩니다. 다음 주제에서는 보안 및 규정 준수 목표를 충족하도록 AWS CLI를 구성하는 방법을 보여줍니다. 또한 AWS CLI를 사용하여 AWS 리소스를 모니터링하고 보호하는 방법에 대해서도 알아봅니다.

주제

AWS CLI에서 데이터 보호

AWS 공동 책임 모델은 AWS Command Line Interface의 데이터 보호에 적용됩니다. 이 모델에서 설명하는 것처럼 AWS는 모든 AWS 클라우드를 실행하는 글로벌 인프라를 보호할 책임이 있습니다. 사용자는 인프라에서 호스팅되는 콘텐츠를 관리해야 합니다. 사용하는 AWS 서비스의 보안 구성과 관리 작업에 대한 책임도 사용자에게 있습니다. 데이터 프라이버시에 대한 자세한 내용은 데이터 프라이버시

FAQ를 참조하세요. 유럽의 데이터 보호에 대한 자세한 내용은 AWS 보안 블로그의 AWS 공동 책임 모델 및 GDPR 블로그 게시물을 참조하세요.

데이터를 보호하려면 AWS 계정보안 인증 정보를 보호하고 AWS IAM Identity Center또는 AWS Identity and Access Management(IAM)를 통해 개별 사용자 계정을 설정하는 것이 좋습니다. 이렇게 하면 개별 사용자에게 자신의 직무를 충실히 이행하는 데 필요한 권한만 부여됩니다. 또한 다음과 같은 방법으로 데이터를 보호하는 것이 좋습니다.

- 각 계정에 멀티 팩터 인증 설정(MFA)을 사용하세요.
- SSL/TLS를 사용하여 AWS 리소스와 통신하세요. TLS 1.2는 필수이며 TLS 1.3을 권장합니다.
- AWS CloudTrail로 API 및 사용자 활동 로깅을 설정하세요. AWS 활동 캡처에 CloudTrail 추적을 사용하는 방법에 대한 자세한 내용은 AWS CloudTrail 사용 설명서의 CloudTrail 추적 작업을 참조하세요.
- AWS 암호화 솔루션을 AWS 서비스 내의 모든 기본 보안 컨트롤과 함께 사용하세요.
- Amazon S3에 저장된 민감한 데이터를 검색하고 보호하는 데 도움이 되는 Amazon Macie와 같은 고급 관리형 보안 서비스를 사용하세요.
- 명령행 인터페이스 또는 API를 통해 AWS에 액세스할 때 FIPS 140-3 검증된 암호화 모듈이 필요한 경우, FIPS 엔드포인트를 사용합니다. 사용 가능한 FIPS 엔드포인트에 대한 자세한 내용은 Federal Information Processing Standard(FIPS) 140-3을 참조하세요.

고객의 이메일 주소와 같은 기밀 정보나 중요한 정보는 태그나 이름 필드와 같은 자유 양식 필드에 입력하지 않는 것이 좋습니다. 여기에는 AWS CLI 또는 기타 AWS 서비스에서 콘솔, API, AWS CLI 또는 AWS SDK를 사용하여 작업하는 경우가 포함됩니다. 이름에 사용되는 태그 또는 자유 형식 텍스트 필드에 입력하는 모든 데이터는 청구 또는 진단 로그에 사용될 수 있습니다. 외부 서버에 URL을 제공할 때 해당 서버에 대한 요청을 검증하기 위해 보안 인증 정보를 URL에 포함해서는 안 됩니다.

데이터 암호화

보안 서비스의 주요 특징은 정보가 활발히 사용되지 않을 때 암호화된다는 것입니다.

저장된 암호화

AWS CLI는 사용자를 대신하여 AWS 서비스와 상호 작용하는 데 필요한 자격 증명 이외의 고객 데이터를 저장하지 않습니다.

AWS CLI를 사용하여 저장을 위해 로컬 컴퓨터로 고객 데이터를 전송하는 AWS 서비스를 호출하는 경우 해당 데이터가 저장, 보호 및 암호화되는 방법에 대한 자세한 내용은 해당 서비스 사용 설명서의 보안 및 규정 준수 장을 참조하십시오.

전송 중 데이터 암호화

기본적으로 AWS CLI 및 AWS 서비스 엔드포인트를 실행하는 클라이언트 컴퓨터에서 전송되는 모든 데이터는 HTTPS/TLS 연결을 통해 모든 데이터를 전송하여 암호화됩니다.

HTTPS/TLS 사용을 활성화하기 위해 어떤 조치도 필요하지 않습니다. `--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 제어할 수 있도록 지원하는 AWS 서비스입니다. IAM 관리자는 어떤 사용자가 AWS 리소스를 사용할 수 있는 인증(로그인) 및 권한(권한 있음)을 받을 수 있는지 제어합니다. IAM은 추가 비용 없이 사용할 수 있는 AWS 서비스입니다.

주제

대상

AWS Identity and Access Management(IAM)을 사용하는 방법은 AWS에서 수행하는 작업에 따라 달라집니다.

서비스 사용자 – AWS 서비스를 사용하여 작업을 수행하는 경우 필요한 보안 인증 정보와 권한을 관리자가 제공합니다. 더 많은 AWS 기능을 사용하여 작업을 수행하게 되면 추가 권한이 필요할 수 있습니다. 액세스 권한 관리 방법을 이해하면 관리자에게 올바른 권한을 요청하는 데 도움이 됩니다. AWS의 기능에 액세스할 수 없는 경우 AWS 보안 인증 및 액세스 문제 해결 또는 사용 중인 AWS 서비스의 사용 설명서를 참조하세요.

서비스 관리자 - 회사에서 AWS 리소스를 책임지고 있는 담당자라면 AWS에 대한 전체 액세스 권한을 가지고 있을 것입니다. 서비스 관리자는 서비스 사용자가 액세스해야 하는 AWS 기능과 리소스를 결정합니다. 그런 다음, IAM 관리자에게 요청을 제출하여 서비스 사용자의 권한을 변경해야 합니다. 이 페

이지의 정보를 검토하여 IAM의 기본 개념을 이해하세요. 회사에서 AWS와 함께 IAM을 사용하는 방법에 대한 자세한 내용은 사용 중인 AWS 서비스의 사용 설명서를 참조하세요.

IAM 관리자 - IAM 관리자라면 AWS에 대한 액세스 권한 관리 정책 작성 방법을 자세히 알고 싶을 것입니다. IAM에서 사용할 수 있는 AWS 보안 인증 기반 정책 예제를 보려면 사용 중인 AWS 서비스의 사용 설명서를 참조하세요.

보안 인증을 통한 인증

인증은 ID 자격 증명을 사용하여 AWS에 로그인하는 방식입니다. AWS 계정 루트 사용자이나 IAM 사용자로, 또는 IAM 역할을 수임하여 인증(AWS에 로그인)받아야 합니다.

ID 소스를 통해 제공된 자격 증명을 사용하여 페더레이션 ID로 AWS에 로그인할 수 있습니다. AWS IAM Identity Center (IAM Identity Center) 사용자, 회사의 Single Sign-On 인증, Google 또는 Facebook 보안 인증이 페더레이션 ID의 예제입니다. 페더레이션형 ID로 로그인할 때 관리자가 이전에 IAM 역할을 사용하여 ID 페더레이션을 설정했습니다. 연동을 사용하여 AWS에 액세스하면 간접적으로 역할을 수임합니다.

사용자 유형에 따라AWS Management Console 또는 AWS 액세스 포털에 로그인할 수 있습니다. AWS에 로그인하는 방법에 대한 자세한 내용은 AWS 로그인 사용 설명서의 AWS 계정에 로그인하는 방법을 참조하세요.

AWS에 프로그래밍 방식으로 액세스하는 경우, AWS에서는 자격 증명을 사용하여 요청에 암호화 방식으로 서명할 수 있는 소프트웨어 개발 키트(SDK) 및 명령줄 인터페이스(CLI)를 제공합니다. AWS 도구를 사용하지 않는 경우, 요청에 직접 서명해야 합니다. 권장 방법을 사용하여 요청에 직접 서명하는 자세한 방법은 IAM 사용 설명서에서 API 요청용 AWS Signature Version 4를 참조하세요.

사용하는 인증 방법에 상관없이 추가 보안 정보를 제공해야 할 수도 있습니다. 예를 들어, AWS은(는) 다중 인증(MFA)을 사용하여 계정의 보안을 강화하는 것을 권장합니다. 자세한 내용은 AWS IAM Identity Center 사용 설명서에서 다중 인증 및 IAM 사용 설명서에서 IAM의 AWS 다중 인증을 참조하세요.

AWS 계정 루트 사용자

AWS 계정을(를) 생성할 때는 해당 계정의 모든 AWS 서비스 및 리소스에 대한 완전한 액세스 권한이 있는 단일 로그인 ID로 시작합니다. 이 ID는 AWS 계정루트 사용자라고 하며, 계정을 생성할 때 사용한 이메일 주소와 암호로 로그인하여 액세스합니다. 일상적인 작업에 루트 사용자를 사용하지 않을 것을 강력히 권장합니다. 루트 사용자 자격 증명을 보호하고 루트 사용자만 수행할 수 있는 작업을 수행하는 데 사용합니다. 루트 사용자로 로그인해야 하는 전체 작업 목록은 IAM 사용 설명서의 루트 사용자 보안 인증이 필요한 작업을 참조하세요.

페더레이션 자격 증명

가장 좋은 방법은 관리자 액세스가 필요한 사용자를 포함한 사용자가 ID 공급자와의 페더레이션을 사용하여 임시 자격 증명을 사용하여 AWS 서비스에 액세스하도록 요구합니다.

페더레이션 ID는 엔터프라이즈 사용자 디렉터리, 웹 ID 공급자, AWS Directory Service, Identity Center 디렉터리의 사용자 또는 자격 증명 소스를 통해 제공된 자격 증명을 사용하여 AWS 서비스에 액세스하는 모든 사용자입니다. 페더레이션 ID는 AWS 계정에 액세스할 때 역할을 수임하고 역할은 임시 자격 증명을 제공합니다.

중앙 집중식 액세스 관리를 위해 AWS IAM Identity Center을(를) 사용하는 것이 좋습니다. IAM Identity Center에서 사용자 및 그룹을 생성하거나 모든 AWS 계정 및 애플리케이션에서 사용하기 위해 고유한 ID 소스의 사용자 및 그룹 집합에 연결하고 동기화할 수 있습니다. IAM Identity Center에 대한 자세한 내용은 AWS IAM Identity Center 사용 설명서에서 IAM Identity Center란 무엇인가요?를 참조하세요.

IAM 사용자 및 그룹

IAM 사용자는 단일 개인 또는 애플리케이션에 대한 특정 권한을 가지고 있는 AWS 계정 내 ID입니다. 가능하면 암호 및 액세스 키와 같은 장기 자격 증명이 있는 IAM 사용자를 생성하는 대신 임시 자격 증명을 사용하는 것이 좋습니다. 하지만 IAM 사용자의 장기 자격 증명이 필요한 특정 사용 사례가 있는 경우, 액세스 키를 교체하는 것이 좋습니다. 자세한 내용은 IAM 사용 설명서의 장기 보안 인증이 필요한 사용 사례의 경우, 정기적으로 액세스 키 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, IAMAdmins라는 그룹이 있고 이 그룹에 IAM 리소스를 관리할 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 자격 증명을 가지고 있지만, 역할은 임시 보안 인증만 제공합니다. 자세한 내용은 IAM 사용 설명서에서 IAM 사용자 사용 사례를 참조하세요.

IAM 역할

IAM 역할은 특정 권한을 가지고 있는 AWS 계정 계정 내 ID입니다. IAM 사용자와 유사하지만, 특정 개인과 연결되지 않습니다. AWS Management Console에서 일시적으로 IAM 역할을 수임하려면 사용자에서 IAM 역할로 전환(콘솔)하면 됩니다. AWS CLI 또는 AWS API 작업을 직접적으로 호출하거나 사용자 지정 URL을 사용하여 역할을 수임할 수 있습니다. 역할 사용 방법에 대한 자세한 내용은 IAM 사용 설명서의 역할 수임 방법을 참조하세요.

임시 보안 인증이 있는 IAM 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 관련 역할에 대한 자세한 내용은 IAM 사용 설명서의 [Create a role for a third-party identity provider (federation)]를 참조하세요. IAM Identity Center를 사용하는 경우, 권한 집합을 구성합니다. 인증 후 ID가 액세스할 수 있는 항목을 제어하기 위해 IAM Identity Center는 권한 집합을 IAM의 역할과 연관짓습니다. 권한 집합에 대한 자세한 내용은 AWS IAM Identity Center 사용 설명서의 [권한 집합]을 참조하세요.
- 임시 IAM 사용자 권한 - IAM 사용자 또는 역할은 IAM 역할을 수임하여 특정 작업에 대한 다양한 권한을 임시로 받을 수 있습니다.
- 교차 계정 액세스 - IAM 역할을 사용하여 다른 계정의 사용자(신뢰할 수 있는 보안 주체)가 내 계정의 리소스에 액세스하도록 허용할 수 있습니다. 역할은 계정 간 액세스를 부여하는 기본적인 방법입니다. 하지만 일부 AWS 서비스을(를) 사용하면 리소스에 정책을 직접 연결할 수 있습니다(역할을 프록시로서 사용하는 대신) . 교차 계정 액세스에 대한 역할과 리소스 기반 정책의 차이점을 알아보려면 IAM 사용 설명서의 [IAM의 교차 계정 리소스 액세스]를 참조하세요.
- 교차 서비스 액세스 - 일부 AWS 서비스는 다른 AWS 서비스의 특성을 사용합니다. 예를 들어, 서비스에서 호출하면 일반적으로 해당 서비스는 Amazon EC2에서 애플리케이션을 실행하거나 Amazon S3에 객체를 저장합니다. 서비스는 직접적으로 호출하는 위탁자의 권한을 사용하거나, 서비스 역할을 사용하거나, 또는 서비스 연결 역할을 사용하여 이 작업을 수행할 수 있습니다.
  - 전달 액세스 세션(FAS) - IAM 사용자 또는 역할을 사용하여 AWS에서 작업을 수행하는 사람은 보안 주체로 간주됩니다. 일부 서비스를 사용하는 경우, 다른 서비스에서 다른 작업을 시작하는 작업을 수행할 수 있습니다. FAS는 AWS 서비스를 직접 호출하는 위탁자의 권한과 요청하는 AWS 서비스를 함께 사용하여 다운스트림 서비스에 대한 요청을 수행합니다. FAS 요청은 서비스에서 완료를 위해 다른 AWS 서비스 또는 리소스와의 상호 작용이 필요한 요청을 받은 경우에만 이루어집니다. 이 경우, 두 작업을 모두 수행할 수 있는 권한이 있어야 합니다. FAS 요청 시 정책 세부 정보는 [전달 액세스 세션]을 참조하세요.
  - 서비스 역할 - 서비스 역할은 서비스가 사용자를 대신하여 작업을 수행하기 위해 맡는 [IAM 역할]입니다. IAM 관리자는 IAM 내에서 서비스 역할을 생성, 수정 및 삭제할 수 있습니다. 자세한 정보는 IAM 사용 설명서의 [Create a role to delegate permissions to an AWS 서비스]를 참조하세요.
  - 서비스 연결 역할 – 서비스 연결 역할은 AWS 서비스에 연결된 서비스 역할의 한 유형입니다. 서비스는 사용자를 대신하여 작업을 수행하기 위해 역할을 수임할 수 있습니다. 서비스 링크 역할은 AWS 계정에 나타나고, 서비스가 소유합니다. IAM 관리자는 서비스 연결 역할의 권한을 볼 수 있지만 편집할 수는 없습니다.

- Amazon EC2에서 실행 중인 애플리케이션 – IAM 역할을 사용하여 EC2 인스턴스에서 실행되고 AWS CLI 또는 AWS API 요청을 수행하는 애플리케이션의 임시 자격 증명을 관리할 수 있습니다. 이는 EC2 인스턴스 내에 액세스 키를 저장할 때 권장되는 방법입니다. EC2 인스턴스에 AWS역할을 할당하고 해당 역할을 모든 애플리케이션에서 사용할 수 있도록 하려면 인스턴스에 연결된 인스턴스 프로필을 생성합니다. 인스턴스 프로필에는 역할이 포함되어 있으며 EC2 인스턴스에서 실행되는 프로그램이 임시 보안 인증을 얻을 수 있습니다. 자세한 정보는 IAM 사용 설명서의 IAM 역할을 사용하여 Amazon EC2 인스턴스에서 실행되는 애플리케이션에 권한 부여를 참조하세요.

정책을 사용하여 액세스 관리

정책을 생성하고 AWS ID 또는 리소스에 연결하여 AWS에서 내 액세스를 제어합니다. 정책은 ID 또는 리소스와 연결될 때 해당 권한을 정의하는 AWS의 객체입니다. AWS는 보안 주체(사용자, 루트 사용자 또는 역할 세션)가 요청을 보낼 때 이러한 정책을 평가합니다. 정책에서 권한은 요청이 허용되거나 거부되는 지를 결정합니다. 대부분의 정책은 AWS에 JSON 문서로 저장됩니다. JSON 정책 문서의 구조와 콘텐츠에 대한 자세한 정보는 IAM 사용 설명서의 JSON 정책 개요를 참조하세요.

관리자는 AWSJSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지를 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

기본적으로, 사용자 및 역할에는 어떠한 권한도 없습니다. 사용자에게 사용자가 필요한 리소스에서 작업을 수행할 권한을 부여하려면 IAM 관리자가 IAM 정책을 생성하면 됩니다. 그런 다음 관리자가 IAM 정책을 역할에 추가하고, 사용자가 역할을 수임할 수 있습니다.

IAM 정책은 작업을 수행하기 위해 사용하는 방법과 상관없이 작업에 대한 권한을 정의합니다. 예를 들어, `iam:GetRole` 작업을 허용하는 정책이 있다고 가정합니다. 해당 정책이 있는 사용자는 AWS Management Console, AWS CLI 또는 AWS API에서 역할 정보를 가져올 수 있습니다.

ID 기반 정책

ID 기반 정책은 IAM 사용자, 사용자 그룹 또는 역할과 같은 ID에 연결할 수 있는 JSON 권한 정책 문서입니다. 이러한 정책은 사용자 및 역할이 어떤 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 제어합니다. 자격 증명 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서에서 고객 관리형 정책으로 사용자 지정 IAM 권한 정의를 참조하세요.

ID 기반 정책은 인라인 정책 또는 관리형 정책으로 한층 더 분류할 수 있습니다. 인라인 정책은 단일 사용자, 그룹 또는 역할에 직접 포함됩니다. 관리형 정책은 AWS 계정에 속한 다수의 사용자, 그룹 및 역할에 독립적으로 추가할 수 있는 정책입니다. 관리형 정책에는 AWS 관리형 정책과 고객 관리형 정책이 포함되어 있습니다. 관리형 정책 또는 인라인 정책을 선택하는 방법을 알아보려면 IAM 사용 설명서의 관리형 정책 및 인라인 정책 중에서 선택을 참조하세요.

리소스 기반 정책

리소스 기반 정책은 리소스에 연결하는 JSON 정책 설명서입니다. 리소스 기반 정책의 예제는 IAM 역할 신뢰 정책과 Amazon S3 버킷 정책입니다. 리소스 기반 정책을 지원하는 서비스에서 서비스 관리자는 이러한 정책을 사용하여 특정 리소스에 대한 액세스를 통제할 수 있습니다. 정책이 연결된 리소스의 경우 정책은 지정된 위탁자가 해당 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 정의합니다. 리소스 기반 정책에서 위탁자를 지정해야 합니다. 보안 주체에는 계정, 사용자, 역할, 페더레이션 사용자 또는 AWS 서비스가 포함될 수 있습니다.

리소스 기반 정책은 해당 서비스에 있는 인라인 정책입니다. 리소스 기반 정책에서는 IAM의 AWS 관리형 정책을 사용할 수 없습니다.

액세스 제어 목록(ACL)

액세스 제어 목록(ACL)은 어떤 보안 주체(계정 멤버, 사용자 또는 역할)가 리소스에 액세스할 수 있는 권한을 가지고 있는지를 제어합니다. ACLs는 JSON 정책 문서 형식을 사용하지 않지만 리소스 기반 정책과 유사합니다.

Amazon S3, AWS WAF 및 Amazon VPC는 ACL을 지원하는 대표적인 서비스입니다. ACL에 관한 자세한 내용은 Amazon Simple Storage Service 개발자 가이드의 액세스 제어 목록(ACL) 개요를 참조하세요.

기타 정책 타입

AWS는 비교적 일반적이지 않은 추가 정책 유형을 지원합니다. 이러한 정책 타입은 더 일반적인 정책 유형에 따라 사용자에게 부여되는 최대 권한을 설정할 수 있습니다.

- 권한 경계 – 권한 경계는 ID 기반 정책에 따라 IAM 엔터티(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 ID 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 정보는 IAM 사용 설명서의 IAM 엔티티에 대한 권한 경계를 참조하세요.
- 서비스 제어 정책(SCP) – SCP는 AWS Organizations에서 조직 또는 조직 단위(OU)에 최대 권한을 지정하는 JSON 정책입니다. AWS Organizations는 기업이 소유하는 여러 개의 AWS 계정을 그룹화하고 중앙에서 관리하기 위한 서비스입니다. 조직에서 모든 특성을 활성화할 경우, 서비스 제어 정책(SCP)을 임의의 또는 모든 계정에 적용할 수 있습니다. SCP는 각 AWS 계정 루트 사용자을 비롯하여 멤버 계정의 엔티티에 대한 권한을 제한합니다. 조직 및 SCP에 대한 자세한 내용은 AWS Organizations 사용 설명서에서 Service control policies을 참조하세요.

- 리소스 제어 정책(RCP) - RCP는 소유한 각 리소스에 연결된 IAM 정책을 업데이트하지 않고 계정의 리소스에 대해 사용 가능한 최대 권한을 설정하는 데 사용할 수 있는 JSON 정책입니다. RCP는 멤버 계정의 리소스에 대한 권한을 제한하며 조직에 속하는지 여부에 관계없이 AWS 계정 루트 사용자를 포함한 ID에 대한 유효 권한에 영향을 줄 수 있습니다. RCP를 지원하는 AWS 서비스 목록을 포함하여 Organizations 및 RCP에 대한 자세한 내용은 AWS Organizations 사용 설명서의 리소스 제어 정책(RCP)을 참조하세요.
- 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 ID 기반 정책의 교차와 세션 정책입니다. 또한 권한을 리소스 기반 정책에서 가져올 수도 있습니다. 이러한 정책 중 하나에 포함된 명시적 거부는 허용을 재정의합니다. 자세한 정보는 IAM 사용 설명서의 세션 정책을 참조하세요.

여러 정책 유형

여러 정책 유형이 요청에 적용되는 경우, 결과 권한은 이해하기가 더 복잡합니다. 여러 정책 유형이 관련될 때 AWS가 요청을 허용할지를 결정하는 방법을 알아보려면 IAM 사용 설명서의 정책 평가 로직을 참조하세요.

AWS 서비스에서 IAM을 사용하는 방식

AWS 서비스에서 대부분의 IAM 기능을 사용하는 방법을 전체적으로 알아보려면 IAM 사용 설명서의 IAM으로 작업하는 AWS 서비스를 참조하세요.

특정 AWS 서비스에 IAM을 사용하는 방법을 알아보려면 관련 서비스 사용 설명서의 보안 섹션을 참조하세요.

AWS 보안 인증 및 액세스 문제 해결

다음 정보를 사용하여 AWS 및 IAM에서 발생할 수 있는 공통적인 문제를 진단하고 수정할 수 있습니다.

주제

- AWS에서 작업을 수행할 권한이 없음
- iam:PassRole을 수행하도록 인증되지 않음
- 내 AWS 계정 외부의 사람이 내 AWS 리소스에 액세스할 수 있게 허용하기를 원합니다.

AWS에서 작업을 수행할 권한이 없음

작업을 수행할 권한이 없다는 오류가 수신되면, 작업을 수행할 수 있도록 정책을 업데이트해야 합니다.

다음의 예제 오류는 mateojackson IAM 사용자가 콘솔을 사용하여 가상 *my-example-widget* 리소스에 대한 세부 정보를 보려고 하지만 가상 awes:*GetWidget* 권한이 없을 때 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 mateojackson 사용자 정책을 업데이트해야 합니다.

도움이 필요한 경우 AWS 관리자에게 문의하세요. 관리자는 로그인 자격 증명을 제공한 사람입니다.

iam:PassRole을 수행하도록 인증되지 않음

iam:PassRole 작업을 수행할 수 있는 권한이 없다는 오류가 수신되면 AWS에 역할을 전달할 수 있도록 정책을 업데이트해야 합니다.

일부 AWS 서비스에서는 새 서비스 역할 또는 서비스 연결 역할을 생성하는 대신, 해당 서비스에 기존 역할을 전달할 수 있습니다. 이렇게 하려면 사용자가 서비스에 역할을 전달할 수 있는 권한을 가지고 있어야 합니다.

다음 예 오류는 marymajor라는 IAM 사용자가 콘솔을 사용하여 AWS에서 작업을 수행하려고 하는 경우에 발생합니다. 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 보안 인증 정보를 제공하는 사람입니다.

내 AWS 계정 외부의 사람이 내 AWS 리소스에 액세스할 수 있게 허용하기를 원합니다.

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 제

어 목록(ACL)을 지원하는 서비스의 경우, 이러한 정책을 사용하여 다른 사람에게 리소스에 대한 액세스 권한을 부여할 수 있습니다.

자세히 알아보려면 다음을 참조하세요.

- AWS에서 이러한 기능을 지원하는지 여부를 알아보려면 AWS 서비스에서 IAM을 사용하는 방식을 참조하세요.
- 소유하고 있는 AWS 계정의 리소스에 대한 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 자신이 소유한 다른 AWS 계정의 IAM 사용자에 대한 액세스 권한 제공을 참조하세요.
- 리소스에 대한 액세스 권한을 서드 파티 AWS 계정에게 제공하는 방법을 알아보려면 IAM 사용 설명서의 서드 파티가 소유한 AWS 계정에 대한 액세스 제공을 참조하세요.
- ID 페더레이션을 통해 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 외부에서 인증된 사용자에게 액세스 권한 제공(ID 페더레이션)을 참조하세요.
- 크로스 계정 액세스에 대한 역할과 리소스 기반 정책 사용의 차이점을 알아보려면 IAM 사용 설명서의 IAM의 크로스 계정 리소스 액세스를 참조하세요.

이 AWS 제품 또는 서비스에 대한 규정 준수 확인

AWS 서비스가 특정 규정 준수 프로그램의 범위에 포함되는지 알아보려면 규정 준수 프로그램 제공 범위 내 AWS 서비스를 참조하고 관심 있는 규정 준수 프로그램을 선택하세요. 일반적인 정보는 AWS 규정 준수 프로그램을 참조하세요.

AWS Artifact(을)를 사용하여 타사 감사 보고서를 다운로드할 수 있습니다. 자세한 내용은 AWS Artifact에서 보고서 다운로드를 참조하세요.

AWS 서비스 사용 시 규정 준수 책임은 데이터의 민감도, 회사의 규정 준수 목표 및 관련 법률과 규정에 따라 결정됩니다.AWS에서는 규정 준수를 지원할 다음과 같은 리소스를 제공합니다.

- 보안 규정 준수 및 거버넌스 - 이러한 솔루션 구현 가이드에서는 아키텍처 고려 사항을 설명하고 보안 및 규정 준수 기능을 배포하는 단계를 제공합니다.
- HIPAA 적격 서비스 참조 - HIPAA 적격 서비스가 나열되어 있습니다. 모든 AWS 서비스에 HIPAA 자격이 있는 것은 아닙니다.
- AWS 규정 준수 리소스 - 고객 조직이 속한 산업 및 위치에 적용될 수 있는 워크북 및 가이드 컬렉션입니다.
- AWS 고객 규정 준수 가이드 - 규정 준수의 관점에서 공동 책임 모델을 이해합니다. 이 가이드에서는 AWS 서비스를 보호하기 위한 모범 사례를 요약하고 여러 프레임워크(미국 표준 기술 연구소(NIST),

결제 카드 산업 보안 표준 위원회(PCI), 국제 표준화기구(ISO) 등)에서 보안 컨트롤에 대한 지침을 매핑합니다.

- AWS Config 개발자 가이드의 규칙을 사용하여 리소스 평가 – AWS Config 서비스는 내부 사례, 산업 지침 및 규제에 대한 리소스 구성의 준수 상태를 평가합니다.
- AWS Security Hub – 이 AWS 서비스(은)는 AWS 내 보안 상태에 대한 포괄적인 보기를 제공합니다. Security Hub는 보안 컨트롤을 사용하여 AWS리소스를 평가하고 보안 업계 표준 및 모범 사례에 대한 규정 준수를 확인합니다. 지원되는 서비스 및 제어 목록은 Security Hub 제어 참조를 참조하세요.
- Amazon GuardDuty - 이 AWS 서비스는 의심스럽고 악의적인 활동이 있는지 환경을 모니터링하여 AWS 계정, 워크로드, 컨테이너 및 데이터에 대한 잠재적 위협을 탐지합니다. GuardDuty는 특정 규정 준수 프레임워크에서 요구하는 침입 탐지 요구 사항을 충족하여 PCI DSS와 같은 다양한 규정 준수 요구 사항을 따르는 데 도움을 줄 수 있습니다.
- AWS Audit Manager - 이 AWS 서비스은(는) AWS 사용을 지속해서 감사하여 위험을 관리하고 규정 및 업계 표준을 준수하는 방법을 간소화할 수 있도록 지원합니다.

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 AWS 규정 준수 프로그램의 AWS 규정 준수 업무 범위에 속하는 서비스를 참조하세요.

이 AWS 제품 또는 서비스에 대한 복원력

AWS 글로벌 인프라는 AWS 리전 및 가용 영역을 중심으로 구축됩니다.

AWS 리전에서는 물리적으로 분리되고 격리된 다수의 가용 영역을 제공하며 이러한 가용 영역은 짧은 지연 시간, 높은 처리량 및 높은 중복성을 갖춘 네트워크에 연결되어 있습니다.

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 인프라를 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 AWS 규정 준수 프로그램의 AWS 규정 준수 업무 범위에 속하는 서비스를 참조하세요.

이 AWS 제품 또는 서비스에 대한 인프라 보안

이 AWS 제품 또는 서비스는 관리형 서비스를 사용하므로 AWS 글로벌 네트워크 보안으로 보호됩니다. AWS 보안 서비스와 AWS의 인프라 보호 방법에 대한 자세한 내용은 AWS 클라우드 보안을 참조하세요. 인프라 보안에 대한 모범 사례를 사용하여 AWS 환경을 설계하려면 보안 원칙 AWS Well-Architected Framework의 인프라 보호를 참조하세요.

AWS에서 게시한 API 직접 호출을 사용하여 네트워크를 통해 이 AWS 제품 또는 서비스에 액세스합니다. 고객은 다음을 지원해야 합니다.

- Transport Layer Security(TLS) TLS 1.2는 필수이며 TLS 1.3을 권장합니다.
- DHE(Ephemeral Diffie-Hellman) 또는 ECDHE(Elliptic Curve Ephemeral Diffie-Hellman)와 같은 완전 전송 보안(PFS)이 포함된 암호 제품군 Java 7 이상의 최신 시스템은 대부분 이러한 모드를 지원합니다.

또한 요청은 액세스 키 ID 및 IAM 위탁자와 관련된 보안 암호 액세스 키를 사용하여 서명해야 합니다. 또는 AWS Security Token Service(AWS STS)를 사용하여 임시 보안 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 AWS 규정 준수 프로그램의 AWS 규정 준수 업무 범위에 속하는 서비스를 참조하세요.

AWS CLI에 최소 버전의 TLS 적용

AWS Command Line Interface(AWS CLI)를 사용할 때, Transport Layer Security(TLS) 프로토콜은 AWS CLI와 AWS 서비스 간의 통신을 보호하는 데 중요한 역할을 합니다. AWS 서비스와 통신할 때 보안을 강화하려면 TLS 1.2 이상을 사용해야 합니다.

AWS CLI와 AWS 서비스는 암호화, 인증 및 데이터 무결성을 제공하는 TLS 프로토콜을 통해 데이터를 안전하게 교환할 수 있습니다. AWS CLI는 TLS 프로토콜을 활용하여 AWS 서비스와의 상호 작용을 무단 액세스 및 데이터 침해로부터 보호함으로써 AWS 에코시스템의 전반적인 보안을 강화합니다.

AWS 공동 책임 모델은 AWS Command Line Interface의 데이터 보호에 적용됩니다. 이 모델에서 설명하는 것처럼 AWS는 모든 AWS 서비스를 실행하는 글로벌 인프라를 보호할 책임이 있습니다. 사용자는 인프라에서 호스팅되는 콘텐츠를 관리해야 합니다. 사용하는 AWS 서비스의 보안 구성과 관리 작업에 대한 책임도 사용자에게 있습니다. 데이터 보호에 대한 자세한 내용은 the section called “데이터 보호” 섹션을 참조하세요.

AWS CLI 버전 1에서 TLS 1.2 이전의 TLS 버전을 사용하지 않도록 하려면 OpenSSL을 다시 컴파일하여 이 최솟값을 적용하고 새로 빌드된 OpenSSL을 사용하도록 Python을 다시 컴파일해야 할 수 있습니다.

주제

- 현재 지원되는 프로토콜 확인
- OpenSSL 및 Python 컴파일

현재 지원되는 프로토콜 확인

먼저 OpenSSL을 사용하여 테스트 서버 및 Python SDK에 사용할 자체 서명된 인증서를 만듭니다.

```
$ openssl req -subj '/CN=localhost' -x509 -newkey rsa:4096 -nodes -keyout key.pem -out
 cert.pem -days 365
```

그런 다음 OpenSSL을 사용하여 테스트 서버를 가동합니다.

```
$ openssl s_server -key key.pem -cert cert.pem -www
```

새 터미널 창에서 가상 환경을 만들고 Python용 SDK를 설치합니다.

```
$ python3 -m venv test-env
source test-env/bin/activate
pip install botocore
```

SDK의 기본 HTTP 라이브러리를 사용하는 `check.py`라는 새로운 Python 스크립트를 만듭니다.

```
$ 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" 이상이면 SDK는 기본적으로 TLS v1.2 이상으로 설정됩니다. 이전 버전인 경우 OpenSSL을 다시 컴파일하고 Python을 다시 컴파일해야 합니다.

그러나 Python이 기본적으로 TLS v1.2 이상으로 설치되더라도 서버가 TLS v1.2 이상을 지원하지 않으면 Python이 TLS v1.2 이전 버전으로 다시 협상할 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 OpenSSL을 사용하는 경우 `-no_tls_3` 플래그를 사용할 수 없을 수 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 버전에서 올바르게 다시 협상되지 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 TLS v1.2 이전의 프로토콜 협상을 비활성화하기 위해 OpenSSL과 Python을 다시 컴파일해야 합니다.

OpenSSL 및 Python 컴파일

SDK를 보장하거나 AWS CLI에서 TLS 1.2 이전 버전을 협상하지 않도록 하려면 OpenSSL과 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
```

이것은 TLS 1.2 이전 버전을 자동으로 협상하지 않는 정적으로 연결된 OpenSSL을 가진 Python 버전을 컴파일합니다. 또한 `/opt/openssl-with-min-tls1_2` 디렉터리에 OpenSSL을 설치하고 `/opt/python-with-min-tls1_2` 디렉터리에 Python을 설치합니다. 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 TLS 1.2 이전 버전을 협상하지 않는지 확인하려면 새로 설치된 Python 버전(즉, 현재 지원되는 프로토콜 확인)을 사용하는 `/opt/python-with-min-tls1_2/bin/python3`의 단계를 다시 실행합니다.

AWS CLI에 대한 오류 문제 해결

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

AWS CLI에서 오류 메시지가 표시되거나 문제가 발생할 경우 문제를 해결하는 데 도움이 되는 다음과 같은 일반적인 팁을 따르는 것이 좋습니다.

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.
- JSON의 경우 추가로 JSON 값에 대한 문제 해결을 참조하세요. 터미널 처리 JSON 형식에 문제가 있는 경우 Blob를 사용하여 JSON 데이터를 AWS CLI에 직접 전달함으로써 터미널의 인용 규칙을 건너뛰는 것이 좋습니다.

특정 명령을 구성하는 방법에 대한 자세한 내용은 AWS CLI 참조 가이드를 참조하세요.

맨 위로 이동

AWS CLI 명령이 사용 중인 AWS 리전 확인

> **Note**
>
> AWS CLI를 사용하여 명시적으로 또는 기본 리전을 설정하여 AWS 리전 리전을 지정해야 합니다. 지정할 수 있는 모든 AWS 리전 목록은 Amazon Web Services 일반 참조의 AWS 리전 및 엔드포인트를 참조하세요. AWS CLI에서 사용하는 AWS Management Console 표기는 AWS 리전 URL 및 서비스 엔드포인트에서 사용하는 것과 동일한 이름입니다.

지정한 AWS 리전에서 AWS 서비스를 사용할 수 없거나 리소스가 다른 AWS 리전에 있는 경우 오류 또는 예기치 않은 결과가 발생할 수 있습니다. AWS 리전은 우선 순위에 따라 다음과 같은 방식으로 설정됩니다.

- `--region` 명령줄 옵션
- `AWS_DEFAULT_REGION` 환경 변수.
- `region` 프로파일 설정입니다.

리소스에 대해 올바른 AWS 리전을 사용하고 있는지 확인하세요.

맨 위로 이동

최신 버전의 AWS CLI를 실행 중인지 확인합니다.

명령이 존재하지 않는다는 오류가 발생하거나 명령이 AWS CLI 참조 가이드에서 사용 가능하다고 나열된 파라미터를 인식하지 못하는 오류가 발생할 경우 먼저 명령 형식이 올바른지 확인합니다. 형식이 올바른 경우 AWS CLI의 최신 버전으로 업그레이드하는 것이 좋습니다. 업데이트된 AWS CLI 버전은 업무일 기준으로 거의 매일 릴리스됩니다. 새로운 AWS 서비스, 기능 및 파라미터가 이러한 새 버전의 AWS CLI에 반영됩니다. 새로운 서비스, 기능 또는 파라미터에 액세스할 수 있는 유일한 방법은 해당 요소가 도입된 이후 릴리스된 버전으로 업그레이드하는 것입니다.

AWS CLI 설치에 설명된 대로 AWS CLI의 버전 업데이트 방법은 원래 설치 방법에 따라 달라집니다.

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

AWS CLI에서 즉시 파악되지 않는 오류를 보고하거나 예상치 못한 결과를 생성하는 경우 --debug 옵션과 함께 명령을 다시 실행하여 오류에 대한 자세한 정보를 얻을 수 있습니다. 이 옵션을 사용하면 AWS CLI가 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 Support에 보낼 수 있습니다.

--debug 옵션을 포함하면 다음과 같은 세부 정보가 포함됩니다.

- 보안 인증 검색
- 제공된 파라미터 구문 분석
- AWS 서버에 보낸 요청 구성
- 에 보낸 요청의 내용AWS
- 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 AWS CLI에서 `aws` 명령 내역을 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.

- botocore에 대한 API 호출
- 상태 코드
- HTTP 응답
- 헤더
- 반환 코드

이 정보를 사용하여 파라미터 데이터 및 API 호출이 예상대로 작동하는지 확인한 다음, 프로세스의 어느 단계에서 명령이 실패했는지 추론할 수 있습니다.

맨 위로 이동

AWS CLI가 구성되었는지 확인

`config` 및 `credentials` 파일이나 IAM 사용자 또는 역할이 올바르게 구성되지 않은 경우 다양한 오류가 발생할 수 있습니다. `config` 및 `credentials` 파일이나 IAM 사용자 또는 역할과 관련된 오류 해결에 대한 자세한 내용은 the section called “액세스 거부 오류” 및 the section called “잘못된 보안 인증 정보 및 키 오류” 섹션을 참조하세요.

맨 위로 이동

명령을 찾을 수 없음 오류

이 오류는 운영 체제에서 AWS CLI 명령을 찾을 수 없음을 의미합니다. 설치가 불완전하거나 업데이트가 필요할 수 있습니다.

가능한 원인: 설치된 버전보다 최신 AWS CLI 기능을 사용하려고 하거나 형식이 잘못됨

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
```

AWS CLI를 처음 설치하거나 업데이트한 후 aws 명령을 찾을 수 없는 경우 PATH 업데이트를 인식하도록 터미널을 다시 시작해야 할 수 있습니다.

맨 위로 이동

가능한 원인: AWS CLI가 완전히 설치되지 않음

오류 텍스트 예:

```
$ aws --version
command not found: aws
```

AWS CLI를 처음 설치하거나 업데이트한 후 aws 명령을 찾을 수 없다면 완전히 설치되지 않았을 수 있습니다. 플랫폼에 해당하는 AWS CLI 설치 단계에 따라 다시 설치해 봅니다.

맨 위로 이동

가능한 원인: AWS CLI에 권한이 없음(Linux)

Linux에 AWS CLI를 처음 설치하거나 업데이트한 후 aws 명령을 찾을 수 없다면 완전히 설치된 폴더에 대한 `execute` 권한이 없기 때문일 수 있습니다. AWS CLI에 `chmod` 권한을 제공하려면 AWS CLI 설치에 다음 명령을 PATH와 함께 실행합니다.

```
$ 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. AWS CLI를 PATH에 추가하려면 사용 중인 운영 체제에 따라 다음에 나온 해당 지침을 따릅니다.

Linux and macOS

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. 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. aws를 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. 경로를 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 AWS CLI에 대해 예상과 다른 PATH를 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 AWS CLI가 있음

AWS CLI를 업데이트하면서 기존 설치와 다른 설치 방법을 사용한 경우 여러 버전이 설치될 수 있습니다. 예를 들어 Linux 또는 macOS에서 현재 설치에 `pip`를 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**
>
> 기존에 설치된 AWS CLI 버전 1과 함께 AWS CLI 버전 2를 설치한 후 이러한 문제가 발생하는 경우 AWS CLI 버전 1에서 마이그레이션할 때의 설치 지침에 나온 마이그레이션 지침을 따릅니다.

맨 위로 이동

AWS CLI를 제거한 후 '`aws --version`' 명령이 버전을 반환함

이는 대개 시스템 어딘가에 AWS CLI가 아직 설치되어 있는 경우 발생합니다.

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: 시스템에 여러 버전의 AWS CLI가 있거나 원래 AWS CLI를 설치하는 데 사용한 것과 동일한 제거 방법을 사용하지 않음

AWS CLI를 설치하는 데 사용한 것과 다른 방법을 사용하여 제거했거나 여러 버전을 설치한 경우 AWS CLI가 올바르게 제거되지 않을 수 있습니다. 예를 들어 현재 설치에 pip를 사용한 경우 pip를 사용하여 제거해야 합니다. 이 문제를 해결하려면 설치에 사용한 것과 동일한 방법을 사용하여 AWS CLI를 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 나열되어 있는 경우 AWS CLI가 다른 방법을 사용하여 설치되었거나 여러 버전이 있을 가능성이 큽니다. AWS CLI를 설치하는 데 사용한 방법을 모르는 경우 버전 출력이 표시되지 않을 때까지 운영 체제에 해당하는 AWS CLI 버전 1 및 AWS CLI 버전 2의 각 제거 방법 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

AWS CLI에서 불완전한 파라미터 이름을 가진 명령을 처리했습니다.

가능한 원인: AWS CLI 파라미터의 알려진 약어를 사용했습니다.

AWS CLI는 Python을 사용하여 빌드되었으므로 AWS CLI에서 [allow_abbrev](https://docs.python.org/3/library/argparse.html#allow-abbrev) 인수를 포함한 Python argparse 라이브러리를 사용합니다. 파라미터의 약어는 AWS CLI에 의해 인식되고 처리됩니다.

다음 create-change-set 명령 예제에서는 CloudFormation 스택 이름을 변경합니다. --change-set-n 파라미터는 --change-set-name의 약어로 인식되며 AWS CLI에서 명령을 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 예제에서는 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 CLI에서는 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 없음

Linux 또는 macOS에서 aws 프로그램이 호출하는 사용자에 대한 실행 권한을 가지고 있는지 확인합니다. 일반적으로 사용 권한은 755로 설정됩니다.

사용자에 대한 실행 권한을 추가하려면 다음 명령을 실행합니다. 이때 *~/.local/bin/aws*를 사용자 컴퓨터의 프로그램 경로로 바꾸세요.

```
$ chmod +x ~/.local/bin/aws
```

맨 위로 이동

가능한 원인: IAM 보안 인증에 작업을 수행할 수 있는 권한이 없음

오류 텍스트 예:

```
$ aws s3 ls
An error occurred (AcessDenied) when calling the ListBuckets operation: Access
 denied.
```

AWS CLI 명령을 실행할 때 이를 실행하는 사용자를 대신해서 해당 사용자를 IAM 계정 또는 역할과 연결하는 보안 인증 정보를 사용하여 AWS 작업이 수행됩니다. 연결된 정책은 사용자가 AWS CLI에서 실행하는 명령에 해당하는 API 작업을 호출할 수 있는 권한을 부여해야 합니다.

대부분의 명령은 명령 이름과 일치하는 이름으로 한 가지 작업을 호출합니다. 그러나 `aws s3 sync` 같은 사용자 지정 명령은 여러 API를 호출합니다. `--debug` 옵션으로 명령이 어떤 API를 호출하는지 확인할 수 있습니다.

사용자 또는 역할에 정책이 할당한 적절한 권한이 있는 경우 필요한 보안 인증 정보를 AWS CLI 명령에서 사용 중인지 확인합니다. 필요한 보안 인증 정보를 AWS CLI에서 사용 중인지 확인하려면 보안 인증 정보에 대한 다음 섹션을 참조하세요.

IAM 권한 할당에 대한 자세한 내용은 IAM 사용 설명서에서 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: AWS CLI가 잘못된 보안 인증 정보 또는 예상과 다른 위치의 보안 인증 정보를 읽습니다.

AWS CLI가 예상과 다른 위치에서 보안 인증 정보를 읽고 있거나 키 페어 정보가 올바르지 않을 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 "구성 설정" 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 date를 실행하여 시간을 확인합니다.

```
$ date
```

몇 분 안에 시스템 클록이 정확하지 않으면 ntpd를 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.
```

AWS CLI에서 명령을 실행하면 암호화된 요청을 AWS 서버에 보내 적절한 AWS 서비스 작업을 수행합니다. 보안 인증(액세스 키 및 보안 키)이 암호화에 포함되며 보안 인증을 통해 AWS에서 요청하는 사람을 인증할 수 있습니다. 다음과 같이 이 프로세스의 올바른 작업에 방해가 될 수 있는 요소가 여러 개 있습니다.

가능한 원인: 클록이 AWS 서버와 동기화되지 않음

[재생 공격(Replay Attack)](https://wikipedia.org/wiki/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 있음

AWS 키에 `-`, `+`, `/` 또는 `%`와 같은 특정 특수 문자가 포함되어 있는 경우 일부 운영 체제 변형에서 해당 문자열을 잘못 처리해서 키 문자열이 잘못 해석됩니다.

보안 인증 파일 생성 중 새 인스턴스에 보안 인증 파일을 작성하는 도구와 같이 다른 도구 또는 스크립트를 사용하여 키를 처리하는 경우, 이러한 도구 및 스크립트를 통해 특수 문자가 AWS에서 더 이상 인식할 수 없는 것으로 변환되도록 특수 문자를 자체적으로 처리할 수도 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. 이것은 일반적으로 설치한 Python `prompt_toolkit`이 오래된 경우 AWS CLI 버전 1에서 표시되는 오류입니다. 이 문제를 해결하려면 Python 웹 사이트에서 `prompt_toolkit` 최신 버전을 설치하세요.

맨 위로 이동

SSL 인증서 오류

가능한 원인: AWS CLI가 프록시 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 메시지를 받게 됩니다. 이것은 사용자의 프록시 인증서가 자체 서명되고 회사가 인증 기관(CA)으로 설정된 것과 같은 요인으로 인해 AWS CLI가 프록시의 인증서를 신뢰하지 않기 때문에 발생합니다. 이렇게 하면 AWS CLI에서 로컬 CA 레지스트리의 회사 CA 루트 인증서를 찾지 못하게 됩니다.

이 문제를 해결하려면 `ca_bundle` 구성 파일 설정, --ca-bundle 명령줄 옵션 또는 `AWS_CA_BUNDLE` 환경 변수를 사용해 회사의 `.pem` 파일을 찾을 수 있는 위치를 AWS CLI에 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 AWS 리전를 사용하지 않음

오류 텍스트 예:

```
$ aws s3 ls
[SSL: CERTIFICATE_ VERIFY_FAILED] certificate verify failed
```

지정한 AWS 리전에서 AWS 서비스를 사용할 수 없거나 리소스가 다른 AWS 리전에 있는 경우 오류 또는 예기치 않은 결과가 발생할 수 있습니다. 문제 해결 단계는 the section called “AWS CLI 명령이 사용 중인 AWS 리전 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 서비스가 디바이스의 TLS 버전과 호환되지 않는 TLS 버전을 사용합니다. 이 문제를 해결하려면 지원되는 TLS 버전으로 업데이트하세요. 자세한 내용은 the section called “최소 TLS 버전 적용” 섹션을 참조하세요.

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 메시지를 받게 됩니다. 이것은 일반적으로 사용자가 예상 JSON 형식으로 명령을 입력했지만 AWS CLI가 JSON을 올바르게 읽을 수 없는 경우 표시되는 오류입니다.

가능한 원인: AWS CLI에서 사용할 유효한 JSON을 입력하지 않음

명령에 유효한 JSON을 입력했는지 확인합니다. 형식 지정에 문제가 있는 JSON에 대해 JSON 검사기를 사용하는 것이 좋습니다.

명령줄에서 고급 JSON을 사용하려면 명령줄 JSON 프로세서(예: jq)를 사용하여 JSON 문자열을 생성하는 것이 좋습니다. jq에 대한 자세한 내용은 GitHub에서 jq 리포지토리를 참조하세요.

맨 위로 이동

가능한 원인: 터미널의 인용 규칙으로 인해 유효한 JSON이 AWS CLI로 전송되지 않음

AWS CLI가 명령에서 무엇이든 수신하기 전에 터미널은 자체 인용 및 이스케이프 규칙을 사용하여 명령을 처리합니다. 터미널의 형식 지정 규칙으로 인해 명령이 AWS CLI에 전달되기 전에 일부 JSON 콘텐츠가 제거될 수 있습니다. 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 JSON이 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 인용 규칙을 사용하여 JSON 입력이 AWS CLI로 전송될 때 발생하는 문제를 해결합니다. 인용 규칙에 대한 자세한 내용은 the section called "문자열과 따옴표" 섹션을 참조하세요.

> **Note**
>
> AWS CLI에 유효한 JSON을 가져오는 데 문제가 있는 경우 Blob을 사용하여 JSON 데이터를 AWS CLI에 직접 전달함으로써 JSON 데이터 입력에 대한 터미널의 인용 규칙을 우회하는 것이 좋습니다. Blob에 대한 자세한 내용은 Blob 섹션을 참조하세요.

맨 위로 이동

추가 리소스

AWS CLI 문제에 추가적인 도움이 필요하면 GitHub의 AWS CLI 커뮤니티 또는 AWS re:Post 커뮤니티를 방문하세요.

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 Foundation 에 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 섹션	멀티 팩터 인증 및 역할을 사용하여 CLI에 액세스하는 방법을 설명하는 새 섹션이 추가되었습니다.	2019년 5월 3일
"CLI 사용" 섹션 업데이트	CLI 사용 지침 및 절차에 대한 주요 개선 사항 및 추가 사항입니다.	2019년 3월 7일
"CLI 설치" 섹션 업데이트	AWS CLI 설치 지침 및 절차에 대한 주요 개선 사항 및 추가 사항입니다.	2019년 3월 7일
"CLI 구성" 섹션 업데이트	AWS CLI 구성 지침 및 절차에 대한 주요 개선 사항 및 추가 사항입니다.	2019년 3월 7일