aws

개발자 안내서

AWS Serverless Application Repository

AWS Serverless Application Repository: 개발자 안내서

Table of Contents

란 무엇입니까 AWS Serverless Application Repository?

를 AWS Serverless Application Repository 사용하면 개발자와 엔터프라이즈가 AWS 클라우드에서 서버리스 애플리케이션을 빠르게 찾고, 배포하고, 게시할 수 있습니다. 서버리스 애플리케이션에 대한 자세한 내용은 AWS 웹 사이트의 서버리스 컴퓨팅 및 애플리케이션을 참조하세요.

손쉽게 애플리케이션을 게시하여 대규모 커뮤니티와 공개적으로 또는 팀이나 조직에서 비공개로 공유할 수 있습니다. 서버리스 애플리케이션(또는 앱)을 게시하려면 AWS Management Console, AWS SAM 명령줄 인터페이스(AWS SAM CLI) 또는 AWS SDKs 사용하여 코드를 업로드할 수 있습니다. 코드와 함께 AWS Serverless Application Model (AWS SAM) 템플릿이라고도 하는 간단한 매니페스트 파일을 업로드합니다. 에 대한 자세한 내용은 AWS Serverless Application Model 개발자 안내서를 AWS SAM참조하세요.

AWS Serverless Application Repository 는 AWS Lambda 콘솔과 깊이 통합됩니다. 이러한 통합을 통해 모든 수준의 개발자가 새로운 무언가를 학습할 필요 없이 서버리스 컴퓨팅을 시작할 수 있습니다. 범주 키워드를 사용하여 웹 및 모바일 백엔드, 데이터 처리 애플리케이션 또는 챗봇과 같은 애플리케이션을 찾을 수 있습니다. 또한 이름, 게시자 또는 이벤트 소스별로 애플리케이션을 검색할 수 있습니다. 애플리케이션을 사용하려면 이를 선택하고 모든 필수 필드를 구성하고 클릭 몇 번만으로 배포하면 됩니다.

이 안내서에서 AWS Serverless Application Repository로 작업하는 두 가지 방법을 학습할 수 있습니다.

- 애플리케이션 게시 - 다른 개발자가 사용할 수 있도록 애플리케이션을 구성 및 업로드하고 새 버전의 애플리케이션을 게시합니다.
- 애플리케이션 배포 - 애플리케이션을 찾아 소스 코드 및 Readme 파일을 포함한 애플리케이션에 대한 정보를 봅니다. 또한 원하는 애플리케이션을 설치, 구성 및 배포합니다.

다음 단계

- 샘플 애플리케이션을에 게시하는 방법에 대한 자습서는 섹션을 AWS Serverless Application Repository참조하세요빠른 시작: 애플리케이션 게시.
- 에서 애플리케이션을 배포하는 방법에 대한 지침은 섹션을 AWS Serverless Application Repository 참조하세요애플리케이션을 배포하는 방법.

빠른 시작: 애플리케이션 게시

이 안내서에서는 AWS SAM CLI를 AWS Serverless Application Repository 사용하여 예제 서버리스 애플리케이션을 다운로드, 빌드, 테스트 및에 게시하는 단계를 안내합니다. 이 예제 애플리케이션을 기초로 삼아 자체 서버리스 애플리케이션을 개발하고 게시할 수 있습니다.

개요

다음 단계에서는 샘플 서버리스 애플리케이션을 다운로드, 빌드 및 게시하는 방법을 간략하게 설명합니다.

1. 초기화. `sam init`을 사용하여 템플릿에서 샘플 애플리케이션을 다운로드합니다.
2. 로컬에서 테스트. `sam local invoke` 및/또는 `sam local start-api`를 사용하여 로컬에서 애플리케이션을 테스트합니다. 이러한 명령을 사용하면 Lambda 함수가 로컬에서 호출되더라도 AWS 클라우드의 AWS 리소스에서 읽고 씁니다.
3. 패키지. Lambda 함수에 만족하면를 사용하여 Lambda 함수, AWS SAM 템플릿 및 모든 종속성을 AWS CloudFormation 배포 패키지에 번들링합니다`sam package`. 이 단계에서는 AWS Serverless Application Repository에 업로드할 애플리케이션에 대한 정보도 포함시킵니다.
4. 게시. `sam publish`를 사용하여 AWS Serverless Application Repository 에 애플리케이션을 게시합니다. 이 단계를 마치면에서 애플리케이션을 보고를 사용하여 AWS 클라우드에 AWS Serverless Application Repository 배포할 수 있습니다 AWS Serverless Application Repository.

다음 단원의 Hello World 애플리케이션 예제에서는 서버리스 애플리케이션을 빌드하고 게시하는 이러한 단계를 안내합니다.

Hello World 애플리케이션

이 연습에서는 간단한 API 백엔드를 나타내는 Hello World 서버리스 애플리케이션을 다운로드하고 테스트합니다. GET 작업과 Lambda 함수를 지원하는 Amazon API Gateway 엔드포인트가 있습니다. GET 요청이 엔드포인트로 전송되면 API Gateway는 Lambda 함수를 호출합니다. 그런 다음 함수를 AWS Lambda 실행하여 메시지를 반환합니다`hello world`.

이 애플리케이션은 다음과 같은 구성 요소로 이루어집니다.

- Hello World 애플리케이션의 두 AWS 리소스, 즉 GET 작업이 있는 API Gateway 서비스와 Lambda 함수를 정의하는 AWS SAM 템플릿입니다. 템플릿은 API Gateway GET 작업과 Lambda 함수 간의 매핑도 정의합니다.
- Python으로 작성된 애플리케이션 코드

시작하기 전

이 연습에 필요한 설정을 완료했는지 확인합니다.

- 관리자 권한이 있는 IAM 사용자가 있는 AWS 계정이 있어야 합니다. 계정 설정을 참조하세요 AWS.
- AWS SAM CLI(명령줄 인터페이스)가 설치되어 있어야 합니다. AWS SAM CLI 설치를 참조하세요.
- 버전 1.16.77 이상의가 AWS CLI 설치되어 있어야 합니다. AWS Command Line Interface설치를 참조하십시오.

1단계: 애플리케이션 초기화

이 단원에서는 AWS SAM 템플릿과 애플리케이션 코드로 구성된 샘플 애플리케이션을 다운로드합니다.

애플리케이션의 초기화

1. AWS SAM CLI 명령 프롬프트에서 다음 명령을 실행합니다.

   ```
   sam init --runtime python3.6
   ```

2. 명령을 통해 생성된 디렉터리(`sam-app/`)의 내용을 검토합니다.

   - `template.yaml` - Hello World 애플리케이션에 필요한 두 가지 AWS 리소스, 즉 Lambda 함수와 GET 작업을 지원하는 API Gateway 엔드포인트를 정의합니다. 또한 템플릿은 두 리소스 간의 매핑을 정의합니다.
   - Hello World 애플리케이션 코드와 관련된 내용:
     - `hello_world/` 디렉터리 - 애플리케이션 코드를 포함하며, 실행 `hello world` 시가 반환합니다.

> **Note**
>
> 이 연습에서는 애플리케이션 코드가 Python으로 작성되고 `init` 명령에서 런타임을 지정합니다.는 애플리케이션 코드 생성을 위한 추가 언어를 AWS Lambda 지원합니다. 지원되는 다른 런타임을 지정할 경우 `init` 명령이 지정된 언어로 작성된 Hello World 코드와 해당 언어에 따라 참조할 수 있는 `README.md` 파일을 제공합니다. 지원되는 런타임에 대한 자세한 내용은 Lambda 실행 환경 및 사용 가능한 라이브러리를 참조하십시오.

2단계: 로컬로 애플리케이션 테스트

이제 로컬 시스템에 AWS SAM 애플리케이션이 있으므로 아래 단계에 따라 로컬에서 애플리케이션을 테스트합니다.

로컬로 애플리케이션을 테스트하려면

1. API Gateway 엔드포인트를 로컬로 시작합니다. `template.yaml` 파일이 들어 있는 디렉터리에서 다음 명령을 실행해야 합니다.

   ```
   sam-app> sam local start-api --region us-east-1
   ```

   명령은 로컬 테스트를 위해 요청을 보낼 수 있는 API Gateway 엔드포인트를 반환합니다.
2. 애플리케이션을 테스트합니다. API Gateway 엔드포인트 URL을 복사하여 브라우저에 붙여 넣고 Enter를 선택합니다. 예제 API Gateway 엔드포인트 URL은 입니다`http://127.0.0.1:3000/hello`.

   API Gateway는 엔드포인트가 매핑되는 Lambda 함수를 로컬로 호출합니다. Lambda 함수는 로컬 Docker 컨테이너에서 실행되고를 반환합니다`hello world`. API Gateway는 텍스트가 포함된 응답을 브라우저에 반환합니다.

연습: 메시지 문자열 변경

샘플 애플리케이션을 성공적으로 테스트한 후에는 반환되는 메시지 문자열을 변경하는 간단한 수정 작업을 해볼 수 있습니다.

1. `/hello_world/app.py` 파일을 편집하여 메시지 문자열을 `'hello world'`에서 `'Hello World!'`로 변경합니다.
2. 브라우저에서 테스트 URL을 다시 로드하여 새 문자열을 확인합니다.

`sam local` 프로세스를 다시 시작하지 않고도 새 코드가 동적으로 로드되는 것을 볼 수 있습니다.

3단계: 애플리케이션 패키징

애플리케이션을 로컬에서 테스트한 후 AWS SAM CLI를 사용하여 배포 패키지와 패키지 AWS SAM 템플릿을 생성합니다.

> **Note**
>
> 다음 단계에서는 애플리케이션 코드가 포함된 `hello_world/` 디렉터리 내용을 .zip 파일로 만듭니다. 이 .zip 파일은 서버리스 애플리케이션의 배포 패키지입니다. 자세한 내용은 AWS Lambda 개발자 안내서의 배포 패키지 생성(Python)을 참조하세요.

Lambda 배포 패키지 생성하기

1. 필요한 애플리케이션 정보를 제공하는 `Metadata` 섹션을 AWS SAM 템플릿 파일에 추가합니다. AWS SAM 템플릿 `Metadata` 섹션에 대한 자세한 내용은 AWS Serverless Application Model 개발자 안내서의 AWS SAM 템플릿 Metdata 섹션 속성을 참조하세요.

   다음은 예제 `Metadata` 섹션입니다.

```
Metadata:
  AWS::ServerlessRepo::Application:
    Name: my-app
    Description: hello world
    Author: user1
    SpdxLicenseId: Apache-2.0
    LicenseUrl: LICENSE.txt
    ReadmeUrl: README.md
    Labels: ['tests']
    HomePageUrl: https://github.com/user1/my-app-project
    SemanticVersion: 0.0.1
    SourceCodeUrl: https://github.com/user1/my-app-project
```

`LicenseUrl` 및 `ReadmeUrl` 속성은 로컬 파일에 대한 참조이거나(위 예제 참조) 이미 이러한 아티팩트를 호스팅하는 Amazon S3 버킷에 대한 링크일 수 있습니다.

2. 패키징된 코드를 저장할 위치에 S3 버킷을 생성합니다. 기존 S3 버킷을 사용하려면 이 단계를 건너뜁니다.

```
sam-app> aws s3 mb s3://bucketname
```

3. 다음 `package` AWS SAM CLI 명령을 실행하여 Lambda 함수 배포 패키지를 생성합니다.

```
sam-app> sam package \
    --template-file template.yaml \
    --output-template-file packaged.yaml \
    --s3-bucket bucketname
```

명령은 다음 작업을 수행합니다.

- `aws-sam/hello_world/` 디렉터리의 내용을 압축하여 Amazon S3에 업로드합니다.
- 배포 패키지, README 파일 및 LICENSE 파일을 `--s3-bucket` 옵션에 지정된 Amazon S3 버킷에 업로드합니다.
- 새로운 템플릿 파일 `packaged.yaml`을 출력합니다. 이 파일은 다음 단계에서 애플리케이션을 AWS Serverless Application Repository에 게시하는 데 사용됩니다. `packaged.yaml` 템플릿 파일은 원래 템플릿 파일(`template.yaml`)과 유사하지만 키 차이가 있습니다. `CodeUri`, `LicenseUrl`및 `ReadmeUrl` 속성은 Amazon S3 버킷과 해당 아티팩트가 포함된 객체를 가리킵니다. 예제 `packaged.yaml` 템플릿 파일의 다음 코드 조각은 `CodeUri` 속성을 보여 줍니다.

```
HelloWorldFunction:
    Type: AWS::Serverless::Function # For more information about function
 resources, see https://github.com/awslabs/serverless-application-model/blob/
master/versions/2016-10-31.md#awsserverlessfunction
    Properties:
      CodeUri: s3://bucketname/fbd77a3647a4f47a352fc0bjectGUID

...
```

4단계: 애플리케이션 게시

이제 배포 패키지를 생성했으므로 이 패키지를 사용하여 AWS Serverless Application Repository에 애플리케이션을 게시합니다.

서버리스 애플리케이션을에 게시하려면 AWS Serverless Application Repository

- 다음 명령을 실행하여 0.0.1이라는 최초 버전으로 새 애플리케이션을 AWS Serverless Application Repository 에 게시합니다.

```
sam-app> sam publish \
    --template packaged.yaml \
    --region us-east-1
```

> **Note**
>
> 애플리케이션은 기본적으로 비공개로 생성됩니다. 다른 AWS 계정이 애플리케이션을 보고 배포하도록 허용하려면 먼저 애플리케이션을 공유해야 합니다. 애플리케이션 공유에 대한 자세한 내용은 아래에 나온 다음 단계를 참조하십시오.

다음 단계

이제 샘플 애플리케이션을 게시했으므로 애플리케이션에 대한 다음과 같은 작업을 수행할 수 있습니다.

- 에서 애플리케이션 보기 AWS Serverless Application Repository - `sam publish` 명령의 출력에는 애플리케이션의 세부 정보 페이지에 대한 AWS Serverless Application Repository 직접 링크가 포함됩니다. AWS Serverless Application Repository 랜딩 페이지로 이동하여 애플리케이션을 검색할 수도 있습니다.
- 애플리케이션 공유 - 애플리케이션은 기본적으로 프라이빗으로 설정되어 있으므로 다른 AWS 계정에 표시되지 않습니다. 애플리케이션을 다른 사용자와 공유하려면 애플리케이션을 공개하거나 특정 AWS 계정 목록에 권한을 부여해야 합니다. 를 사용하여 애플리케이션을 공유하는 방법에 대한 자세한 내용은 섹션을 AWS CLI 참조하세요AWS Serverless Application Repository 애플리케이션 정책 예제. 콘솔을 사용한 애플리케이션 공유에 대한 자세한 내용은 애플리케이션 공유 단원을 참조하십시오.

추가 정보

AWS SAM 템플릿의 `Metadata` 섹션`sam package`과 AWS SAM CLI의 `sam publish` 명령에 대한 자세한 내용은 AWS Serverless Application Model 개발자 안내서의 AWS SAM CLI를 사용하여 애플리케이션 게시를 참조하세요.

애플리케이션 게시

서버리스 애플리케이션을에 게시 AWS Serverless Application Repository할 때 다른 사용자가 찾아 배포할 수 있도록 합니다.

먼저 AWS Serverless Application Model (AWS SAM) 템플릿을 사용하여 애플리케이션을 정의합니다. 애플리케이션을 정의할 때는 애플리케이션의 소비자가 애플리케이션의 기능을 승인해야 하는지 여부를 고려해야 합니다. 기능 사용 AWS SAM 및 승인에 대한 자세한 내용은 섹션을 참조하세요와 AWS SAM 함께 사용 AWS Serverless Application Repository.

AWS Management Console, AWS SAM 명령줄 인터페이스(AWS SAM CLI) 또는 AWS SDK를 사용하여 서버리스 애플리케이션을 게시할 수 있습니다. 에 애플리케이션을 게시하는 절차에 대한 자세한 내용은 섹션을 AWS Serverless Application Repository참조하세요애플리케이션을 게시하는 방법.

애플리케이션을 게시할 때 애플리케이션이 처음에 프라이빗으로 설정되므로 애플리케이션을 생성한 AWS 계정에서만 사용할 수 있습니다. 애플리케이션을 다른 사용자와 공유하려면 애플리케이션을 비공개로 공유(특정 AWS 계정 집합과만 공유)하거나 공개적으로 공유(모든 사용자와 공유)하도록 설정해야 합니다.

애플리케이션을에 게시 AWS Serverless Application Repository 하고 퍼블릭으로 설정하면 서비스는 모든 리전의 소비자가 애플리케이션을 사용할 수 있도록 합니다. 소비자가 애플리케이션이 처음 게시된 리전 이외의 리전에 퍼블릭 애플리케이션을 배포하면는 애플리케이션의 배포 아티팩트를 대상 리전의 Amazon S3 버킷에 AWS Serverless Application Repository 복사합니다. 대신 대상 리전의 Amazon S3 버킷에 있는 파일을 참조하도록 해당 아티팩트를 사용하는 AWS SAM 템플릿의 모든 리소스를 업데이트합니다. 배포 아티팩트에는 Lambda 함수 코드, API 정의 파일 등이 포함될 수 있습니다.

> **Note**
>
> 프라이빗 및 프라이빗 공유 애플리케이션은 생성된 AWS 리전에서만 사용할 수 있습니다. 공개적으로 공유된 애플리케이션은 모든 AWS 리전에서 사용할 수 있습니다. 애플리케이션 공유에 대한 자세한 내용은 AWS Serverless Application Repository 애플리케이션 정책 예제 단원을 참조하십시오.

주제

- 검증된 작성자 배지
- Lambda 계층 공유

와 AWS SAM 함께 사용 AWS Serverless Application Repository

AWS Serverless Application Model (AWS SAM)는에서 서버리스 애플리케이션을 구축하는 데 사용할 수 있는 오픈 소스 프레임워크입니다 AWS. 를 사용하여 서버리스 애플리케이션을 빌드 AWS SAM 하는 방법에 대한 자세한 내용은 AWS Serverless Application Model 개발자 안내서를 참조하세요.

에 게시할 애플리케이션을 빌드할 때는 사용할 수 있는 지원되는 AWS 리소스 및 정책 템플릿 세트를 고려해야 AWS Serverless Application Repository합니다. 아래 단원에서 이러한 주제에 대해 자세히 설명합니다.

에서 지원되는 AWS 리소스 AWS Serverless Application Repository

는 많은 AWS SAM 및 AWS CloudFormation 리소스로 구성된 서버리스 애플리케이션을 AWS Serverless Application Repository 지원합니다. 에서 지원하는 AWS 리소스의 전체 목록을 보려면 섹션을 AWS Serverless Application Repository참조하세요지원되는 AWS 리소스 목록.

추가 AWS 리소스에 대한 지원을 요청하려면 AWS Support에 문의하세요.

Important

애플리케이션 템플릿에 다음 사용자 지정 IAM 역할이나 리소스 정책 중 하나가 포함되어 있을 경우 기본적으로는 검색 결과에 표시되지 않습니다. 또한 고객은 애플리케이션의 사용자 지정 IAM 역할이나 리소스 정책을 승인해야 애플리케이션을 배포할 수 있습니다. 자세한 정보는 애플리케이션 기능 승인을 참조하십시오.
이것이 적용되는 리소스 목록은 다음과 같습니다.

- IAM 역할: AWS::IAM::Group, AWS::IAM::InstanceProfile, AWS::IAM::Policy, and AWS::IAM::Role.
- 리소스 정책: AWS::Lambda::LayerVersionPermission, AWS::Lambda::Permission, AWS::Events::EventBusPolicy, AWS::IAM:Policy, AWS::ApplicationAutoScaling::ScalingPolicy, AWS::S3::BucketPolicy, AWS::SQS::QueuePolicy, and AWS::SNS:TopicPolicy.

애플리케이션에 AWS::Serverless::Application 리소스가 포함되어 있을 경우, 고객은 애플리케이션에 중첩 애플리케이션이 포함되어 있음을 승인해야 애플리케이션을 배포할 수 있습니다.

중첩 애플리케이션에 대한 자세한 내용은 AWS Serverless Application Model 개발자 안내서의 중첩 애플리케이션을 참조하세요. 기능 승인에 대한 자세한 내용은 애플리케이션 기능 승인을 참조하십시오.

정책 템플릿

AWS SAM 는 Lambda 함수의 권한을 애플리케이션에서 사용하는 리소스로 범위를 지정하는 정책 템플릿 목록을 제공합니다. 정책 템플릿을 사용하는 경우 애플리케이션을 검색, 탐색 또는 배포하기 위해 고객이 추가로 승인할 필요가 없습니다.

표준 AWS SAM 정책 템플릿 목록은 AWS Serverless Application Model 개발자 안내서의 AWS SAM 정책 템플릿을 참조하세요.

지원되는 AWS 리소스 목록

다음은에서 지원하는 AWS 리소스의 전체 목록입니다 AWS Serverless Application Repository.

- `AWS::AccessAnalyzer::Analyzer`
- `AWS::AmazonMQ::Broker`
- `AWS::AmazonMQ::Configuration`
- `AWS::AmazonMQ::ConfigurationAssociation`
- `AWS::ApiGateway::Account`
- `AWS::ApiGateway::ApiKey`
- `AWS::ApiGateway::Authorizer`
- `AWS::ApiGateway::BasePathMapping`
- `AWS::ApiGateway::ClientCertificate`
- `AWS::ApiGateway::Deployment`
- `AWS::ApiGateway::DocumentationPart`
- `AWS::ApiGateway::DocumentationVersion`
- `AWS::ApiGateway::DomainName`
- `AWS::ApiGateway::GatewayResponse`
- `AWS::ApiGateway::Method`
- `AWS::ApiGateway::Model`

- AWS::ApiGateway::RequestValidator
- AWS::ApiGateway::Resource
- AWS::ApiGateway::RestApi
- AWS::ApiGateway::Stage
- AWS::ApiGateway::UsagePlan
- AWS::ApiGateway::UsagePlanKey
- AWS::ApiGateway::VpcLink
- AWS::ApiGatewayV2::Api
- AWS::ApiGatewayV2::ApiMapping
- AWS::ApiGatewayV2::Authorizer
- AWS::ApiGatewayV2::DomainName
- AWS::ApiGatewayV2::Deployment
- AWS::ApiGatewayV2::Integration
- AWS::ApiGatewayV2::IntegrationResponse
- AWS::ApiGatewayV2::Model
- AWS::ApiGatewayV2::Route
- AWS::ApiGatewayV2::RouteResponse
- AWS::ApiGatewayV2::Stage
- AWS::AppSync::ApiKey
- AWS::AppSync::DataSource
- AWS::AppSync::GraphQLApi
- AWS::AppSync::GraphQLSchema
- AWS::AppSync::Resolver
- AWS::ApplicationAutoScaling::AutoScalingGroup
- AWS::ApplicationAutoScaling::LaunchConfiguration
- AWS::ApplicationAutoScaling::ScalableTarget
- AWS::ApplicationAutoScaling::ScalingPolicy
- AWS::Athena::NamedQuery
- AWS::Athena::WorkGroup

- AWS::CertificateManager::Certificate
- AWS::Chatbot::SlackChannelConfiguration
- AWS::CloudFormation::CustomResource
- AWS::CloudFormation::Interface
- AWS::CloudFormation::Macro
- AWS::CloudFormation::WaitConditionHandle
- AWS::CloudFront::CachePolicy
- AWS::CloudFront::CloudFrontOriginAccessIdentity
- AWS::CloudFront::Distribution
- AWS::CloudFront::Function
- AWS::CloudFront::OriginRequestPolicy
- AWS::CloudFront::ResponseHeadersPolicy
- AWS::CloudFront::StreamingDistribution
- AWS::CloudTrail::Trail
- AWS::CloudWatch::Alarm
- AWS::CloudWatch::AnomalyDetector
- AWS::CloudWatch::Dashboard
- AWS::CloudWatch::InsightRule
- AWS::CodeBuild::Project
- AWS::CodeCommit::Repository
- AWS::CodePipeline::CustomActionType
- AWS::CodePipeline::Pipeline
- AWS::CodePipeline::Webhook
- AWS::CodeStar::GitHubRepository
- AWS::CodeStarNotifications::NotificationRule
- AWS::Cognito::IdentityPool
- AWS::Cognito::IdentityPoolRoleAttachment
- AWS::Cognito::UserPool
- AWS::Cognito::UserPoolClient

- AWS::Cognito::UserPoolDomain
- AWS::Cognito::UserPoolGroup
- AWS::Cognito::UserPoolResourceServer
- AWS::Cognito::UserPoolUser
- AWS::Cognito::UserPoolUserToGroupAttachment
- AWS::Config::AggregationAuthorization
- AWS::Config::ConfigRule
- AWS::Config::ConfigurationAggregator
- AWS::Config::ConfigurationRecorder
- AWS::Config::DeliveryChannel
- AWS::Config::RemediationConfiguration
- AWS::DataPipeline::Pipeline
- AWS::DynamoDB::Table
- AWS::EC2::EIP
- AWS::EC2::InternetGateway
- AWS::EC2::NatGateway
- AWS::EC2::Route
- AWS::EC2::RouteTable
- AWS::EC2::SecurityGroup
- AWS::EC2::SecurityGroupEgress
- AWS::EC2::SecurityGroupIngress
- AWS::EC2::Subnet
- AWS::EC2::SubnetRouteTableAssociation
- AWS::EC2::VPC
- AWS::EC2::VPCGatewayAttachment
- AWS::EC2::VPCPeeringConnection
- AWS::ECR::Repository
- AWS::Elasticsearch::Domain
- AWS::Events::EventBus
- AWS::Events::EventBusPolicy

- AWS::Events::Rule
- AWS::EventSchemas::Discoverer
- AWS::EventSchemas::Registry
- AWS::EventSchemas::Schema
- AWS::Glue::Classifier
- AWS::Glue::Connection
- AWS::Glue::Crawler
- AWS::Glue::Database
- AWS::Glue::DevEndpoint
- AWS::Glue::Job
- AWS::Glue::Partition
- AWS::Glue::SecurityConfiguration
- AWS::Glue::Table
- AWS::Glue::Trigger
- AWS::Glue::Workflow
- AWS::IAM::Group
- AWS::IAM::InstanceProfile
- AWS::IAM::ManagedPolicy
- AWS::IAM::OIDCProvider
- AWS::IAM::Policy
- AWS::IAM::Role
- AWS::IAM::ServiceLinkedRole
- AWS::IoT::Certificate
- AWS::IoT::Policy
- AWS::IoT::PolicyPrincipalAttachment
- AWS::IoT::Thing
- AWS::IoT::ThingPrincipalAttachment
- AWS::IoT::TopicRule
- AWS::KMS::Alias
- AWS::KMS::Key

- AWS::Kinesis::Stream
- AWS::Kinesis::StreamConsumer
- AWS::Kinesis::Streams
- AWS::KinesisAnalytics::Application
- AWS::KinesisAnalytics::ApplicationOutput
- AWS::KinesisFirehose::DeliveryStream
- AWS::Lambda::Alias
- AWS::Lambda::EventInvokeConfig
- AWS::Lambda::EventSourceMapping
- AWS::Lambda::Function
- AWS::Lambda::LayerVersion
- AWS::Lambda::LayerVersionPermission
- AWS::Lambda::Permission
- AWS::Lambda::Version
- AWS::Location::GeofenceCollection
- AWS::Location::Map
- AWS::Location::PlaceIndex
- AWS::Location::RouteCalculator
- AWS::Location::Tracker
- AWS::Location::TrackerConsumer
- AWS::Logs::Destination
- AWS::Logs::LogGroup
- AWS::Logs::LogStream
- AWS::Logs::MetricFilter
- AWS::Logs::SubscriptionFilter
- AWS::Route53::HealthCheck
- AWS::Route53::HostedZone
- AWS::Route53::RecordSet
- AWS::Route53::RecordSetGroup
- AWS::S3::Bucket

- AWS::S3::BucketPolicy
- AWS::SNS::Subscription
- AWS::SNS::Topic
- AWS::SNS::TopicPolicy
- AWS::SQS::Queue
- AWS::SQS::QueuePolicy
- AWS::SSM::Association
- AWS::SSM::Document
- AWS::SSM::MaintenanceWindowTask
- AWS::SSM::Parameter
- AWS::SSM::PatchBaseline
- AWS::SSM::ResourceDataSync
- AWS::SecretsManager::ResourcePolicy
- AWS::SecretsManager::RotationSchedule
- AWS::SecretsManager::Secret
- AWS::SecretsManager::SecretTargetAttachment
- AWS::Serverless::Api
- AWS::Serverless::Application
- AWS::Serverless::Function
- AWS::Serverless::HttpApi
- AWS::Serverless::LayerVersion
- AWS::Serverless::SimpleTable
- AWS::Serverless::StateMachine
- AWS::ServiceDiscovery::HttpNamespace
- AWS::ServiceCatalog::CloudFormationProvisionedProduct
- AWS::ServiceDiscovery::Instance
- AWS::ServiceDiscovery::PrivateDnsNamespace
- AWS::ServiceDiscovery::PublicDnsNamespace
- AWS::ServiceDiscovery::Service
- AWS::SES::ReceiptRule

- `AWS::SES::ReceiptRuleSet`
- `AWS::StepFunctions::Activity`
- `AWS::StepFunctions::StateMachine`
- `AWS::Wisdom::Assistant`
- `AWS::Wisdom::AssistantAssociation`
- `AWS::Wisdom::KnowledgeBase`

애플리케이션을 게시하는 방법

이 섹션에서는 AWS SAM CLI 또는를 사용하여 서버리스 애플리케이션을 AWS Serverless Application Repository 에 게시하는 절차를 제공합니다 AWS Management Console. 또한 다른 사용자가 배포할 수 있도록 응용 프로그램을 공유하고 AWS Serverless Application Repository에서 애플리케이션을 삭제하는 방법을 보여 줍니다.

> **Important**
>
> 애플리케이션을 게시할 때 입력하는 정보는 암호화되지 않습니다. 이 정보에는 작성자 이름과 같은 데이터가 포함됩니다. 저장 또는 공개를 원치 않는 개인 식별 정보가 있는 경우 애플리케이션을 게시할 때 이 정보를 입력하지 않는 것이 좋습니다.

애플리케이션 게시(AWS CLI)

에 애플리케이션을 게시하는 가장 쉬운 방법은 AWS SAM CLI 명령 세트를 사용하는 AWS Serverless Application Repository 것입니다. 자세한 내용은 AWS Serverless Application Model (AWS SAM) 개발자 안내서의 AWS SAM CLI를 사용하여 애플리케이션 게시를 참조하세요.

새 애플리케이션 게시(콘솔)

이 섹션에서는를 사용하여 AWS Management Console 에 새 애플리케이션을 게시하는 방법을 보여줍니다 AWS Serverless Application Repository. 기존 애플리케이션의 새 버전을 게시하는 지침은 기존 애플리케이션의 새 버전 게시 단원을 참조하십시오.

사전 조건

에 애플리케이션을 게시하기 전에 다음이 AWS Serverless Application Repository필요합니다.

- 유효한 AWS 계정입니다.
- 사용되는 AWS 리소스를 정의하는 valid AWS Serverless Application Model (AWS SAM) 템플릿입니다. AWS SAM 템플릿에 대한 자세한 내용은 AWS SAM 템플릿 기본 사항을 참조하세요.
- 에 대한 `package` 명령을 사용하여 AWS CloudFormation 생성한 애플리케이션의 패키지입니다 AWS CLI. 이 명령은 AWS SAM 템플릿이 참조하는 로컬 아티팩트(로컬 경로)를 패키징합니다. 자세한 내용은 AWS CloudFormation 설명서의 패키지를 참조하세요.
- 애플리케이션을 공개적으로 게시하고자 하는 경우 애플리케이션의 소스 코드를 가리키는 URL.
- readme.txt 파일. 이 파일은 고객이 애플리케이션을 사용하는 방법과 자신의 AWS 계정에 배포하기 전에 구성하는 방법을 설명해야 합니다.
- SPDX 웹 사이트에 있는 license.txt 파일 또는 유효한 라이선스 식별자. 애플리케이션을 공개적으로 공유하려는 경우에만 라이선스가 필요합니다. 애플리케이션을 비공개로 유지하거나 비공개로만 공유하려는 경우, 라이선스를 지정할 필요가 없습니다.
- 애플리케이션을 패키징할 때 Amazon S3에 업로드된 아티팩트에 대한 읽기 권한을 서비스에 부여하는 유효한 Amazon S3 버킷 정책입니다. 이 정책을 설정하려면 다음 단계를 수행합니다.

  1. https://console.aws.amazon.com/s3/에서 Amazon S3 콘솔을 엽니다.
  2. 애플리케이션을 패키징하는 데 사용한 Amazon S3 버킷을 선택합니다.
  3. 권한 탭을 선택합니다.
  4. 버킷 정책 버튼을 선택합니다.
  5. 다음 정책 문을 버킷 정책 편집기에 붙여 넣습니다. 요소의 버킷 이름과 `Resource` 요소의 AWS 계정 ID를 대체해야 합니다`Condition`. `Condition` 요소의 표현식은에 지정된 AWS 계정의 애플리케이션에 액세스할 수 있는 권한 AWS Serverless Application Repository 만 있는지 확인합니다. 요소에 대한 자세한 내용은 IAM 사용 설명서의 IAM JSON 정책 요소 참조를 참조하세요.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service":  "serverlessrepo.amazonaws.com"
            },
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::bucketname/*",
            "Condition" : {
                "StringEquals": {
```

```
                    "aws:SourceAccount": "123456789012"
                }
            }
        }
    ]
}
```

6. 저장 버튼을 선택합니다.

절차

다음 절차에 AWS Serverless Application Repository 따라에서 새 애플리케이션을 생성합니다.

에서 새 애플리케이션을 생성하려면 AWS Serverless Application Repository

1. AWS Serverless Application Repository 콘솔을 열고 Publish applications(애플리케이션 게시)를 선택합니다.
2. Publish an application(애플리케이션 게시) 페이지에서 다음 애플리케이션 정보를 입력하고 Publish application(애플리케이션 게시)를 선택합니다.

속성	필수	설명
애플리케이션 이름	TRUE	애플리케이션의 이름입니다. 최소 길이: 1. 최대 길이: 140. 패턴: "[a-zA-Z0-9\\-]+";
작성자	TRUE	애플리케이션을 게시하는 작성자의 이름입니다. 최소 길이: 1. 최대 길이: 127. 패턴: "^[a-z0-9](([a-z0-9]|-(?!-))*[a-z0-9])?$";
홈 페이지	FALSE	애플리케이션에 대한 자세한 정보가 들어 있는 URL(예: 애플리케이션의 GitHub 리포지토리 위치)

속성	필수	설명
설명	TRUE	애플리케이션에 대한 설명입니다. 최소 길이: 1. 최대 길이: 256.
레이블	FALSE	검색 결과에서 애플리케이션의 검색을 향상시키는 레이블입니다. 최소 길이: 1. 최대 길이: 127. 최대 레이블 수: 10. 패턴: "^[a-zA-Z0-9+\\-_:\\/@]+$";
Spdx 라이선스(드롭다운 목록)	FALSE	SPDX 웹 사이트에서 사용할 수 있는 라이선스의 드롭다운 목록에서 유효한 라이선스 식별자를 선택합니다. 드롭다운 목록에서 항목을 선택하면 그 아래에 있는 License(라이선스) 텍스트 상자가 채워집니다. 참고: 드롭다운 목록에서 라이선스를 선택하면 License(라이선스) 텍스트 상자의 내용이 바뀌고 수동으로 편집한 내용은 취소됩니다.

속성	필수	설명
라이선스	FALSE	.txt 라이선스 파일을 업로드하거나, 앞 행에 설명된 Spdx license(Spdx 라이선스) 드롭다운 목록에서 라이선스를 선택합니다. Spdx license(Spdx 라이선스) 드롭다운 목록에서 라이선스를 선택하면 License(라이선스) 텍스트 상자가 자동으로 채워집니다. 라이선스 파일을 업로드하거나 Spdx license(Spdx 라이선스) 드롭다운 목록에서 선택한 후 이 텍스트 상자의 내용을 수동으로 편집할 수 있습니다. 그러나 드롭다운 목록에서 다른 Spdx 라이선스를 선택한 경우 수동으로 편집한 내용이 모두 무시됩니다. 선택적 필드이지만 애플리케이션을 공개적으로 공유하려면 라이선스를 제공해야 합니다.
Readme	FALSE	Readme 파일(텍스트 또는 마크다운 형식)의 내용을 업로드합니다. AWS Serverless Application Repository의 애플리케이션 세부 정보 페이지에 해당 내용이 표시됩니다. 파일을 업로드한 후 이 텍스트 상자의 내용을 수동으로 편집할 수 있습니다.

속성	필수	설명
의미 체계 버전	FALSE	애플리케이션의 의미 체계 버전입니다. 자세한 내용은 의미 체계 버전 관리 웹 사이트를 참조하십시오. 애플리케이션을 공개로 설정하려면 이 속성의 값을 지정해야 합니다.
소스 코드 URL	FALSE	애플리케이션의 소스 코드가 있는 퍼블릭 리포지토리의 링크입니다.
SAM 템플릿	TRUE	사용되는 AWS 리소스를 정의하는 valid AWS Serverless Application Model (AWS SAM) 템플릿입니다.

애플리케이션 공유

게시된 애플리케이션에 다음 세 범주 중 하나의 사용 권한을 설정할 수 있습니다.

- 프라이빗(기본값) - 동일한 계정으로 생성되었으며 다른 AWS 계정과 공유되지 않은 애플리케이션입니다. AWS 계정을 공유하는 소비자만 프라이빗 애플리케이션을 배포할 수 있는 권한이 있습니다.
- 비공개 공유 - 게시자가 특정 AWS 계정 집합 또는 AWS 조직의 AWS 계정과 명시적으로 공유한 애플리케이션입니다. 소비자는 AWS 계정 또는 AWS 조직과 공유된 애플리케이션을 배포할 수 있는 권한이 있습니다. 에 대한 자세한 내용은 AWS Organizations 사용 설명서를 AWS Organizations참조하세요.
- 공개 공유 - 게시자가 모든 사람과 공유한 애플리케이션입니다. 모든 소비자는 공개적 공유 애플리케이션을 배포할 수 있는 권한이 있습니다.

애플리케이션을에 게시한 후 AWS Serverless Application Repository기본적으로 프라이빗으로 설정됩니다. 이 섹션에서는 애플리케이션을 특정 AWS 계정 또는 AWS 조직과 비공개로 공유하거나 모든 사람과 공개적으로 공유하는 방법을 보여줍니다.

콘솔을 통해 애플리케이션 공유

애플리케이션을 다른 사용자와 공유하기 위한 두 가지 옵션이 있습니다. 1) 특정 AWS 계정 또는 AWS 조직 내 AWS 계정과 공유하거나 2) 모든 사용자와 공개적으로 공유합니다. 에 대한 자세한 내용은 AWS Organizations 사용 설명서를 AWS Organizations참조하세요.

옵션 1: 특정 AWS 계정(들) 또는 AWS 조직 내 계정과 애플리케이션을 공유하려면

1. AWS Serverless Application Repository 콘솔을 엽니다.
2. 탐색 창에서 Published Applications(게시된 애플리케이션)를 선택하면 생성한 애플리케이션 목록이 표시됩니다.
3. 공유하고자 하는 애플리케이션을 선택합니다.
4. 공유 탭을 선택합니다.
5. Application policy statements(애플리케이션 정책 문) 섹션에서 Create Statement(문 생성) 버튼을 선택합니다.
6. Statement Configuration(문 구성) 창에서 원하는 애플리케이션 공유 방법에 따라 필드에 값을 입력합니다.

   > **Note**
   >
   > 조직과 공유하는 경우 AWS 계정이 속한 조직만 지정할 수 있습니다. 사용자가 멤버가 아닌 AWS 조직을 지정하려고 하면 오류가 발생합니다.
   > 애플리케이션을 AWS 조직과 공유하려면 나중에 공유를 취소해야 하는 경우를 대비하여 `UnshareApplication` 작업이 정책 설명에 추가될 것임을 확인해야 합니다.

7. 저장 버튼을 선택합니다.

옵션 2: 애플리케이션을 모든 사람과 공개적으로 공유하려면

1. AWS Serverless Application Repository 콘솔을 엽니다.
2. 탐색 창에서 Published Applications(게시된 애플리케이션)를 선택하면 생성한 애플리케이션 목록이 표시됩니다.
3. 공유하고자 하는 애플리케이션을 선택합니다.
4. 공유 탭을 선택합니다.
5. Public Sharing(공개 공유) 섹션에서 편집 버튼을 선택합니다.
6. Public Sharing(공개 공유) 에서 사용 라디오 버튼을 선택합니다.

7. 텍스트 상자에 애플리케이션의 이름을 입력한 다음 저장 버튼을 선택합니다.

> **Note**
>
> 애플리케이션을 공개적으로 공유하려면 `SemanticVersion` 및 `LicenseUrl` 속성이 둘 다 설정되어 있어야 합니다.

AWS CLI를 통해 애플리케이션 공유

를 사용하여 애플리케이션을 공유하려면 `put-application-policy` 명령을 사용하여 공유하려는 AWS 계정(들)을 보안 주체로 지정하는 권한을 AWS CLI 부여합니다.

AWS CLI를 사용하여 애플리케이션을 공유하는 방법에 대한 자세한 내용은 섹션을 참조하세요AWS Serverless Application Repository 애플리케이션 정책 예제.

애플리케이션 공유 해제

AWS 조직에서 애플리케이션을 공유 해제하는 방법에는 두 가지가 있습니다.

1. 애플리케이션 게시자는 `put-application-policy` 명령을 사용하여 권한을 제거할 수 있습니다.
2. AWS 조직의 관리 계정의 사용자는 다른 계정의 사용자가 애플리케이션을 게시한 경우에도 조직과 공유된 모든 애플리케이션에서 공유 해제 애플리케이션 작업을 수행할 수 있습니다.

   > **Note**
   >
   > "애플리케이션 공유 해제" 작업을 사용하여 AWS 조직에서 애플리케이션을 공유 해제하면 AWS Organization과 다시 공유할 수 없습니다.

   에 대한 자세한 내용은 AWS Organizations 사용 설명서를 AWS Organizations참조하세요.

게시자가 권한 제거

콘솔을 통해 게시자가 권한 제거

를 통해 애플리케이션을 공유 해제하려면 다른 AWS 계정과 공유하는 정책 설명을 AWS Management Console제거합니다. 이렇게 하려면 다음 단계를 따릅니다.

1. AWS Serverless Application Repository 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 Available Applications(사용 가능한 애플리케이션)를 선택합니다.
3. 공유 해제할 애플리케이션을 선택합니다.
4. 공유 탭을 선택합니다.
5. Application policy statements(애플리케이션 정책 문) 섹션에서 공유를 해제하려는 계정과 그 애플리케이션을 공유하는 정책 문을 선택합니다.
6. Delete(삭제)를 선택합니다.
7. 확인 메시지가 나타납니다. 다시 삭제를 선택합니다.

를 통해 권한 제거 게시자 AWS CLI

를 통해 애플리케이션을 공유 해제하기 위해 AWS CLI게시자는 `put-application-policy` 명령을 사용하여 애플리케이션을 비공개로 설정하거나 다른 AWS 계정 집합과 공유할 수 있는 권한을 제거하거나 변경할 수 있습니다.

AWS CLI를 사용하여 권한을 변경하는 방법에 대한 자세한 내용은 섹션을 참조하세요AWS Serverless Application Repository 애플리케이션 정책 예제.

애플리케이션 공유 해제 관리 계정

콘솔을 통해 AWS 조직에서 애플리케이션 공유 해제하는 관리 계정

를 통해 AWS 조직에서 애플리케이션을 공유 해제하려면 관리 계정의 AWS Management Console사용자가 다음을 수행할 수 있습니다.

1. AWS Serverless Application Repository 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 Available Applications(사용 가능한 애플리케이션)를 선택합니다.
3. 애플리케이션 타일에서 Unshare(공유 해제)를 선택합니다.
4. 공유 해제 메시지 상자에서 조직 ID와 애플리케이션 이름을 입력한 다음 저장을 선택하여 애플리케이션의 공유 해제를 확인합니다.

를 통해 AWS 조직에서 애플리케이션 공유 해제하는 관리 계정 AWS CLI

AWS 조직에서 애플리케이션을 공유 해제하려면 관리 계정의 사용자가 `aws serverlessrepo unshare-application` 명령을 실행할 수 있습니다.

다음 명령은 AWS 조직에서 애플리케이션을 공유 해제합니다. 여기서 *application-id*는 애플리케이션의 Amazon 리소스 이름(ARN)이고 *organization-id*는 조직 ID입니다. AWS

```
aws serverlessrepo unshare-application --application-id application-id --organization-id organization-id
```

애플리케이션 삭제

AWS Management Console 또는 CLI를 AWS Serverless Application Repository 사용하여에서 애플리케이션을 삭제할 수 있습니다 AWS SAM .

애플리케이션 삭제(콘솔)

를 통해 게시된 애플리케이션을 삭제하려면 다음을 AWS Management Console수행합니다.

1. AWS Serverless Application Repository 콘솔을 엽니다.
2. 내 애플리케이션에서 삭제하고자 하는 애플리케이션을 선택합니다.
3. 애플리케이션의 세부 정보 페이지에서 애플리케이션 삭제를 선택합니다.
4. 애플리케이션 삭제를 선택하여 삭제를 완료합니다.

애플리케이션 삭제(AWS CLI)

를 사용하여 게시된 애플리케이션을 삭제하려면 `aws serverlessrepo delete-application` 명령을 AWS CLI실행합니다.

다음 명령은 애플리케이션을 삭제합니다. 여기서 *application-id*는 애플리케이션의 ARN(Amazon 리소스 이름)입니다.

```
aws serverlessrepo delete-application --application-id application-id
```

기존 애플리케이션의 새 버전 게시

이 섹션에서는 AWS SAM CLI 또는를 AWS Serverless Application Repository 사용하여 기존 애플리케이션의 새 버전을에 게시하는 방법을 보여줍니다 AWS Management Console. 새 애플리케이션을 게시하는 지침은 애플리케이션을 게시하는 방법 단원을 참조하십시오.

기존 애플리케이션의 새 버전 게시(AWS CLI)

기존 애플리케이션의 새 버전을 게시하는 가장 쉬운 방법은 AWS SAM CLI 명령 세트를 사용하는 것입니다. 자세한 내용은 AWS Serverless Application Model (AWS SAM) 개발자 안내서의 AWS SAM CLI를 사용하여 애플리케이션 게시를 참조하세요.

기존 애플리케이션의 새 버전 게시(콘솔)

이전에 게시한 애플리케이션의 새 버전을 게시하려면 다음 단계를 수행합니다.

1. AWS Serverless Application Repository 콘솔을 엽니다.
2. 탐색 창에서 My Applications(내 애플리케이션)를 선택하면 생성한 애플리케이션 목록이 표시됩니다.
3. 새 버전을 게시하고자 하는 애플리케이션을 선택합니다.
4. [새 버전 발행]을 선택합니다.
5. Versions(버전)에서 다음 애플리케이션 정보를 입력합니다.

속성	필수	설명
의미 체계 버전	TRUE	애플리케이션의 의미 체계 버전입니다. 자세한 내용은 의미 체계 버전 관리 웹 사이트를 참조하십시오. 애플리케이션을 공개로 설정하려면 이 속성의 값을 지정해야 합니다.
소스 코드 URL	FALSE	애플리케이션의 소스 코드가 있는 퍼블릭 리포지토리의 링크입니다.
SAM 템플릿	TRUE	사용되는 AWS 리소스를 정의하는 valid AWS Serverless Application Model (AWS SAM) 템플릿입니다.

6. Publish version(버전 게시)을 선택합니다.

검증된 작성자 배지

의 검증된 작성자 AWS Serverless Application Repository 는 합리적이고 신중 AWS 한 서비스 공급자로서 요청자가 제공한 정보를 선의로 검토하고 요청자의 자격 증명이 청구된 것과 같다고 확인한 작성자입니다.

검증된 작성자의 애플리케이션에는 작성자의 공개 프로필 링크와 함께 검증된 작성자 배지가 표시됩니다. 검증된 작성자 배지는 검색 결과와 애플리케이션 세부 정보 페이지에 모두 표시됩니다.

검증된 작성자 배지 신청

serverlessrepo-verified-author@amazon.com으로 이메일을 AWS Serverless Application Repository 보내에서 확인된 작성자로 승인을 요청할 수 있습니다. 다음 정보를 제공해야 합니다.

- 작성자 이름
- AWS 계정 ID
- GitHub 또는 LinkedIn 프로필과 같은 공개적으로 액세스 가능한 프로필 링크

확인된 작성자 배지에 대한 요청을 제출한 후 며칠 AWS 이내에에서 응답을 받을 수 있습니다. 신청이 승인되기 전에 추가 정보를 제공하도록 요청받을 수 있습니다.

신청이 승인되면 하루 이내에 신청에 대한 검증된 작성자 배지가 표시됩니다.

> **Note**
>
> 확인된 작성자 배지는 AWS 계정 및 작성자 이름과 모두 일치하는 모든 애플리케이션에 표시됩니다. AWS 계정에는 여러 작성자가 있을 수 있으므로 작성자 이름이 다른 애플리케이션에는 배지가 표시되지 않습니다. 작성자 이름이 다른 애플리케이션에 작성자 배지를 표시하려면 해당 작성자에 대해 별도의 신청을 제출해야 합니다.

Lambda 계층 공유

Lambda 계층에서 기능을 구현한 경우 글로벌 인스턴스를 호스팅하지 않고 계층을 공유할 수 있습니다. 이러한 방식으로 계층을 공유하면 다른 사용자가 계층의 인스턴스를 자신의 계정에 배포할 수 있습니다. 따라서 클라이언트 애플리케이션이 계층의 전역 인스턴스에 의존하지 않게 됩니다. 를 AWS Serverless Application Repository 사용하면 이러한 방식으로 Lambda 계층을 쉽게 공유할 수 있습니다.

Lambda 계층에 대한 자세한 내용은 AWS Lambda 개발자 안내서의 AWS Lambda 계층을 참조하세요.

작동 방식

다음은 AWS Serverless Application Repository를 사용하여 계층을 공유하는 단계입니다. 이렇게 하면 사용자 AWS 계정에 계층의 복사본을 생성할 수 있습니다.

1. 계층을 리소스로 포함하는 AWS SAM 템플릿, 즉 `AWS::Serverless::LayerVersion` 또는 `AWS::Lambda::LayerVersion` 리소스를 사용하여 서버리스 애플리케이션을 정의합니다.
2. 애플리케이션을에 게시 AWS Serverless Application Repository하고 공유(공개 또는 비공개)합니다.
3. 고객이 애플리케이션을 배포하여 자신의 AWS 계정에 계층의 복사본을 생성합니다. 이제 고객은 클라이언트 애플리케이션에서 AWS 계정에 있는 계층의 Amazon 리소스 이름(ARN)을 참조할 수 있습니다.

예제

다음은 공유하려는 Lambda 계층이 포함된 애플리케이션의 예제 AWS SAM 템플릿입니다.

```
Resources:
  SharedLayer:
    Type: AWS::Serverless::LayerVersion
    Properties:
      LayerName: shared-layer
      ContentUri: source/layer-code/
      CompatibleRuntimes:
        - python3.7
Outputs:
  LayerArn:
    Value: !Ref SharedLayer
```

고객이에서 애플리케이션을 배포하면 AWS 계정에 AWS Serverless Application Repository계층이 생성됩니다. 계층의 ARN은 다음과 같습니다.

`arn:aws:lambda:us-east-1:012345678901:layer:shared-layer:1`

이제 고객은 다음 예와 같이 자체 클라이언트 애플리케이션에서 이 ARN을 참조할 수 있습니다.

```
Resources:
```

```
  MyFunction:
    Type: AWS::Serverless::Function
    Properties:
      Handler: index.handler
      Runtime: python3.7
      CodeUrl: source/app-code/
      Layers:
        - arn:aws:lambda:us-east-1:012345678901:layer:shared-layer:1
```

애플리케이션 배포

이 단원에서는 AWS Serverless Application Repository에 게시된 서버리스 애플리케이션을 찾고 배포하는 방법을 알아볼 수 있습니다. 퍼블릭 사이트를 방문하여 AWS 계정이 없어도 공개적으로 사용할 수 있는 애플리케이션을 찾을 수 있습니다. 또는 AWS Lambda 콘솔 내에서 애플리케이션을 찾을 수 있습니다.

일부 애플리케이션에는 작성자 프로필 링크와 함께 검증된 작성자 배지가 있습니다. AWS 가 합리적이고 신중한 서비스 공급자로서 요청자가 제공한 정보를 성실하게 검토하고 요청자의 자격 증명이 청구된 대로임을 확인한 경우 작성자는 확인된 작성자로 간주됩니다.

에서 애플리케이션을 배포하기 전에 다음 주제를 AWS Serverless Application Repository참조하여 애플리케이션 배포 권한 및 애플리케이션 기능에 대해 알아봅니다.

주제

- 애플리케이션 배포 권한
- 애플리케이션 기능: IAM 역할, 리소스 정책 및 중첩 애플리케이션
- 애플리케이션을 배포하는 방법

애플리케이션 배포 권한

에 애플리케이션을 배포하려면 그렇게 할 권한이 AWS Serverless Application Repository있어야 합니다. 배포할 수 있는 권한이 있는 애플리케이션에는 다음과 같은 3가지 범주가 있습니다.

- 프라이빗 - 동일한 계정으로 생성되었으며 다른 계정과 공유되지 않은 애플리케이션입니다. 계정을 사용하여 생성된 애플리케이션을 배포할 수 있는 권한이 있습니다 AWS .
- 비공개 공유 - 게시자가 특정 AWS 계정 집합과 명시적으로 공유한 애플리케이션입니다. 계정과 공유된 애플리케이션을 배포할 수 있는 권한이 있습니다 AWS .
- 공개 공유 - 게시자가 모든 사람과 공유한 애플리케이션입니다. 소비자는 공개적 공유 애플리케이션을 배포할 수 있는 권한이 있습니다.

권한이 있는 애플리케이션만 검색하고 찾아볼 수 있습니다. 여기에는 계정을 사용하여 생성되고, AWS 계정과 비공개로 공유되고 AWS , 공개적으로 공유된 애플리케이션이 포함됩니다. 다른 모든 애플리케이션은 표시되지 않습니다.

> **Important**
>
> 중첩 애플리케이션이 포함된 애플리케이션은 중첩 애플리케이션의 공유 제한을 상속합니다. 예를 들어 애플리케이션이 공개적으로 공유되지만 상위 애플리케이션을 생성한 AWS 계정과 비공개로만 공유되는 중첩 애플리케이션이 포함되어 있다고 가정해 보겠습니다. 이 경우 AWS 계정에 중첩된 애플리케이션을 배포할 권한이 없는 경우 상위 애플리케이션을 배포할 수 없습니다. 중첩 애플리케이션에 대한 자세한 내용은 AWS Serverless Application Model 개발자 안내서의 중첩 애플리케이션을 참조하세요.

애플리케이션 기능: IAM 역할, 리소스 정책 및 중첩 애플리케이션

애플리케이션을 배포하기 전에는 애플리케이션의 템플릿에서 템플릿이 생성하도록 지정하는 IAM 역할, AWS 리소스 정책 및 중첩된 애플리케이션을 AWS Serverless Application Repository 확인합니다. 전체 액세스 권한이 있는 IAM 역할과 같은 IAM 리소스는 AWS 계정의 모든 리소스를 수정할 수 있습니다. 따라서 에스컬레이션된 권한을 가진 리소스를 실수로 생성하는 일이 없도록 계속하기 전에 각 애플리케이션에 연결된 권한을 검토하는 것이 좋습니다. 이렇게 하려면가 사용자를 대신하여 애플리케이션을 AWS Serverless Application Repository 배포하기 전에 애플리케이션에 기능이 포함되어 있음을 확인해야 합니다.

애플리케이션에는 `CAPABILITY_IAM`, `CAPABILITY_NAMED_IAM`, `CAPABILITY_RESOURCE_POLICY`, `CAPABILITY_AUTO_EXPAND` 등의 기능이 포함될 수 있습니다.

다음 리소스에는 `CAPABILITY_IAM` 또는 `CAPABILITY_NAMED_IAM`을 지정해야 합니다. AWS::IAM::Group, AWS::IAM::InstanceProfile, AWS::IAM::Policy, and AWS::IAM::Role. 애플리케이션에 사용자 지정 이름을 가진 IAM 리소스가 포함되어 있는 경우 `CAPABILITY_NAMED_IAM`을 지정해야 합니다. 기능을 지정하는 방법에 대한 예제는 애플리케이션 기능 확인 및 승인(AWS CLI) 단원을 참조하십시오.

다음 리소스에는 `CAPABILITY_RESOURCE_POLICY`를 지정해야 합니다. AWS::Lambda::LayerVersionPermission, AWS::Lambda::Permission, AWS::Events::EventBusPolicy, AWS::IAM:Policy, AWS::ApplicationAutoScaling::ScalingPolicy, AWS::S3::BucketPolicy, AWS::SQS::QueuePolicy, and AWS::SNS::TopicPolicy.

중첩 애플리케이션을 한 개 이상 포함하는 애플리케이션은 `CAPABILITY_AUTO_EXPAND`를 지정해야 합니다. 중첩 애플리케이션에 대한 자세한 내용은 AWS Serverless Application Model 개발자 안내서의 중첩 애플리케이션을 참조하세요.

애플리케이션 기능 확인 및 승인(콘솔)

AWS Serverless Application Repository AWS Serverless Application Repository 웹 사이트의 또는 Lambda 콘솔(탭 아래의 함수 생성 페이지)에서 사용 가능한 애플리케이션을 찾을 수 있습니다 AWS Serverless Application Repository.

사용자 지정 IAM 역할이나 리소스 정책을 생성하기 위해 기능을 승인해야 하는 애플리케이션은 기본적으로 검색 결과에 표시되지 않습니다. 이러한 기능을 포함하는 애플리케이션을 검색하려면 Show apps that create custom IAM roles or resource policies(사용자 지정 IAM 역할 또는 리소스 정책을 생성하는 앱 표시) 확인란을 선택해야 합니다.

애플리케이션을 선택할 때 권한 탭에서 애플리케이션 기능을 검토할 수 있습니다. 애플리케이션을 배포하려면 I acknowledge this application creates custom IAM roles or resource polices(이 애플리케이션이 사용자 지정 IAM 역할이나 리소스 정책을 생성함을 승인함) 확인란을 선택해야 합니다. 이러한 기능을 승인하지 않으면 승인 필요 오류 메시지가 표시됩니다. 배포하려면 애플리케이션 파라미터 구성 섹션에서 확인란을 선택합니다.

애플리케이션 기능 보기(AWS CLI)

를 사용하여 애플리케이션의 기능을 보려면 AWS CLI먼저 애플리케이션의 Amazon 리소스 이름(ARN)이 필요합니다. 그리고 다음 명령을 실행할 수 있습니다.

```
aws serverlessrepo get-application \
--application-id application-arn
```

requiredCapabilities 응답 속성에는 애플리케이션을 배포하려면 승인해야 하는 애플리케이션 기능 목록이 들어 있습니다. requiredCapabilities 속성이 비어 있으면 애플리케이션에 필요한 기능이 없습니다.

애플리케이션을 배포하는 방법

이 섹션에서는 AWS Management Console 또는를 AWS Serverless Application Repository 사용하여에서 서버리스 애플리케이션을 배포하는 절차를 제공합니다 AWS CLI.

새 애플리케이션 배포(콘솔)

이 섹션에서는를 AWS Serverless Application Repository 사용하여에서 새 애플리케이션을 배포하는 방법을 보여줍니다 AWS Management Console. 기존 애플리케이션의 새 버전을 배포하는 지침은 애플리케이션 업데이트 단원을 참조하십시오.

애플리케이션 찾기, 검색 및 배포

다음 절차를 AWS Serverless Application Repository 사용하여에서 애플리케이션을 찾고 구성하고 배포합니다.

에서 애플리케이션을 찾고 구성하려면 AWS Serverless Application Repository

1. AWS Serverless Application Repository 공개 홈 페이지를 열거나 AWS Lambda 콘솔을 엽니다. 함수 생성을 선택한 다음 Browse serverless app repository(서버리스 앱 리포지토리 찾아보기)를 선택합니다.
2. 애플리케이션을 찾거나 검색합니다.

   > **Note**
   >
   > 사용자 지정 IAM 역할이나 리소스 정책을 포함하는 애플리케이션을 표시하려면 Show apps that create custom IAM roles or resource policies(사용자 지정 IAM 역할 또는 리소스 정책을 생성하는 앱 표시) 확인란을 선택합니다. 사용자 지정 IAM 역할 및 리소스 정책에 대한 자세한 정보는 애플리케이션 기능 승인을 참조하십시오.

3. 애플리케이션을 선택하여 권한, 기능 및 AWS 고객이 배포한 횟수와 같은 세부 정보를 봅니다.

   배포 수는 애플리케이션을 배포하려는 AWS 리전에 대해 표시됩니다.
4. 애플리케이션 세부 정보 페이지에서 AWS SAM 템플릿, 라이선스 및 readme 파일을 확인하여 애플리케이션의 권한 및 애플리케이션 리소스를 확인합니다. 이 페이지에서 공개적으로 공유된 애플리케이션에 대한 소스 코드 URL을 찾을 수도 있습니다. 애플리케이션에 중첩 애플리케이션이 포함되어 있으면 이 페이지에서 중첩 애플리케이션의 세부 정보를 볼 수도 있습니다.
5. 애플리케이션 설정 섹션에서 애플리케이션을 구성합니다. 특정 애플리케이션 구성에 대한 지침은 애플리케이션의 readme 파일을 참조하십시오.

   예를 들어 구성 요구 사항에는 애플리케이션에서 액세스하고자 하는 리소스의 이름 지정이 포함될 수 있습니다. 이러한 리소스는 Amazon DynamoDB 테이블, Amazon S3 버킷 또는 Amazon API Gateway API일 수 있습니다.
6. 배포(Deploy)를 선택합니다. 그러면 [Deployment status] 페이지로 이동하게 됩니다.

   > **Note**
   >
   > 애플리케이션에 승인이 필요한 기능이 있는 경우 애플리케이션을 배포하기 전에 I acknowledge this application creates custom IAM roles or resource polices(이 애플리케

이션이 사용자 지정 IAM 역할이나 리소스 정책을 생성함을 승인함) 확인란을 선택해야 합니다. 이렇게 하지 않으면 오류가 발생합니다. 사용자 지정 IAM 역할 및 리소스 정책에 대한 자세한 정보는 애플리케이션 기능 승인을 참조하십시오.

7. 배포 상태 페이지에서 배포의 진행 상황을 볼 수 있습니다. 배포가 완료될 때까지 기다리는 동안 다른 애플리케이션을 검색하고 찾아 Lambda 콘솔을 통해이 페이지로 돌아갈 수 있습니다.

애플리케이션이 성공적으로 배포되면 기존 AWS 도구를 사용하여 생성된 리소스를 검토하고 관리할 수 있습니다.

새 애플리케이션 배포(AWS CLI)

이 섹션에서는를 AWS Serverless Application Repository 사용하여에서 새 애플리케이션을 배포하는 방법을 보여줍니다 AWS CLI. 기존 애플리케이션의 새 버전을 배포하는 지침은 애플리케이션 업데이트 단원을 참조하십시오.

애플리케이션 기능 확인 및 승인(AWS CLI)

를 사용하여 애플리케이션의 기능을 승인하려면 다음 단계를 AWS CLI수행합니다.

1. 애플리케이션의 기능을 검토합니다. 다음 AWS CLI 명령을 사용하여 애플리케이션의 기능을 검토합니다.

   ```
   aws serverlessrepo get-application \
   --application-id application-arn
   ```

   requiredCapabilities 응답 속성에는 애플리케이션을 배포하려면 승인해야 하는 애플리케이션 기능 목록이 들어 있습니다. AWS SDKs에서 GetApplication API를 사용하여이 데이터를 가져올 수도 있습니다.

2. 변경 세트를 생성합니다. AWS CloudFormation 변경 세트를 생성할 때 필요한 기능 세트를 제공해야 합니다. 예를 들어 다음 AWS CLI 명령을 사용하여 기능을 확인하여 애플리케이션을 배포합니다.

   ```
   aws serverlessrepo create-cloud-formation-change-set \
   --application-id application-arn \
   --stack-name unique-name-for-cloud-formation-stack \
   --capabilities list-of-capabilities
   ```

이 명령이 성공적으로 실행되면 변경 세트 ID가 반환됩니다. 다음 단계를 위해 변경 세트 ID가 필요합니다. 또한 AWS SDKs에서 CreateCloudFormationChangeSet API를 사용하여 변경 세트를 생성할 수 있습니다.

예를 들어 다음 AWS CLI 명령은 사용자 지정 이름과 하나 이상의 중첩된 애플리케이션이 있는 AWS::IAM::Role 리소스가 포함된 애플리케이션을 확인합니다.

```
aws serverlessrepo create-cloud-formation-change-set \
--application-id application-arn \
--stack-name unique-name-for-cloud-formation-stack \
--capabilities CAPABILITY_NAMED_IAM CAPABILITY_AUTO_EXPAND
```

3. 변경 세트를 실행합니다. 변경 세트를 실행하면 실제로 배포가 수행됩니다. 이전 단계에서 변경 세트를 생성할 때 반환된 변경 세트 ID를 입력합니다.

   다음 예제 AWS CLI 명령은 애플리케이션 변경 세트를 실행하여 애플리케이션을 배포합니다.

   ```
   aws cloudformation execute-change-set \
   --change-set-name changeset-id-arn
   ```

   또한 AWS SDKs의 ExecuteChangeSet API를 사용하여 변경 세트를 실행할 수 있습니다.

애플리케이션 스택 삭제

를 사용하여 이전에 배포한 애플리케이션을 삭제하려면 AWS CloudFormation 스택 삭제와 동일한 절차를 AWS Serverless Application Repository따릅니다.

- AWS Management Console:를 사용하여 애플리케이션을 삭제하려면 사용 설명서의 AWS CloudFormation 콘솔에서 스택 삭제를 AWS Management Console참조하세요. AWS CloudFormation
- AWS CLI:를 사용하여 애플리케이션을 삭제하려면 사용 설명서의 스택 삭제를 AWS CLI참조하세요. AWS CloudFormation

애플리케이션 업데이트

에서 애플리케이션을 배포한 후 업데이트할 AWS Serverless Application Repository수 있습니다. 예를 들어 애플리케이션 설정을 변경하거나 애플리케이션을 게시된 최신 버전으로 업데이트할 수 있습니다.

다음 섹션에서는 AWS Management Console 또는를 사용하여 새 버전의 애플리케이션을 배포하는 방법을 설명합니다 AWS CLI.

애플리케이션 업데이트(콘솔)

이전에 배포한 애플리케이션을 업데이트하려면 새 애플리케이션을 배포하는 것과 동일한 절차를 사용하고 원래 배포에 사용한 것과 동일한 애플리케이션 이름을 지정합니다. 특히는 애플리케이션 이름 `serverlessrepo-` 앞에 AWS Serverless Application Repository 표시됩니다. 새 버전의 애플리케이션을 배포하려면 앞에 `serverlessrepo-`를 추가하지 않고 원래 애플리케이션 이름을 지정해야 합니다.

예를 들어, 이름이 `MyApplication`인 애플리케이션을 배포한 경우 스택 이름은 `serverlessrepo-MyApplication`이 됩니다. 해당 애플리케이션을 업데이트하려면 이름을 `MyApplication` 다시 입력합니다. 전체 스택 이름을 로 지정하지 마십시오`serverlessrepo-MyApplication`.

다른 모든 애플리케이션 설정의 경우 값을 이전 배포와 동일하게 유지하거나 새 값을 지정할 수 있습니다.

애플리케이션 업데이트(AWS CLI)

이전에 배포한 애플리케이션을 업데이트하려면 새 애플리케이션을 배포하는 것과 동일한 절차를 사용하고 원래 배포에 사용한 것과 동일한 *`--stack-name`*을 지정합니다. 특히 AWS Serverless Application Repository 는 스택 이름 `serverlessrepo-` 앞에 옵니다. 새 버전의 애플리케이션을 배포하려면 앞에 `serverlessrepo-`를 추가하지 않고 원래 스택 이름을 지정해야 합니다.

예를 들어, 스택 이름이 `MyApplication`인 애플리케이션을 배포한 경우 생성되는 스택 이름은 `serverlessrepo-MyApplication`이 됩니다. 해당 애플리케이션을 업데이트하려면 이름을 `MyApplication` 다시 입력합니다.의 전체 스택 이름을 지정하지 마십시오`serverlessrepo-MyApplication`.

의 보안 AWS Serverless Application Repository

의 클라우드 보안 AWS 이 최우선 순위입니다. AWS 고객은 보안에 가장 민감한 조직의 요구 사항을 충족하도록 구축된 데이터 센터 및 네트워크 아키텍처의 이점을 누릴 수 있습니다.

보안은 AWS 와 사용자 간의 공동 책임입니다. 공동 책임 모델은 이 사항을 클라우드 내 보안 및 클라우드의 보안으로 설명합니다.

- 클라우드 보안 - AWS 는 클라우드에서 AWS AWS 서비스를 실행하는 인프라를 보호할 책임이 있습니다. AWS 또한는 안전하게 사용할 수 있는 서비스를 제공합니다. 서드 파티 감사원은 정기적으로 AWS 규정 준수 프로그램의 일환으로 보안 효과를 테스트하고 검증합니다. AWS Serverless Application Repository에 적용되는 규정 준수 프로그램에 대해 알아보려면 규정 준수 프로그램 제공 범위 내AWS 서비스를 참조하십시오.
- 클라우드의 보안 - 사용자의 책임은 사용하는 AWS 서비스에 따라 결정됩니다. 또한 귀하는 귀사의 데이터 민감도, 귀사의 요구 사항, 관련 법률 및 규정을 비롯한 기타 요소에 대해서도 책임이 있습니다.

이 설명서는 AWS Serverless Application Repository를 사용할 때 공동 책임 모델을 적용하는 방법을 이해하는 데 도움이 됩니다. 다음 주제에서는 보안 및 규정 준수 목표에 맞게 AWS Serverless Application Repository 를 구성하는 방법을 보여줍니다. 또한 AWS Serverless Application Repository 리소스를 모니터링하고 보호하는 데 도움이 되는 다른 AWS 서비스를 사용하는 방법도 알아봅니다.

주제

의 데이터 보호 AWS Serverless Application Repository

AWS 공동 책임 모델의 데이터 보호에 적용됩니다 AWS Serverless Application Repository. 이 모델에 설명된 대로 AWS 는 모든를 실행하는 글로벌 인프라를 보호할 책임이 있습니다 AWS 클라우드. 사용

자는 인프라에서 호스팅되는 콘텐츠를 관리해야 합니다. 사용하는 AWS 서비스 의 보안 구성과 관리 태스크에 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.

데이터 보호를 위해 자격 증명을 보호하고 AWS 계정 AWS IAM Identity Center 또는 AWS Identity and Access Management (IAM)를 사용하여 개별 사용자를 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 다중 인증(MFA)을 사용하세요.
- SSL/TLS를 사용하여 AWS 리소스와 통신합니다. TLS 1.2는 필수이며 TLS 1.3을 권장합니다.
- 를 사용하여 API 및 사용자 활동 로깅을 설정합니다 AWS CloudTrail. CloudTrail 추적을 사용하여 AWS 활동을 캡처하는 방법에 대한 자세한 내용은 AWS CloudTrail 사용 설명서의 CloudTrail 추적 작업을 참조하세요.
- AWS 암호화 솔루션과 내부의 모든 기본 보안 제어를 사용합니다 AWS 서비스.
- Amazon S3에 저장된 민감한 데이터를 검색하고 보호하는 데 도움이 되는 Amazon Macie와 같은 고급 관리형 보안 서비스를 사용하세요.
- 명령줄 인터페이스 또는 API를 AWS 통해에 액세스할 때 FIPS 140-3 검증 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 형식 텍스트 필드에 입력하지 않는 것이 좋습니다. 여기에는 콘솔, API AWS CLI또는 AWS SDKs를 사용하여 AWS Serverless Application Repository 또는 다른 AWS 서비스 로 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 자격 증명을 URL에 포함해서는 안 됩니다.

전송 중 데이터 암호화

AWS Serverless Application Repository API 엔드포인트는 HTTPS를 통한 보안 연결만 지원합니다. AWS Management Console, AWS SDK 또는 AWS Serverless Application Repository API를 사용하여 리소스를 관리 AWS Serverless Application Repository 하면 모든 통신이 TLS(전송 계층 보안)로 암호화됩니다.

API 엔드포인트의 전체 목록은의 AWS 리전 및 엔드포인트를 참조하세요AWS 일반 참조.

유휴 데이터 암호화

는 배포 패키지 및 계층 아카이브를 AWS Serverless Application Repository포함하여 AWS Serverless Application Repository 에 업로드하는 파일을 암호화합니다.

AWS Serverless Application Repository에 대한 Identity and Access Management

AWS Identity and Access Management (IAM)는 관리자가 AWS 리소스에 대한 액세스를 안전하게 제어할 수 AWS 서비스 있도록 도와주는 입니다. IAM 관리자는 AWS Serverless Application Repository 리소스를 사용할 수 있는 인증(로그인) 및 권한 부여(권한 있음)를 받을 수 있는 사용자를 제어합니다. IAM은 추가 비용 없이 사용할 수 AWS 서비스 있는 입니다.

IAM 작동 방식에 대한 개요를 보려면 IAM 사용 설명서의 IAM 작동 방식 이해를 참조하세요.

주제

- 대상
- 자격 증명을 사용하여 인증
- 정책을 사용하여 액세스 관리
- 가 IAM과 AWS Serverless Application Repository 작동하는 방식
- AWS Serverless Application Repository 자격 증명 기반 정책 예제
- AWS Serverless Application Repository 애플리케이션 정책 예제
- AWS Serverless Application Repository API 권한: 작업 및 리소스 참조
- AWS Serverless Application Repository 자격 증명 및 액세스 문제 해결

대상

사용 방법 AWS Identity and Access Management (IAM)은 수행하는 작업에 따라 다릅니다 AWS Serverless Application Repository.

서비스 사용자 - AWS Serverless Application Repository 서비스를 사용하여 작업을 수행하는 경우 필요한 자격 증명과 권한을 관리자가 제공합니다. 더 많은 AWS Serverless Application Repository 기능을 사용하여 작업을 수행하게 되면 추가 권한이 필요할 수 있습니다. 액세스 권한 관리 방법을 이해하면 관리자에게 올바른 권한을 요청하는 데 도움이 됩니다. AWS Serverless Application Repository의

기능에 액세스할 수 없는 경우 AWS Serverless Application Repository 자격 증명 및 액세스 문제 해결을 참조하세요.

서비스 관리자 - 회사에서 AWS Serverless Application Repository 리소스를 책임지고 있는 경우에 대한 전체 액세스 권한이 있을 수 있습니다 AWS Serverless Application Repository. 서비스 사용자가 액세스해야 하는 AWS Serverless Application Repository 기능과 리소스를 결정하는 것은 사용자의 작업입니다. 그런 다음 IAM 관리자에게 요청을 제출하여 서비스 사용자의 권한을 변경해야 합니다. 이 페이지의 정보를 검토하여 IAM의 기본 개념을 이해하세요. 회사가에서 IAM을 사용하는 방법에 대한 자세한 내용은 섹션을 AWS Serverless Application Repository참조하세요가 IAM과 AWS Serverless Application Repository 작동하는 방식.

IAM 관리자 - IAM 관리자라면 AWS Serverless Application Repository에 대한 액세스 권한 관리 정책 작성 방법을 자세히 알고 싶을 것입니다. IAM에서 사용할 수 있는 자격 AWS Serverless Application Repository 증명 기반 정책 예제를 보려면 섹션을 참조하세요AWS Serverless Application Repository 자격 증명 기반 정책 예제.

자격 증명을 사용하여 인증

인증은 AWS 자격 증명으로에 로그인하는 방법입니다. IAM 사용자 또는 AWS 계정 루트 사용자 IAM 역할을 수임하여 로 인증(로그인 AWS)되어야 합니다.

자격 증명 소스를 통해 제공된 자격 증명을 사용하여 페더레이션 자격 증명 AWS 으로에 로그인할 수 있습니다. AWS IAM Identity Center (IAM Identity Center) 사용자, 회사의 Single Sign-On 인증 및 Google 또는 Facebook 자격 증명은 페더레이션 자격 증명의 예입니다. 페더레이션형 ID로 로그인할 때 관리자가 이전에 IAM 역할을 사용하여 ID 페더레이션을 설정했습니다. 페더레이션을 사용하여 AWS 에 액세스하면 간접적으로 역할을 수임하게 됩니다.

사용자 유형에 따라 AWS Management Console 또는 AWS 액세스 포털에 로그인할 수 있습니다. 에 로그인하는 방법에 대한 자세한 내용은 AWS 로그인 사용 설명서의에 로그인하는 방법을 AWS참조하세요. AWS 계정

AWS 프로그래밍 방식으로에 액세스하는 경우는 자격 증명을 사용하여 요청에 암호화 방식으로 서명할 수 있는 소프트웨어 개발 키트(SDK)와 명령줄 인터페이스(CLI)를 AWS 제공합니다. AWS 도구를 사용하지 않는 경우 직접 요청에 서명해야 합니다. 권장 방법을 사용하여 요청에 직접 서명하는 자세한 방법은 IAM 사용 설명서에서 API 요청용AWS Signature Version 4를 참조하세요.

사용하는 인증 방법에 상관없이 추가 보안 정보를 제공해야 할 수도 있습니다. 예를 들어는 다중 인증(MFA)을 사용하여 계정의 보안을 강화할 것을 AWS 권장합니다. 자세한 내용은 AWS IAM Identity Center 사용 설명서에서 다중 인증 및 IAM 사용 설명서에서 IAM의AWS 다중 인증을 참조하세요.

AWS 계정 루트 사용자

를 생성할 때 계정의 모든 AWS 서비스 및 리소스에 대한 전체 액세스 권한이 있는 하나의 로그인 자격 증명으로 AWS 계정시작합니다. 이 자격 증명을 AWS 계정 테루트 사용자라고 하며 계정을 생성하는 데 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 자격 증명이 필요한 작업을 참조하세요.

IAM 사용자 및 그룹

IAM 사용자는 한 사람 또는 애플리케이션에 대한 특정 권한이 AWS 계정 있는 내 자격 증명입니다. 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 권한이 AWS 계정 있는 내 자격 증명입니다. IAM 사용자와 유사하지만, 특정 개인과 연결되지 않습니다. 에서 IAM 역할을 일시적으로 수임하려면 사용자에서 IAM 역할(콘솔)로 전환할 AWS Management Console수 있습니다. https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_use_switch-role-console.html 또는 AWS API 작업을 호출하거나 사용자 지정 URL을 AWS CLI 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. 페

더레이션 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 집합을 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. 그러나 일부 에서는 (역할을 프록시로 사용하는 대신) 정책을 리소스에 직접 연결할 AWS 서비스 수 있습니다. 교차 계정 액세스에 대한 역할과 리소스 기반 정책의 차이점을 알아보려면 IAM 사용 설명서의 IAM의 교차 계정 리소스 액세스를 참조하세요.
- 교차 서비스 액세스 - 일부는 다른에서 기능을 AWS 서비스 사용합니다 AWS 서비스. 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 사용하여에서 작업을 수행하는 경우 AWS보안 주체로 간주됩니다. 일부 서비스를 사용하는 경우, 다른 서비스에서 다른 작업을 시작하는 작업을 수행할 수 있습니다. FAS는를 호출하는 보안 주체의 권한을 다운스트림 서비스에 AWS 서비스 대한 요청과 AWS 서비스함께 사용합니다. FAS 요청은 서비스가 다른 AWS 서비스 또는 리소스와의 상호 작용을 완료해야 하는 요청을 수신할 때만 수행됩니다. 이 경우, 두 작업을 모두 수행할 수 있는 권한이 있어야 합니다. FAS 요청 시 정책 세부 정보는 전달 액세스 세션을 참조하세요.
  - 서비스 역할 - 서비스 역할은 서비스가 사용자를 대신하여 작업을 수행하기 위해 맡는 IAM 역할입니다. 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 서비스를 참조하세요.
  - 서비스 연결 역할 - 서비스 연결 역할은에 연결된 서비스 역할의 한 유형입니다 AWS 서비스. 서비스는 사용자를 대신하여 작업을 수행하기 위해 역할을 수임할 수 있습니다. 서비스 연결 역할은에 표시 AWS 계정 되며 서비스가 소유합니다. IAM 관리자는 서비스 링크 역할의 권한을 볼 수 있지만 편집은 할 수 없습니다.
- Amazon EC2에서 실행되는 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 AWS 역할을 할당하고 모든 애플리케이션에서 사용할 수 있도록 하려면 인스턴스에 연결된 인스턴스 프로파일을 생성합니다. 인스턴스 프로필에는 역할이 포함되어 있으며 EC2 인스턴스에서 실행되는 프로그

램이 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 AWS 생성하고 자격 증명 또는 리소스에 연결하여 AWS 에서 액세스를 제어합니다. 정책은 자격 증명 또는 리소스와 연결된 AWS 경우 권한을 정의하는의 객체입니다.는 보안 주체(사용자, 루트 사용자 또는 역할 세션)가 요청할 때 이러한 정책을 AWS 평가합니다. 정책에서 권한은 요청이 허용되거나 거부되는 지를 결정합니다. 대부분의 정책은에 JSON 문서 AWS 로 저장됩니다. JSON 정책 문서의 구조와 콘텐츠에 대한 자세한 정보는 IAM 사용 설명서의 JSON 정책 개요를 참조하세요.

관리자는 AWS JSON 정책을 사용하여 누가 무엇을 액세스할 수 있는지 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 API에서 역할 정보를 가져올 수 있습니다 AWS .

자격 증명 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 정책은의 여러 사용자, 그룹 및 역할에 연결할 수 있는 독립 실행형 정책입니다 AWS 계정. 관리형 정책에는 AWS 관리형 정책 및 고객 관리형 정책이 포함됩니다. 관리형 정책 또는 인라인 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 또는가 포함될 수 있습니다 AWS 서비스.

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. ACL은 JSON 정책 문서 형식을 사용하지 않지만 리소스 기반 정책과 유사합니다.

Amazon S3 AWS WAF및 Amazon VPC는 ACLs. ACL에 관한 자세한 내용은 Amazon Simple Storage Service 개발자 가이드의 액세스 제어 목록(ACL) 개요를 참조하세요.

기타 정책 유형

AWS 는 덜 일반적인 추가 정책 유형을 지원합니다. 이러한 정책 타입은 더 일반적인 정책 유형에 따라 사용자에게 부여되는 최대 권한을 설정할 수 있습니다.

- 권한 경계 – 권한 경계는 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(SCPs) - SCPs는의 조직 또는 조직 단위(OU)에 대한 최대 권한을 지정하는 JSON 정책입니다 AWS Organizations. AWS Organizations 는 비즈니스가 소유 AWS 계정 한 여러를 그룹화하고 중앙에서 관리하기 위한 서비스입니다. 조직에서 모든 기능을 활성화할 경우, 서비스 제어 정책(SCP)을 임의의 또는 모든 계정에 적용할 수 있습니다. SCP는 각각을 포함하여 멤버 계정의 엔터티에 대한 권한을 제한합니다 AWS 계정 루트 사용자. 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 AWS 계정 루트 사용자관계없이를 포함한 자격 증명에 대한 유효 권한에 영향을 미칠 수 있습니다. RCP를 AWS 서비스 지원하는 목록을 포함하여 조직 및 RCPs에 대한 자세한 내용은 AWS Organizations 사용 설명서의 리소스 제어 정책(RCPs)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. 가 여러 정책 유형이 관련될 때 요청을 허용할지 여부를 AWS 결정하는 방법을 알아보려면 IAM 사용 설명서의 정책 평가 로직을 참조하세요.

가 IAM과 AWS Serverless Application Repository 작동하는 방식

IAM을 사용하여에 대한 액세스를 관리하기 전에에서 사용할 수 있는 IAM 기능을 이해해야 AWS Serverless Application Repository합니다 AWS Serverless Application Repository.

IAM 작동 방식에 대한 개요를 보려면 IAM 사용 설명서의 IAM 작동 방식 이해를 참조하세요. AWS Serverless Application Repository 및 기타 AWS 서비스가 IAM에서 작동하는 방식을 전체적으로 알아보려면 IAM 사용 설명서의 AWS IAM에서 작동하는 서비스를 참조하세요.

주제

- AWS Serverless Application Repository ID 기반 정책
- AWS Serverless Application Repository 애플리케이션 정책
- AWS Serverless Application Repository 태그 기반 권한 부여
- AWS Serverless Application Repository IAM 역할

AWS Serverless Application Repository ID 기반 정책

IAM 자격 증명 기반 정책을 사용하면 허용되거나 거부되는 작업과 리소스는 물론, 작업이 허용되거나 거부되는 조건도 지정할 수 있습니다. AWS Serverless Application Repository 은 특정 작업, 리소스 및 조건 키를 지원합니다. JSON 정책에서 사용하는 모든 요소에 대해 알고 싶다면 IAM 사용 설명서의 IAM JSON 정책 요소 참조를 참조하세요.

다음은 권한 정책의 예입니다.

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CreateApplication",
            "Effect": "Allow",
            "Action": [
                "serverlessrepo:CreateApplication"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CreateApplicationVersion",
            "Effect": "Allow",
            "Action": [
                "serverlessrepo:CreateApplicationVersion"
            ],
            "Resource": "arn:partition:serverlessrepo:region:account-
id:applications/application-name"
        }
    ]
}
```

이 정책에는 두 명령문이 있습니다:

- 첫 번째 문은 와일드카드 문자(*)로 지정된 대로 모든 AWS Serverless Application Repository 리소스`serverlessrepo:CreateApplication`에 대해 AWS Serverless Application Repository 작업에 대한 권한을 부여합니다`Resource`.
- 두 번째 문은 AWS Serverless Application Repository 애플리케이션에 Amazon AWS 리소스 이름(ARN)을 사용하여 리소스`serverlessrepo:CreateApplicationVersion`에 대한 AWS Serverless Application Repository 작업에 대한 권한을 부여합니다. 애플리케이션은 `Resource` 값으로 지정됩니다.

자격 증명 기반 정책에서는 권한을 가질 보안 주체를 지정하지 않으므로 이 정책은 `Principal` 요소를 지정하지 않습니다. 정책을 사용자에게 연결할 경우 사용자는 암시적인 보안 주체입니다. IAM 역할에 권한 정책을 연결할 경우 역할의 신뢰 정책에 식별된 보안 주체는 권한을 가집니다.

모든 AWS Serverless Application Repository API 작업과 해당 작업이 적용되는 AWS 리소스를 보여주는 표는 섹션을 참조하세요AWS Serverless Application Repository API 권한: 작업 및 리소스 참조.

작업

관리자는 AWS JSON 정책을 사용하여 누가 무엇을 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 위탁자가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

JSON 정책의 `Action` 요소는 정책에서 액세스를 허용하거나 거부하는 데 사용할 수 있는 작업을 설명합니다. 정책 작업은 일반적으로 연결된 AWS API 작업과 이름이 동일합니다. 일치하는 API 작업이 없는 권한 전용 작업 같은 몇 가지 예외도 있습니다. 정책에서 여러 작업이 필요한 몇 가지 작업도 있습니다. 이러한 추가 작업을 일컬어 종속 작업이라고 합니다.

연결된 작업을 수행할 수 있는 권한을 부여하기 위한 정책에 작업을 포함하세요.

의 정책 작업은 작업 앞에 접두사를 AWS Serverless Application Repository 사용합니다`serverlessrepo:`. 예를 들어 API `SearchApplications` 작업으로 AWS Serverless Application Repository 인스턴스를 실행할 수 있는 AWS Serverless Application Repository 권한을 부여하려면 해당 정책에 `serverlessrepo:SearchApplications` 작업을 포함합니다. 정책 문에는 `Action` 또는 `NotAction` 요소가 포함되어야 합니다. 는이 서비스로 수행할 수 있는 작업을 설명하는 고유한 작업 세트를 AWS Serverless Application Repository 정의합니다.

명령문 하나에 여러 태스크를 지정하려면 다음과 같이 쉼표로 구분합니다.

```
"Action": [
      "serverlessrepo:action1",
      "serverlessrepo:action2"
]
```

와일드카드(*)를 사용하여 여러 작업을 지정할 수 있습니다. 예를 들어, `List`라는 단어로 시작하는 모든 태스크를 지정하려면 다음 태스크를 포함합니다.

```
"Action": "serverlessrepo:List*"
```

AWS Serverless Application Repository 작업 목록을 보려면 IAM 사용 설명서의 에서 정의한 작업을 AWS Serverless Application Repository 참조하세요.

리소스

관리자는 AWS JSON 정책을 사용하여 누가 무엇을 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

`Resource` JSON 정책 요소는 작업이 적용되는 하나 이상의 객체를 지정합니다. 문에는 `Resource`또는 `NotResource`요소가 반드시 추가되어야 합니다. 모범 사례에 따라 Amazon 리소스 이름(ARN)을 사용하여 리소스를 지정합니다. 리소스 수준 권한이라고 하는 특정 리소스 유형을 지원하는 작업에 대해 이를 수행할 수 있습니다.

작업 나열과 같이 리소스 수준 권한을 지원하지 않는 작업의 경우, 와일드카드(*)를 사용하여 해당 문이 모든 리소스에 적용됨을 나타냅니다.

```
"Resource": "*"
```

에서 AWS Serverless Application Repository기본 AWS 리소스는 다음 표와 AWS Serverless Application Repository 같이 application. AWS Serverless Application Repository applications에 고유한 Amazon 리소스 이름(ARNs)이 연결되어 있습니다.

AWS 리소스 유형	Amazon 리소스 이름(ARN) 형식
Application	arn:*partition* :serverlessrepo:*region*:*account-id* :applications/*application-name*

ARN 형식에 대한 자세한 내용은 Amazon 리소스 이름(ARNs) 및 AWS 서비스 네임스페이스를 참조하세요.

다음은 모든 AWS 리소스에 대해 `serverlessrepo:ListApplications` 작업에 대한 권한을 부여하는 정책의 예입니다. 현재 구현에서 AWS Serverless Application Repository 는 일부 API 작업에 AWS 리소스 ARNs(AWS 리소스 수준 권한이라고도 함)을 사용하여 특정 리소스를 식별하는 것을 지원하지 않습니다. 이 경우 와일드카드 문자(*)를 지정해야 합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ListExistingApplications",
            "Effect": "Allow",
            "Action": [
                "serverlessrepo:ListApplications"
            ],
            "Resource": "*"
        }
    ]
```

```
}
```

모든 AWS Serverless Application Repository API 작업과 해당 작업이 적용되는 AWS 리소스를 보여주는 표는 섹션을 참조하세요AWS Serverless Application Repository API 권한: 작업 및 리소스 참조.

조건 키

AWS Serverless Application Repository 는 서비스별 조건 키를 제공하지 않지만 일부 전역 조건 키 사용을 지원합니다. 모든 AWS 전역 조건 키를 보려면 IAM 사용 설명서의 AWS 전역 조건 컨텍스트 키를 참조하세요.

예시

AWS Serverless Application Repository 자격 증명 기반 정책의 예를 보려면 섹션을 참조하세요AWS Serverless Application Repository 자격 증명 기반 정책 예제.

AWS Serverless Application Repository 애플리케이션 정책

애플리케이션 정책에 따라 지정된 보안 주체 또는 principalOrg가 AWS Serverless Application Repository 애플리케이션에서 수행할 수 있는 작업이 결정됩니다.

AWS Serverless Application Repository 애플리케이션과 연결된 정책에 권한을 추가할 수 있습니다. AWS Serverless Application Repository 애플리케이션에 연결된 권한 정책을 애플리케이션 정책이라고 합니다. 애플리케이션 정책은 IAM 리소스 기반 정책의 확장입니다. 기본 리소스는 애플리케이션입니다 AWS Serverless Application Repository . AWS Serverless Application Repository 애플리케이션 정책을 사용하여 애플리케이션 배포 권한을 관리할 수 있습니다.

AWS Serverless Application Repository 애플리케이션 정책은 주로 게시자가 소비자에게 애플리케이션을 배포할 수 있는 권한과 해당 애플리케이션의 세부 정보를 검색하고 보는 등의 관련 작업을 부여하는 데 사용됩니다. 게시자는 다음과 같은 3가지 범주로 애플리케이션 권한을 설정할 수 있습니다.

- 프라이빗 - 동일한 계정으로 생성되었으며 다른 계정과 공유되지 않은 애플리케이션입니다. 계정을 사용하여 생성된 애플리케이션을 배포할 수 있는 권한이 있습니다 AWS .
- 비공개 공유 - 게시자가 특정 AWS 계정 또는 AWS 조직 집합과 명시적으로 공유한 애플리케이션입니다. 계정 또는 AWS 조직과 공유된 애플리케이션을 배포할 수 있는 권한이 있습니다 AWS .
- 공개 공유 - 게시자가 모든 사람과 공유한 애플리케이션입니다. 소비자는 공개적 공유 애플리케이션을 배포할 수 있는 권한이 있습니다.

AWS CLI, AWS SDKs 또는를 사용하여 권한을 부여할 수 있습니다 AWS Management Console.

예시

AWS Serverless Application Repository 애플리케이션 정책 관리의 예를 보려면 섹션을 참조하세요AWS Serverless Application Repository 애플리케이션 정책 예제.

AWS Serverless Application Repository 태그 기반 권한 부여

AWS Serverless Application Repository 는 태그를 기반으로 리소스 또는 작업에 대한 액세스 제어를 지원하지 않습니다.

AWS Serverless Application Repository IAM 역할

IAM 역할은 AWS 계정 내에서 특정 권한이 있는 엔터티입니다.

에서 임시 자격 증명 사용 AWS Serverless Application Repository

임시 자격 증명을 사용하여 페더레이션을 통해 로그인하거나, IAM 역할을 수임하거나, 교차 계정 역할을 수임할 수 있습니다. AssumeRole 또는 GetFederationToken과 같은 AWS STS API 작업을 호출하여 임시 보안 자격 증명을 얻습니다.

는 임시 자격 증명 사용을 AWS Serverless Application Repository 지원합니다.

서비스 연결 역할

AWS Serverless Application Repository 는 서비스 연결 역할을 지원하지 않습니다.

서비스 역할

AWS Serverless Application Repository 는 서비스 역할을 지원하지 않습니다.

AWS Serverless Application Repository 자격 증명 기반 정책 예제

기본적으로 IAM 사용자 및 역할은 AWS Serverless Application Repository 리소스를 생성하거나 수정할 수 있는 권한이 없습니다. 또한 AWS Management Console AWS CLI또는 AWS API를 사용하여 작업을 수행할 수 없습니다. IAM 관리자는 지정된 리소스에서 특정 API 작업을 수행할 수 있는 권한을 사용자와 역할에게 부여하는 IAM 정책을 생성해야 합니다. 그런 다음 관리자는 해당 권한이 필요한 IAM 사용자 또는 그룹에 이러한 정책을 연결해야 합니다.

이러한 예제 JSON 정책 문서를 사용하여 IAM 자격 증명 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서의 JSON 탭에서 정책 생성을 참조하세요.

주제

정책 모범 사례

자격 증명 기반 정책은 매우 강력합니다. 계정에서 사용자가 AWS Serverless Application Repository 리소스를 생성, 액세스 또는 삭제할 수 있는지 여부를 결정합니다. 이러한 작업으로 AWS 인해 계정에 비용이 발생할 수 있습니다. ID 기반 정책을 생성하거나 편집할 때는 다음 지침과 권장 사항을 따릅니다.

- 최소 권한 부여 - 사용자 지정 정책을 생성할 때 태스크를 수행하는 데 필요한 권한만 부여합니다. 최소한의 권한 조합으로 시작하여 필요에 따라 추가 권한을 부여합니다. 처음부터 권한을 많이 부여한 후 나중에 줄이는 방법보다 이 방법이 안전합니다. 자세한 내용은 IAM 사용 설명서에서 최소 권한 부여를 참조하세요.
- 중요한 작업에 대해 MFA 활성화 - 보안을 강화하기 위해 IAM 사용자가 중요한 리소스 또는 API 작업에 액세스할 때 멀티 팩터 인증(MFA)을 사용하도록 합니다. 자세한 내용은 IAM 사용 설명서의 AWS에서 멀티 팩터 인증(MFA) 사용을 참조하세요.
- 보안 강화를 위해 정책 조건 사용 – 실제로 가능한 경우 자격 증명 기반 정책이 리소스에 대한 액세스를 허용하는 조건을 정의합니다. 예를 들어 요청을 할 수 있는 IP 주소의 범위를 지정하도록 조건을 작성할 수 있습니다. 지정된 날짜 또는 시간 범위 내에서만 요청을 허용하거나, SSL 또는 MFA를 사용해야 하는 조건을 작성할 수도 있습니다. 자세한 정보는 IAM 사용 설명서의 IAM JSON 정책 요소: 조건을 참조하세요.

AWS Serverless Application Repository 콘솔 사용

AWS Serverless Application Repository 콘솔은 AWS Serverless Application Repository 애플리케이션을 검색하고 관리할 수 있는 통합 환경을 제공합니다. 콘솔은에 설명된 API별 권한 외에도 AWS Serverless Application Repository 애플리케이션을 관리하는 데 종종 권한이 필요한 기능과 워크플로를 제공합니다AWS Serverless Application Repository API 권한: 작업 및 리소스 참조.

AWS Serverless Application Repository 콘솔을 사용하는 데 필요한 권한에 대한 자세한 내용은 섹션을 참조하세요고객 관리형 정책 예.

사용자가 자신이 권한을 볼 수 있도록 허용

이 예제는 IAM 사용자가 자신의 사용자 ID에 연결된 인라인 및 관리형 정책을 볼 수 있도록 허용하는 정책을 생성하는 방법을 보여줍니다. 이 정책에는 콘솔에서 또는 AWS CLI 또는 AWS API를 사용하여 프로그래밍 방식으로이 작업을 완료할 수 있는 권한이 포함됩니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

고객 관리형 정책 예

이 섹션의 예에서는 사용자에게 연결할 수 있는 샘플 정책 그룹을 제공합니다. 정책을 처음 생성하는 경우 순서대로 먼저 계정에서 IAM 사용자를 생성하고 정책을 사용자에게 연결하는 것이 좋습니다. 또

한 이러한 예제를 사용하여 여러 작업을 수행할 수 있는 권한이 포함된 단일 사용자 지정 정책을 만든 다음 사용자에게 연결할 수 있습니다.

정책을 사용자에게 연결하는 방법에 대한 자세한 내용은 IAM 사용 설명서의 사용자에게 권한 추가를 참조하세요.

예시

- 게시자 예제 1: 게시자에게 애플리케이션 목록 생성 허용
- 게시자 예제 2: 게시자에게 애플리케이션 또는 애플리케이션 버전 세부 정보 보기 허용
- 게시자 예제 3: 게시자에게 애플리케이션 또는 애플리케이션 버전 생성 허용
- 게시자 예제 4: 게시자에게 애플리케이션 정책 생성 및 애플리케이션 공유 허용
- 소비자 예제 1: 소비자에게 애플리케이션 검색 허용
- 소비자 예제 2: 소비자에게 애플리케이션 세부 정보 보기 허용
- 소비자 예제 3: 소비자에게 애플리케이션 배포 허용
- 소비자 예제 4: 배포 자산에 대한 액세스 거부
- 소비자 예제 5: 소비자의 공개 애플리케이션 검색 및 배포 방지

게시자 예제 1: 게시자에게 애플리케이션 목록 생성 허용

계정의 IAM 사용자는 `serverlessrepo:ListApplications` 작업 권한이 있어야만 콘솔의 내용을 확인할 수 있습니다. 이러한 권한을 부여하면 콘솔에 사용자가 속한 특정 AWS 리전에서 생성된 AWS 계정의 AWS Serverless Application Repository 애플리케이션 목록이 표시될 수 있습니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ListExistingApplications",
            "Effect": "Allow",
            "Action": [
                "serverlessrepo:ListApplications"
            ],
            "Resource": "*"
        }
    ]
}
```

게시자 예제 2: 게시자에게 애플리케이션 또는 애플리케이션 버전 세부 정보 보기 허용

사용자는 AWS Serverless Application Repository 애플리케이션을 선택하고 애플리케이션의 세부 정보를 볼 수 있습니다. 이러한 세부 정보에는 작성자, 설명, 버전 및 기타 구성 정보가 포함됩니다. 이를 수행하려면 사용자에게 AWS Serverless Application Repository에 대한 `serverlessrepo:GetApplication` 및 `serverlessrepo:ListApplicationVersions` 작업 권한이 필요합니다.

다음 예제에서 Amazon 리소스 이름(ARN)이 `Resource` 값으로 지정된 특정 애플리케이션에 대해 이러한 권한이 부여됩니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewApplication",
            "Effect": "Allow",
            "Action": [
                "serverlessrepo:GetApplication",
                "serverlessrepo:ListApplicationVersions"
            ],
            "Resource": "arn:aws:serverlessrepo:region:account-
id:applications/application-name"
        }
    ]
}
```

게시자 예제 3: 게시자에게 애플리케이션 또는 애플리케이션 버전 생성 허용

사용자가 AWS Serverless Application Repository 애플리케이션을 생성할 수 있는 권한을 갖도록 허용하려면 다음 정책과 같이 `serverlessrepo:CreateApplication` 및 `serverlessrepo:CreateApplicationVersions` 작업에 권한을 부여해야 합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CreateApplication",
            "Effect": "Allow",
            "Action": [
```

```
                "serverlessrepo:CreateApplication",
                "serverlessrepo:CreateApplicationVersion",
            ],
            "Resource": "*"
        }
    ]
}
```

게시자 예제 4: 게시자에게 애플리케이션 정책 생성 및 애플리케이션 공유 허용

사용자가 애플리케이션을 다른 사용자와 공유하려면 사용자에게 다음 정책에 나온 것과 같이 애플리케이션 정책을 생성할 권한을 부여해야 합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ShareApplication",
            "Effect": "Allow",
            "Action": [
                "serverlessrepo:PutApplicationPolicy",
                "serverlessrepo:GetApplicationPolicy",
            ],
            "Resource": "*"
        }
    ]
}
```

소비자 예제 1: 소비자에게 애플리케이션 검색 허용

소비자가 애플리케이션을 검색하려면 소비자에게 다음 권한을 부여해야 합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "SearchApplications",
            "Effect": "Allow",
            "Action": [
```

```
                "serverlessrepo:SearchApplications"
            ],
            "Resource": "*"
        }
    ]
}
```

소비자 예제 2: 소비자에게 애플리케이션 세부 정보 보기 허용

사용자는 AWS Serverless Application Repository 애플리케이션을 선택하고 작성자, 설명, 버전 및 기타 구성 정보와 같은 애플리케이션의 세부 정보를 볼 수 있습니다. 이렇게 하려면 사용자에게 다음 AWS Serverless Application Repository 작업에 대한 권한이 있어야 합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewApplication",
            "Effect": "Allow",
            "Action": [
                "serverlessrepo:GetApplication",
                "serverlessrepo:ListApplicationVersions"
            ],
            "Resource": "*"
        }
    ]
}
```

소비자 예제 3: 소비자에게 애플리케이션 배포 허용

소비자가 애플리케이션을 배포하려면 소비자에게 여러 작업을 수행할 권한을 부여해야 합니다. 다음 정책은 사용자에게 필요한 권한을 제공합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DeployApplication",
            "Effect": "Allow",
```

```
            "Action": [
                "serverlessrepo:CreateCloudFormationChangeSet",
                "cloudformation:CreateChangeSet",
                "cloudformation:ExecuteChangeSet",
                "cloudformation:DescribeStacks"

            ],
            "Resource": "*"
        }
    ]
}
```

> **Note**
>
> 애플리케이션을 배포하려면 추가 AWS 리소스를 사용할 권한이 필요할 수 있습니다. 는와 동일한 기본 배포 메커니즘을 AWS Serverless Application Repository 사용하므로 자세한 내용은 AWS Identity and Access Management를 사용한 액세스 제어를 AWS CloudFormation참조하세요. 또한 권한 관련 배포 문제에 도움이 되도록 문제 해결: IAM 권한 부족을 참조할 수 있습니다.

소비자 예제 4: 배포 자산에 대한 액세스 거부

애플리케이션이 AWS 계정과 비공개로 공유되는 경우 기본적으로 해당 계정의 모든 사용자는 동일한 계정에 있는 다른 모든 사용자의 배포 자산에 액세스할 수 있습니다. 다음 정책은 계정의 사용자가의 Amazon S3 버킷에 저장된 배포 자산에 액세스하지 못하도록 합니다 AWS Serverless Application Repository.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DenyDeploymentAssetAccess",
            "Effect": "Deny",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::awsserverlessrepo-changesets*/*"
            ]
        }
```

```
    ]
}
```

소비자 예제 5: 소비자의 공개 애플리케이션 검색 및 배포 방지

사용자가 애플리케이션에 대해 특정 작업을 수행하지 못하게 할 수 있습니다.

`serverlessrepo:applicationType`을 `public`으로 지정하여 다음 정책을 공개 애플리케이션을 적용합니다. 그러면 사용자가 `Effect`를 `Deny`로 지정하여 각종 작업을 수행하지 못하게 됩니다. 사용 가능한 조건 키에 대한 자세한 내용은에 대한 작업, 리소스 및 조건 키를 AWS Serverless Application Repository참조하세요. AWS Serverless Application Repository

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Condition": {
                "StringEquals": {
                    "serverlessrepo:applicationType": "public"
                }
            },
            "Action": [
                "serverlessrepo:SearchApplications",
                "serverlessrepo:GetApplication",
                "serverlessrepo:CreateCloudFormationTemplate",
                "serverlessrepo:CreateCloudFormationChangeSet",
                "serverlessrepo:ListApplicationVersions",
                "serverlessrepo:ListApplicationDependencies"
            ],
            "Resource": "*",
            "Effect": "Deny"
        }
    ]
}
```

> **Note**
>
> 이 정책 설명은 서비스 제어 정책으로 사용되어 AWS 조직에 적용될 수도 있습니다. 서비스 제어 정책에 대한 자세한 내용은 AWS Organizations 사용 설명서의 서비스 제어 정책을 참조하세요.

AWS Serverless Application Repository 애플리케이션 정책 예제

AWS Serverless Application Repository 애플리케이션에 연결된 권한 정책을 애플리케이션 정책이라고 합니다. 애플리케이션 정책은 지정된 보안 주체 또는 principalOrg가 AWS Serverless Application Repository 애플리케이션에서 수행할 수 있는 작업을 결정합니다.

AWS Serverless Application Repository 애플리케이션은의 기본 AWS 리소스입니다 AWS Serverless Application Repository. AWS Serverless Application Repository 애플리케이션 정책은 주로 게시자가 소비자에게 애플리케이션을 배포할 수 있는 권한을 부여하고 해당 애플리케이션의 세부 정보를 검색하고 보는 등의 관련 작업을 부여하는 데 사용됩니다.

게시자는 다음과 같은 3가지 범주로 애플리케이션 권한을 설정할 수 있습니다.

- 프라이빗 - 동일한 계정으로 생성되었으며 다른 계정과 공유되지 않은 애플리케이션입니다. AWS 계정을 공유하는 소비자만 프라이빗 애플리케이션을 배포할 수 있는 권한이 있습니다.
- 비공개 공유 - 게시자가 특정 AWS 계정 집합 또는 AWS 조직의 계정과 AWS 명시적으로 공유한 애플리케이션입니다. 소비자는 AWS 계정 또는 AWS 조직과 공유된 애플리케이션을 배포할 수 있는 권한이 있습니다. AWS 조직에 대한 자세한 내용은 AWS Organizations 사용 설명서를 참조하세요.
- 공개 공유 - 게시자가 모든 사람과 공유한 애플리케이션입니다. 모든 소비자는 공개적 공유 애플리케이션을 배포할 수 있는 권한이 있습니다.

> **Note**
>
> 비공개 공유 애플리케이션의 경우 AWS Serverless Application Repository 만 AWS 계정을 보안 주체로 지원합니다. 게시자는 AWS 계정 내 모든 사용자를 애플리케이션에 단일 그룹으로 부여하거나 거부할 수 AWS Serverless Application Repository 있습니다. 게시자는 AWS 계정 내의 개별 사용자에게 애플리케이션에 대한 권한을 부여하거나 거부할 수 AWS Serverless Application Repository 없습니다.

를 사용하여 애플리케이션 권한을 설정하는 방법에 대한 지침은 섹션을 AWS Management Console참조하세요애플리케이션 공유.

AWS CLI 및 예제를 사용하여 애플리케이션 권한을 설정하는 방법에 대한 지침은 다음 섹션을 참조하세요.

애플리케이션 권한(AWS CLI 및 AWS SDKs)

AWS CLI 또는 AWS SDKs를 사용하여 애플리케이션에 대한 AWS Serverless Application Repository 권한을 설정하는 경우 다음 작업을 지정할 수 있습니다.

작업	설명
GetApplication	애플리케이션 관련 정보를 볼 수 있는 권한을 부여합니다.
CreateCloudFormationChangeSet	애플리케이션이 배포될 권한을 부여합니다. 참고: 이 작업은 배포 외에 어떠한 권한도 부여하지 않습니다.
CreateCloudFormationTemplate	애플리케이션에 대한 AWS CloudFormation 템플릿을 생성할 수 있는 권한을 부여합니다.
ListApplicationVersions	애플리케이션의 버전 목록 작성 권한을 부여합니다.
ListApplicationDependencies	컨테이닝 애플리케이션에 중첩 애플리케이션의 목록을 표시할 수 있는 권한을 부여합니다.
SearchApplications	애플리케이션이 검색될 권한을 부여합니다.
배포	이 표의 위에 나열된 모든 작업을 가능하게 합니다 즉, 애플리케이션을 보고, 배포하고, 검색하고, 버전 목록을 작성할 수 있는 권한을 부여합니다.

애플리케이션 정책 예제

다음 예제에서는 AWS CLI를 사용하여 권한을 부여하는 방법을 보여 줍니다. 를 사용하여 권한을 부여하는 방법에 대한 자세한 내용은 섹션을 AWS Management Console참조하세요애플리케이션 공유.

이 섹션의 모든 예제는 다음 AWS CLI 명령을 사용하여 AWS Serverless Application Repository 애플리케이션과 연결된 권한 정책을 관리합니다.

- put-application-policy
- get-application-policy

주제

예제 1: 다른 계정과 애플리케이션 공유

애플리케이션을 다른 특정 계정과 공유하지만 다른 계정과 공유되지 않도록 하려면 보안 주체로 공유할 AWS 계정 ID를 지정합니다. 다시 말해 애플리케이션을 비공개 공유로 설정합니다. 이렇게 하려면 다음 AWS CLI 명령을 사용합니다.

```
aws serverlessrepo put-application-policy \
--region region \
--application-id application-arn \
--statements Principals=account-id,Actions=Deploy
```

> **Note**
>
> 프라이빗 공유 애플리케이션은 애플리케이션이 생성된 리전과 동일한 AWS 리전에서만 사용할 수 있습니다.

예제 2: 애플리케이션을 공개적으로 공유

애플리케이션을 공개하려면 다음 예제와 같이 보안 주체로 "*"를 지정하여 모두와 공유합니다. 공개적으로 공유되는 애플리케이션은 모든 리전에서 사용할 수 있습니다.

```
aws serverlessrepo put-application-policy \
--region region \
--application-id application-arn \
--statements Principals=*,Actions=Deploy
```

> **Note**
>
> 애플리케이션을 공개적으로 공유하려면 `SemanticVersion` 및 `LicenseUrl` 속성이 둘 다 설정되어 있어야 합니다.

예제 3: 애플리케이션 비공개

애플리케이션을 비공개로 설정할 수 있으므로 다른 사람과 공유되지 않으며 애플리케이션을 소유한 AWS 계정에서만 배포할 수 있습니다. 이렇게 하려면 정책에서 보안 주체와 작업을 지웁니다. 그러면 AWS 조직 내 다른 계정의 권한도 제거되어 애플리케이션을 배포할 수 없습니다.

```
aws serverlessrepo put-application-policy \
--region region \
--application-id application-arn \
--statements '[]'
```

> **Note**
>
> 프라이빗 애플리케이션은 애플리케이션이 생성된 리전과 동일한 AWS 리전에서만 사용할 수 있습니다.

예제 4: 여러 계정 및 권한 지정

여러 권한을 부여할 수 있으며 한 번에 두 AWS 개 이상의 계정에 부여할 수 있습니다. 이렇게 하려면 다음 예제와 같이 보안 주체 및 작업으로 목록을 지정합니다.

```
aws serverlessrepo put-application-policy \
--region region \
--application-id application-arn \
--statements Principals=account-id-1,account-
id-2,Actions=GetApplication,CreateCloudFormationChangeSet
```

예제 5: AWS 조직의 모든 계정과 애플리케이션 공유

AWS 조직 내 모든 사용자에게 권한을 부여할 수 있습니다. 다음 예와 같이 조직 ID를 지정하면 됩니다.

```
aws serverlessrepo put-application-policy \
--region region \
```

```
--application-id application-arn \
--statements Principals=*,PrincipalOrgIDs=org-id,Actions=Deploy,UnshareApplication
```

AWS 조직에 대한 자세한 내용은 AWS Organizations 사용 설명서를 참조하세요.

> **Note**
>
> AWS 계정이 속한 AWS 조직만 지정할 수 있습니다. 사용자가 멤버가 아닌 AWS 조직을 지정하려고 하면 오류가 발생합니다.
> 애플리케이션을 AWS 조직과 공유하려면 나중에 공유를 취소해야 하는 경우를 대비하여 UnshareApplication 작업에 대한 권한을 포함해야 합니다.

예제 6: 조직의 일부 계정 AWS 과 애플리케이션 공유

AWS 조직 내 특정 계정에 권한을 부여할 수 있습니다. 다음 예제와 같이 AWS 계정 목록을 보안 주체로 지정하고 조직 ID를 지정하여이 작업을 수행합니다.

```
aws serverlessrepo put-application-policy \
--region region \
--application-id application-arn \
--statements Principals=account-id-1,account-id-2,PrincipalOrgIDs=org-
id,Actions=Deploy,UnshareApplication
```

> **Note**
>
> AWS 계정이 속한 AWS 조직만 지정할 수 있습니다. 사용자가 멤버가 아닌 AWS 조직을 지정하려고 하면 오류가 발생합니다.
> 애플리케이션을 AWS 조직과 공유하려면 나중에 공유를 취소해야 하는 경우를 대비하여 UnshareApplication 작업에 대한 권한을 포함해야 합니다.

예제 7: 애플리케이션 정책 검색

현재 공유 상태인지 여부를 확인하기 위해서 등 애플리케이션의 현재 정책을 보려면 다음 예제와 같이 get-application-policy 명령을 사용합니다.

```
aws serverlessrepo get-application-policy \
--region region \
--application-id application-arn
```

예제 8: 특정 계정에서 애플리케이션을 중첩할 수 있도록 허용

퍼블릭 애플리케이션을 누구나 중첩할 수 있습니다. 애플리케이션을 특정 계정에서 중첩할 수 있도록 허용하려면 다음 예제에 나온 것처럼 최소한의 사용 권한을 설정해야 합니다.

```
aws serverlessrepo put-application-policy \
--region region \
--application-id application-arn \
--statements Principals=account-id-1,account-
id-2,Actions=GetApplication,CreateCloudFormationTemplate
```

AWS Serverless Application Repository API 권한: 작업 및 리소스 참조

IAM 자격 증명에 연결할 수 있는 액세스 제어 및 쓰기 권한 정책(자격 증명 기반 정책)을 설정할 때 다음 표를 참조로 사용할 수 있습니다. 테이블에는 있습니다. AWS Serverless Application Repository AWS 정책의 `Action`필드에서 작업을 지정하고, 정책의 `Resource`필드에서 리소스 값을 지정합니다.

작업을 지정하려면 `serverlessrepo:` 접두사 다음에 API 작업 명칭을 사용합니다(예: `serverlessrepo:ListApplications`).

Operation	URI	메서드	AWS 리소스(ARNs)
작업: ListApplications 필요한 권한: serverlessrepo:ListApplications	/애플리케이션	GET	*
작업: CreateApplication 필요한 권한: serverlessrepo:CreateApplication	/애플리케이션	POST	*
작업: GetApplication 필요한 권한: serverlessrepo:GetApplication	/applications/*application-id*	GET	arn:aws:serverlessrepo:*region*:*account-id* :applications/*application-name*

Operation	URI	메서드	AWS 리소스(ARNs)
작업: DeleteApplication 필요한 권한: serverlessrepo:DeleteApplication	/applications/*application-id*	DELETE	arn:aws:serverlessrepo:*region*:*account-id* :applications/*application-name*
작업: UpdateApplication 필요한 권한: serverlessrepo:UpdateApplication	/applications/*application-id*	PATCH	arn:aws:serverlessrepo:*region*:*account-id* :applications/*application-name*
작업: CreateCloudFormationChangeSet 필요한 권한: serverlessrepo:CreateCloudFormationChangeSet	/applications/*application-id*/changesets	POST	arn:aws:serverlessrepo:*region*:*account-id* :applications/*application-name*
작업: GetApplicationPolicy 필요한 권한: serverlessrepo:GetApplicationPolicy	/applications/*application-id*/policy	GET	arn:aws:serverlessrepo:*region*:*account-id* :applications/*application-name*
작업: PutApplicationPolicy 필요한 권한: serverlessrepo:PutApplicationPolicy	/applications/*application-id*/policy	PUT	arn:aws:serverlessrepo:*region*:*account-id* :applications/*application-name*

Operation	URI	메서드	AWS 리소스(ARNs)
작업: ListApplicationVersions 필요한 권한: serverlesrepo:ListApplicationVersions	/applications/*application-id*/versions	GET	arn:aws:serverlessrepo:*region*:*account-id* :applications/*application-name*
작업: CreateApplicationVersion 필요한 권한: serverlesrepo:CreateApplicationVersion	/applications/*application-id* /versions/*semantic-version*	PUT	arn:aws:serverlessrepo:*region*:*account-id* :applications/*application-name*
작업: ListApplicationDependencies 필요한 권한: serverlesrepo:ListApplicationDependencies	/applications/*application-id*/dependencies	GET	arn:aws:serverlessrepo:*region*:*account-id* :applications/*application-name*
작업: SearchApplications 필요한 권한: serverlesrepo:SearchApplications	해당 사항 없음	해당 사항 없음	*

AWS Serverless Application Repository 자격 증명 및 액세스 문제 해결

다음 정보를 사용하여 및 IAM으로 작업할 때 발생할 수 있는 일반적인 문제를 진단 AWS Serverless Application Repository 하고 수정할 수 있습니다.

주제

- AWS Serverless Application Repository에서 작업을 수행할 권한이 없음
- iam:PassRole을 수행하도록 인증되지 않음

- 관리자인데 다른 사람이에 액세스하도록 허용하고 싶습니다. AWS Serverless Application Repository
- 내 AWS 계정 외부의 사람이 내 AWS Serverless Application Repository 리소스에 액세스하도록 허용하고 싶습니다.

AWS Serverless Application Repository에서 작업을 수행할 권한이 없음

에서 작업을 수행할 권한이 없다는 AWS Management Console 메시지가 표시되면 관리자에게 문의하여 지원을 받아야 합니다. 관리자는 사용자 이름과 비밀번호를 제공한 사람입니다.

다음 예제 오류는 `mateojackson` IAM 사용자가 콘솔을 사용하여 애플리케이션에 대한 세부 정보를 보려고 하지만 `serverlessrepo:GetApplication` 권한이 없는 경우에 발생합니다.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 serverlessrepo:GetApplication on resource: my-example-application
```

이 경우 Mateo는 `serverlessrepo:GetApplication` 작업을 사용하여 *my-example-application* 리소스에 액세스하도록 허용하는 정책을 업데이트할 것을 관리자에게 요청합니다.

iam:PassRole을 수행하도록 인증되지 않음

`iam:PassRole` 작업을 수행할 수 있는 권한이 없다는 오류가 수신되면 AWS Serverless Application Repository에 역할을 전달할 수 있도록 정책을 업데이트해야 합니다.

일부 AWS 서비스 에서는 새 서비스 역할 또는 서비스 연결 역할을 생성하는 대신 기존 역할을 해당 서비스에 전달할 수 있습니다. 이렇게 하려면 사용자가 서비스에 역할을 전달할 수 있는 권한을 가지고 있어야 합니다.

다음 예 오류는 `marymajor`라는 IAM 사용자가 콘솔을 사용하여 AWS Serverless Application Repository에서 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 자격 증명을 제공한 사람입니다.

관리자인데 다른 사람이에 액세스하도록 허용하고 싶습니다. AWS Serverless Application Repository

다른 사용자가 액세스하도록 허용하려면 액세스가 필요한 사용자 또는 애플리케이션에 권한을 부여해야 AWS Serverless Application Repository합니다. AWS IAM Identity Center 를 사용하여 사용자 및 애플리케이션을 관리하는 경우 사용자 또는 그룹에 권한 세트를 할당하여 액세스 수준을 정의합니다. 권한 세트는 IAM 정책을 자동으로 생성하고 사용자 또는 애플리케이션과 연결된 IAM 역할에 할당합니다. 자세한 내용은 AWS IAM Identity Center 사용 설명서에서 권한 세트를 참조하세요.

IAM Identity Center를 사용하지 않는 경우 액세스가 필요한 사용자 또는 애플리케이션에 대한 IAM 엔터티(사용자 또는 역할)를 생성해야 합니다. 그런 다음 AWS Serverless Application Repository에 대한 올바른 권한을 부여하는 정책을 엔터티에 연결해야 합니다. 권한이 부여되면 사용자 또는 애플리케이션 개발자에게 자격 증명을 제공합니다. 이들은 이 자격 증명을 사용하여 AWS에 액세스합니다. IAM 사용자, 그룹, 정책, 권한 생성에 대한 자세한 내용은 IAM 사용 설명서의 IAM ID 및 IAM 정책과 권한을 참조하세요.

내 AWS 계정 외부의 사람이 내 AWS Serverless Application Repository 리소스에 액세스하도록 허용하고 싶습니다.

다른 계정의 사용자 또는 조직 외부의 사람이 리소스에 액세스할 때 사용할 수 있는 역할을 생성할 수 있습니다. 역할을 수임할 신뢰할 수 있는 사람을 지정할 수 있습니다. 리소스 기반 정책 또는 액세스 제어 목록(ACL)을 지원하는 서비스의 경우, 이러한 정책을 사용하여 다른 사람에게 리소스에 대한 액세스 권한을 부여할 수 있습니다.

자세히 알아보려면 다음을 참조하세요.

- 가 이러한 기능을 AWS Serverless Application Repository 지원하는지 여부를 알아보려면 섹션을 참조하세요가 IAM과 AWS Serverless Application Repository 작동하는 방식.
- 소유 AWS 계정 한의 리소스에 대한 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 소유 AWS 계정 한 다른의 IAM 사용자에게 액세스 권한 제공을 참조하세요.
- 타사에 리소스에 대한 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 타사 AWS 계정 소유에 대한 액세스 권한 제공을 AWS 계정참조하세요.
- 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.

AWS Serverless Application Repository의 로깅 및 모니터링

모니터링은 AWS 솔루션의 신뢰성, 가용성 및 성능을 유지하는 데 중요한 부분입니다. 다중 지점 장애가 발생할 경우 더 쉽게 디버깅할 수 있도록 AWS 솔루션의 모든 부분에서 모니터링 데이터를 수집해야 합니다.는 다음과 같이 AWS Serverless Application Repository 리소스를 모니터링하고 잠재적 인시던트에 대응하기 위한 여러 도구를 AWS 제공합니다.

AWS CloudTrail 로그

> AWS Serverless Application Repository 는에서 사용자 AWS CloudTrail, 역할 또는 서비스가 수행한 작업에 대한 레코드를 제공하는 AWS 서비스와 통합됩니다 AWS Serverless Application Repository. CloudTrail은에 대한 모든 API 호출을 이벤트 AWS Serverless Application Repository 로 캡처합니다.

주제

를 사용하여 AWS Serverless Application Repository API 호출 로깅 AWS CloudTrail

AWS Serverless Application Repository 는에서 사용자 AWS CloudTrail, 역할 또는 서비스가 수행한 작업에 대한 레코드를 제공하는 AWS 서비스인와 통합됩니다 AWS Serverless Application Repository. CloudTrail은에 대한 모든 API 호출을 이벤트 AWS Serverless Application Repository 로 캡처합니다. 캡처된 호출에는 AWS Serverless Application Repository 콘솔의 호출과 AWS Serverless Application Repository API 작업에 대한 코드 호출이 포함됩니다.

추적을 생성하는 경우에 대한 이벤트를 포함하여 CloudTrail 이벤트를 Amazon S3 버킷으로 지속적으로 전송할 수 있습니다 AWS Serverless Application Repository. 트레일을 구성하지 않은 경우에도 CloudTrail 콘솔의 이벤트 기록에서 최신 이벤트를 볼 수 있습니다.

CloudTrail에서 수집한 정보를 사용하여에 수행된 요청을 확인할 수 있습니다 AWS Serverless Application Repository. 또한 어떤 IP 주소에서 요청했는지, 누가 언제 요청했는지 등의 추가 세부 정보도 확인할 수 있습니다.

CloudTrail에 대한 자세한 설명은 AWS CloudTrail 사용자 가이드를 참조하십시오.

AWS Serverless Application Repository CloudTrail의 정보

AWS 계정을 생성할 때 계정에서 CloudTrail이 활성화됩니다. 에서 활동이 발생하면 AWS Serverless Application Repository해당 활동이 이벤트 기록의 다른 AWS 서비스 이벤트와 함께 CloudTrail 이벤트에 기록됩니다. AWS 계정에서 최근 이벤트를 보고 검색하고 다운로드할 수 있습니다. 자세한 정보는 CloudTrail 이벤트 기록을 사용하여 이벤트 보기를 참조하세요.

에 대한 이벤트를 포함하여 AWS 계정의 이벤트를 지속적으로 기록하려면 추적을 AWS Serverless Application Repository생성합니다. CloudTrail은 추적을 사용하여 Amazon S3 버킷으로 로그 파일을 전송할 수 있습니다. 기본적으로 콘솔에서 추적을 생성하면 추적이 모든 AWS 리전에 적용됩니다. 추적은 AWS 파티션의 모든 AWS 리전에서 이벤트를 로깅하고 지정한 Amazon S3 버킷으로 로그 파일을 전송합니다. 또한 CloudTrail 로그에서 수집된 이벤트 데이터를 추가로 분석하고 조치를 취하도록 다른 AWS 서비스를 구성할 수 있습니다. 자세한 내용은 다음 자료를 참조하세요.

- 트레일 생성 개요
- CloudTrail 지원 서비스 및 통합
- CloudTrail에서 Amazon SNS 알림 구성
- 여러 리전으로부터 CloudTrail 로그 파일 받기 및 여러 계정으로부터 CloudTrail 로그 파일 받기

모든 AWS Serverless Application Repository 작업은 CloudTrail에서 로깅되며 AWS Serverless Application Repository 리소스 페이지에 문서화됩니다. 예컨대, `CreateApplication`, `UpdateApplications`및 `ListApplications`작업에 대한 호출은 CloudTrail 로그 파일의 항목을 생성합니다.

모든 이벤트 또는 로그 항목에는 요청을 생성했던 사용자에 대한 정보가 포함됩니다. 자격 증명을 이용하면 다음을 쉽게 판단할 수 있습니다.

- 요청이 루트 또는 AWS Identity and Access Management (IAM) 사용자 자격 증명으로 이루어졌는지 여부입니다.
- 역할 또는 페더레이션 사용자에 대한 임시 보안 인증을 사용하여 요청이 생성되었는지 여부.
- 다른 AWS 서비스에서 요청을 했는지 여부입니다.

자세한 설명은 CloudTrail userIdentity 요소를 참조하세요.

AWS Serverless Application Repository 로그 파일 항목 이해

추적이란 지정한 Amazon S3 버킷에 이벤트를 로그 파일로 입력할 수 있게 하는 구성입니다. CloudTrail 로그 파일에는 하나 이상의 로그 항목이 포함될 수 있습니다. 이벤트는 모든 소스로부터의 단일 요청을 나타내며 요청 작업, 작업 날짜와 시간, 요청 파라미터 등에 대한 정보가 들어 있습니다. CloudTrail 로그 파일은 퍼블릭 API 직접 호출의 주문 스택 트레이스가 아니므로 특정 순서로 표시되지 않습니다.

다음은 `CreateApplication` 작업을 보여주는 CloudTrail 로그 항목이 나타낸 예시입니다.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "Root",
        "principalId": "999999999999",
        "arn": "arn:aws:iam::999999999999:root",
        "accountId": "999999999999",
        "accessKeyId": "ASIAUVPLBDH76HEXAMPLE",
        "sessionContext": {
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2018-07-30T16:40:42Z"
            }
        },
        "invokedBy": "signin.amazonaws.com"
    },
    "eventTime": "2018-07-30T17:37:37Z",
    "eventSource": "serverlessrepo.amazonaws.com",
    "eventName": "CreateApplication",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "72.21.217.161",
    "userAgent": "signin.amazonaws.com",
    "requestParameters": {
        "licenseBody": "<content of license>",
        "sourceCodeUrl": "<sample url>",
        "spdxLicenseId": "<sample license id>",
        "readmeBody": "<content of readme>",
        "author": "<author name>",
        "templateBody": "<content of SAM template>",
        "name": "<application name>",
        "semanticVersion": "<version>",
        "description": "<content of description>",
        "homePageUrl": "<sample url>",
```

```
        "labels": [
            "<label1>",
            "<label2>"
        ]
    },
    "responseElements": {
        "licenseUrl": "<url to access content of license>",
        "readmeUrl": "<url to access content of readme>",
        "spdxLicenseId": "<sample license id>",
        "creationTime": "2018-07-30T17:37:37.045Z",
        "author": "<author name>",
        "name": "<application name>",
        "description": "<content of description>",
        "applicationId": "arn:aws:serverlessrepo:us-
east-1:999999999999:applications/<application name>",
        "homePageUrl": "<sample url>",
        "version": {
            "applicationId": "arn:aws:serverlessrepo:us-
east-1:999999999999:applications/<application name>",
            "semanticVersion": "<version>",
            "sourceCodeUrl": "<sample url>",
            "templateUrl": "<url to access content of SAM template>",
            "creationTime": "2018-07-30T17:37:37.027Z",
            "parameterDefinitions": [
                {
                    "name": "<parameter name>",
                    "description": "<parameter description>",
                    "type": "<parameter type>"
                }
            ]
        },
        "labels": [
            "<label1>",
            "<label2>"
        ]
    },
    "requestID": "3f50d899-941f-11e8-ab18-01063f863be5",
    "eventID": "a66a6490-d388-4a4f-8c7b-9d6ec61ab262",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "recipientAccountId": "999999999999"
}
```

에 대한 규정 준수 검증 AWS Serverless Application Repository

타사 감사자는 여러 규정 준수 프로그램의 AWS Serverless Application Repository 일환으로의 보안 및 AWS 규정 준수를 평가합니다. 여기에는 SOC, PCI, FedRAMP 등이 포함됩니다.

특정 규정 준수 프로그램의 범위에 속하는 AWS 서비스 목록은 AWS 규정 준수 프로그램 제공 범위 내 서비스를 참조하세요. 일반 정보는 AWS 규정 준수 프로그램을 참조하세요.

를 사용하여 타사 감사 보고서를 다운로드할 수 있습니다 AWS Artifact. 자세한 내용은 AWS 아티팩트에서 보고서 다운로드를 참조하세요.

를 사용할 때의 규정 준수 책임은 데이터의 민감도, 회사의 규정 준수 목표 및 관련 법률과 규정에 따라 결정됩니다. AWS Serverless Application Repository AWS 는 규정 준수에 도움이 되는 다음 리소스를 제공합니다.

- 보안 및 규정 준수 빠른 시작 안내서 -이 배포 안내서에서는 아키텍처 고려 사항에 대해 설명하고 보안 중심 및 규정 준수 중심 기준 환경을 배포하기 위한 단계를 제공합니다 AWS.
- AWS 규정 준수 리소스 -이 워크북 및 가이드 모음은 업계 및 위치에 적용될 수 있습니다.
- AWS Config -이 AWS 서비스는 리소스 구성이 내부 관행, 업계 지침 및 규정을 얼마나 잘 준수하는지 평가합니다.
- AWS Security Hub -이 AWS 서비스는 보안 업계 표준 및 모범 사례 준수를 확인하는 데 도움이 AWS 되는 내 보안 상태에 대한 포괄적인 보기를 제공합니다.

의 복원력 AWS Serverless Application Repository

AWS 글로벌 인프라는 AWS 리전 및 가용 영역을 기반으로 구축됩니다. AWS 리전은 지연 시간이 짧고 처리량이 높으며 중복성이 높은 네트워킹과 연결된 물리적으로 분리되고 격리된 여러 가용 영역을 제공합니다. 가용 영역을 사용하면 중단 없이 가용 영역 간에 자동으로 장애 조치가 이루어지는 애플리케이션 및 데이터베이스를 설계하고 운영할 수 있습니다. 가용 영역은 기존의 단일 또는 복수 데이터 센터 인프라보다 가용성, 내결함성, 확장성이 뛰어납니다.

AWS 리전 및 가용 영역에 대한 자세한 내용은 AWS 글로벌 인프라를 참조하세요.

의 인프라 보안 AWS Serverless Application Repository

관리형 서비스인는 AWS 글로벌 네트워크 보안으로 보호 AWS Serverless Application Repository 됩니다. AWS 보안 서비스 및가 인프라를 AWS 보호하는 방법에 대한 자세한 내용은 AWS 클라우드 보안

을 참조하세요. 인프라 보안 모범 사례를 사용하여 AWS 환경을 설계하려면 Security Pillar AWS Well-Architected Framework의 인프라 보호를 참조하세요.

AWS 게시된 API 호출을 사용하여 네트워크를 AWS Serverless Application Repository 통해에 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 자격 증명을 생성하여 요청에 서명할 수 있습니다.

인터페이스 엔드포인트를 AWS Serverless Application Repository 사용한 액세스(AWS PrivateLink)

AWS PrivateLink 를 사용하여 VPC와 간에 프라이빗 연결을 생성할 수 있습니다 AWS Serverless Application Repository. 인터넷 게이트웨이, NAT 디바이스, VPN 연결 또는 AWS Direct Connect 연결을 사용하지 않고 VPC에 있는 AWS Serverless Application Repository 것처럼에 액세스할 수 있습니다. VPC의 인스턴스에서 AWS Serverless Application Repository API에 액세스하는 데는 퍼블릭 IP 주소가 필요하지 않습니다.

AWS PrivateLink에서 제공되는 인터페이스 엔드포인트를 생성하여 이 프라이빗 연결을 설정합니다. 인터페이스 엔드포인트에 대해 사용 설정하는 각 서브넷에서 엔드포인트 네트워크 인터페이스를 생성합니다. 이는 AWS Serverless Application Repository로 향하는 트래픽의 진입점 역할을 하는 요청자 관리형 네트워크 인터페이스입니다.

자세한 내용은AWS PrivateLink 가이드의 AWS PrivateLink를 통해 AWS 서비스 에 액세스를 참조하세요.

에 대한 고려 사항 AWS Serverless Application Repository

에 대한 인터페이스 엔드포인트를 설정하기 전에 AWS PrivateLink 가이드의 고려 사항을 AWS Serverless Application Repository검토하세요.

AWS Serverless Application Repository 는 인터페이스 엔드포인트를 통해 모든 API 작업에 대한 호출을 지원합니다.

에 대한 인터페이스 엔드포인트 생성 AWS Serverless Application Repository

Amazon VPC 콘솔 또는 AWS Command Line Interface ()를 AWS Serverless Application Repository 사용하여 용 인터페이스 엔드포인트를 생성할 수 있습니다AWS CLI. 자세한 내용은 AWS PrivateLink 설명서의 인터페이스 엔드포인트 생성을 참조하세요.

다음 서비스 이름을 AWS Serverless Application Repository 사용하여 용 인터페이스 엔드포인트를 생성합니다.

```
com.amazonaws.region.serverlessrepo
```

인터페이스 엔드포인트에 프라이빗 DNS를 사용하도록 설정하는 경우, 리전에 대한 기본 DNS 이름(예: AWS Serverless Application Repository)을 사용하여 에 API 요청을 할 수 있습니다. 예: `serverlessrepo.us-east-1.amazonaws.com`.

엔드포인트의 엔드포인트 정책 생성

엔드포인트 정책은 인터페이스 인터페이스 엔드포인트에 연결할 수 있는 IAM 리소스입니다. 기본 엔드포인트 정책은 인터페이스 엔드포인트를 AWS Serverless Application Repository 통해에 대한 전체 액세스를 허용합니다. VPC AWS Serverless Application Repository 에서에 허용되는 액세스를 제어하려면 인터페이스 엔드포인트에 사용자 지정 엔드포인트 정책을 연결합니다.

엔드포인트 정책은 다음 정보를 지정합니다.

- 작업을 수행할 수 있는 보안 주체 (AWS 계정, IAM 사용자, IAM 역할)
- 수행할 수 있는 작업.
- 작업을 수행할 수 있는 리소스.

자세한 내용은AWS PrivateLink 가이드의 엔드포인트 정책을 사용하여 서비스에 대한 액세스 제어를 참조하세요.

예: AWS Serverless Application Repository 작업에 대한 VPC 엔드포인트 정책

다음은 사용자 지정 엔드포인트 정책의 예입니다. 이 정책을 인터페이스 엔드포인트에 연결하면 모든 리소스의 모든 보안 주체에 대해 나열된 AWS Serverless Application Repository 작업에 대한 액세스

권한을 부여합니다. 다음 예제에서는 모든 사용자에게 VPC 엔드포인트를 통해 애플리케이션을 생성할 수 있는 권한을 허용합니다.

```
{
   "Statement": [
      {
         "Principal": "*",
         "Effect": "Allow",
         "Action": [
            "serverlessrepo:CreateApplication"
         ],
         "Resource":"*"
      }
   ]
}
```

AWS Serverless Application Repository 할당량

AWS Serverless Application Repository 에는 각 AWS 리전에서 AWS 계정이 보유할 수 있는 퍼블릭 애플리케이션 수에 대한 할당량이 있습니다. 이 할당량은 리전별로 적용되며 늘릴 수 있습니다. 할당량 증대를 요청하려면 지원 센터 콘솔을 이용하세요.

리소스	기본 할당량
퍼블릭 애플리케이션(AWS 리전별 AWS 계정당)	100

코드 패키지 및 애플리케이션 정책에 사용할 수 있는 스토리지에는 다음 할당량이 적용됩니다. 이러한 할당량은 변경할 수 없습니다.

리소스	할당량
코드 패키지를 위한 무료 Amazon S3 스토리지(AWS 리전별 AWS 계정당)	5GB
애플리케이션 정책 길이	6,144자

문제 해결 AWS Serverless Application Repository

를 사용하면 애플리케이션을 생성 AWS Serverless Application Repository, 업데이트 또는 삭제할 때 문제가 발생할 수 있습니다. 이 단원을 사용하여 발생할 수 있는 일반적인 문제를 해결하십시오. AWS Serverless Application Repository 포럼에서 답을 검색하고 질문을 올릴 수도 있습니다.

> **Note**
>
> 의 애플리케이션 AWS Serverless Application Repository 은를 사용하여 배포됩니다 AWS CloudFormation. AWS CloudFormation 문제 해결에 대한 자세한 내용은 AWS CloudFormation 문제 해결 안내서를 참조하세요.

주제

애플리케이션을 공개로 만들 수 없음

애플리케이션을 공개로 만들 수 없는 경우 OSI(오픈 소스 이니셔티브)에서 승인한 애플리케이션에 대한 라이선스 파일이 누락되었을 수도 있습니다.

애플리케이션을 공개로 만들려면 OSI 승인 라이선스 파일과 해당 버전에 대한 소스 코드 URL이 포함된 애플리케이션의 게시된 버전이 필요합니다. 애플리케이션 생성 이후에는 애플리케이션의 라이선스를 업데이트할 수 없습니다.

애플리케이션을 공개로 만들 수 없는 경우 라이선스 파일이 누락되었고, 애플리케이션을 삭제하고 동일한 이름의 새 애플리케이션을 생성하기 때문입니다. OSI(오픈 소스 이니셔티브) 조직에서 승인한 하나 이상의 오픈 소스 라이선스를 제공해야 합니다.

할당량이 초과됨

할당량이 초과되었다는 오류 메시지를 수신한 경우 리소스 할당량에 도달했는지 확인하십시오. AWS Serverless Application Repository 할당량은 섹션을 참조하세요AWS Serverless Application Repository 할당량.

업데이트된 Readme 파일이 즉시 표시되지 않음

애플리케이션을 공개로 만들 때 애플리케이션의 내용을 업데이트하는 데 최대 24시간이 소요될 수 있습니다. 24시간 이상 지연되는 경우 AWS Support에 문의하여 도움을 받으세요. 자세한 내용은 다음을 참조하십시오.

충분하지 않은 IAM 권한으로 인해 애플리케이션을 배포할 수 없음

AWS Serverless Application Repository 애플리케이션을 배포하려면 AWS Serverless Application Repository 리소스 및 AWS CloudFormation 스택에 대한 권한이 필요합니다. 애플리케이션에서 설명하는 기본 서비스를 사용할 권한 또한 필요합니다. 예를 들어 Amazon S3 버킷 또는 Amazon DynamoDB 테이블을 생성하는 경우 Amazon S3 또는 DynamoDB에 대한 권한이 필요합니다.

이러한 유형의 문제가 발생하면 AWS Identity and Access Management (IAM) 정책을 검토하고 필요한 권한이 있는지 확인합니다. 자세한 내용은 AWS Identity and Access Management를 사용한 액세스 제어를 참조하세요.

동일한 애플리케이션을 두 번 배포할 수 없음

제공하는 애플리케이션 이름은 AWS CloudFormation 스택의 이름으로 사용됩니다. 애플리케이션을 배포하는 데 문제가 있는 경우 이름이 같은 기존 AWS CloudFormation 스택이 없는지 확인합니다. 그러한 경우 다른 애플리케이션 이름을 입력하거나 기존 스택을 제거하여 동일한 이름의 애플리케이션을 배포합니다.

애플리케이션을 공개적으로 사용할 수 없는 이유

애플리케이션은 기본적으로 비공개입니다. 애플리케이션을 공개로 만들려면 여기에서 다음 단계를 따릅니다.

Support에 문의

경우에 따라 이 단원 또는 AWS Serverless Application Repository 포럼에서 문제 해결 솔루션을 찾지 못할 수 있습니다. AWS 프리미엄 지원이 있는 경우 AWS Support에서 기술 지원 사례를 생성할 수 있습니다.

AWS Support에 문의하기 전에 질문이 있는 애플리케이션의 Amazon 리소스 이름(ARN)을 얻어야 합니다. AWS Serverless Application Repository 콘솔에서 애플리케이션 ARN을 확인할 수 있습니다.

운영

AWS Serverless Application Repository REST API에는 다음 작업이 포함됩니다.

- CreateApplication

  동일한 호출에서 첫 번째 애플리케이션 버전을 생성하기 위해 선택적으로 AWS SAM 파일을 포함하는 애플리케이션을 생성합니다.
- CreateApplicationVersion

  애플리케이션 버전을 생성합니다.
- CreateCloudFormationChangeSet

  지정된 애플리케이션에 대한 AWS CloudFormation 변경 세트를 생성합니다.
- CreateCloudFormationTemplate

  AWS CloudFormation 템플릿을 생성합니다.
- DeleteApplication

  지정된 애플리케이션을 삭제합니다.
- GetApplication

  지정된 애플리케이션을 가져옵니다.
- GetApplicationPolicy

  애플리케이션에 대한 정책을 검색합니다.
- GetCloudFormationTemplate

  지정된 AWS CloudFormation 템플릿을 가져옵니다.
- ListApplicationDependencies

  포함된 애플리케이션에 중첩된 애플리케이션 목록을 검색합니다.
- ListApplications

  요청자가 소유한 애플리케이션을 나열합니다.
- ListApplicationVersions

  지정된 애플리케이션의 버전을 나열합니다.

- PutApplicationPolicy

  애플리케이션에 대한 권한 정책을 설정합니다. 이 작업에 지원되는 작업 목록은 애플리케이션 권한 섹션을 참조하세요.
- UnshareApplication

  AWS 조직에서 애플리케이션을 공유 해제합니다.

  이 작업은 조직의 관리 계정에서만 호출할 수 있습니다.
- UpdateApplication

  지정된 애플리케이션을 업데이트합니다.

리소스

AWS Serverless Application Repository REST API에는 다음 리소스가 포함되어 있습니다.

주제

Applications

URI

`/applications`

HTTP 메소드

GET

작업 ID: `ListApplications`

요청자가 소유한 애플리케이션을 나열합니다.

쿼리 파라미터

명칭	유형	필수	설명
`maxItems`	String	False	반환할 총 항목 수입니다.

명칭	유형	필수	설명
nextToken	String	False	페이지 매김을 시작할 위치를 지정하기 위한 토큰입니다.

응답

상태 코드	응답 모델	설명
200	ApplicationPage	Success
400	BadRequestException	요청의 파라미터 중 하나가 잘못되었습니다.
403	ForbiddenException	클라이언트가 인증되지 않았습니다.
404	NotFoundException	요청에 지정된 리소스(예: 액세스 정책 설명)가 존재하지 않습니다.
500	InternalServerErrorException	AWS Serverless Application Repository 서비스에서 내부 오류가 발생했습니다.

POST

작업 ID: CreateApplication

동일한 호출에서 첫 번째 애플리케이션 버전을 생성하기 위해 선택적으로 AWS SAM 파일을 포함하는 애플리케이션을 생성합니다.

응답

상태 코드	응답 모델	설명
201	Application	Success

상태 코드	응답 모델	설명
400	BadRequestException	요청의 파라미터 중 하나가 잘못되었습니다.
403	ForbiddenException	클라이언트가 인증되지 않았습니다.
409	ConflictException	리소스가 이미 존재합니다.
429	TooManyRequestsException	클라이언트가 시간 단위당 허용된 요청 수를 초과하여 전송하고 있습니다.
500	InternalServerErrorException	AWS Serverless Application Repository 서비스에서 내부 오류가 발생했습니다.

OPTIONS

응답

상태 코드	응답 모델	설명
200	None	응답 200개

스키마

요청 본문

POST 스키마

```
{
  "name": "string",
  "description": "string",
  "author": "string",
  "spdxLicenseId": "string",
  "licenseBody": "string",
  "licenseUrl": "string",
```

```
  "readmeBody": "string",
  "readmeUrl": "string",
  "labels": [
    "string"
  ],
  "homePageUrl": "string",
  "semanticVersion": "string",
  "templateBody": "string",
  "templateUrl": "string",
  "sourceCodeUrl": "string",
  "sourceCodeArchiveUrl": "string"
}
```

응답 본문

ApplicationPage 스키마

```
{
  "applications": [
    {
      "applicationId": "string",
      "name": "string",
      "description": "string",
      "author": "string",
      "spdxLicenseId": "string",
      "labels": [
        "string"
      ],
      "creationTime": "string",
      "homePageUrl": "string"
    }
  ],
  "nextToken": "string"
}
```

Application 스키마

```
{
  "applicationId": "string",
  "name": "string",
  "description": "string",
  "author": "string",
```

```
  "isVerifiedAuthor": boolean,
  "verifiedAuthorUrl": "string",
  "spdxLicenseId": "string",
  "licenseUrl": "string",
  "readmeUrl": "string",
  "labels": [
    "string"
  ],
  "creationTime": "string",
  "homePageUrl": "string",
  "version": {
    "applicationId": "string",
    "semanticVersion": "string",
    "sourceCodeUrl": "string",
    "sourceCodeArchiveUrl": "string",
    "templateUrl": "string",
    "creationTime": "string",
    "parameterDefinitions": [
      {
        "name": "string",
        "defaultValue": "string",
        "description": "string",
        "type": "string",
        "noEcho": boolean,
        "allowedPattern": "string",
        "constraintDescription": "string",
        "minValue": integer,
        "maxValue": integer,
        "minLength": integer,
        "maxLength": integer,
        "allowedValues": [
          "string"
        ],
        "referencedByResources": [
          "string"
        ]
      }
    ],
    "requiredCapabilities": [
      enum
    ],
    "resourcesSupported": boolean
  }
```

```
}
```

BadRequestException 스키마

```
{
  "message": "string",
  "errorCode": "string"
}
```

ForbiddenException 스키마

```
{
  "message": "string",
  "errorCode": "string"
}
```

NotFoundException 스키마

```
{
  "message": "string",
  "errorCode": "string"
}
```

ConflictException 스키마

```
{
  "message": "string",
  "errorCode": "string"
}
```

TooManyRequestsException 스키마

```
{
  "message": "string",
  "errorCode": "string"
}
```

InternalServerErrorException 스키마

```
{
  "message": "string",
  "errorCode": "string"
}
```

속성

Application

애플리케이션에 대한 세부 정보입니다.

applicationId

애플리케이션 Amazon 리소스 이름(ARN).

유형: 문자열
필수: True

name

애플리케이션의 이름입니다.

최소 길이: 1. 최대 길이=140

패턴: "[a-zA-Z0-9\\-]+";

유형: 문자열
필수: True

description

애플리케이션에 대한 설명입니다.

최소 길이: 1. 최대 길이=256

유형: 문자열
필수: True

author

앱을 게시하는 작성자의 이름입니다.

최소 길이: 1. 최대 길이: 127.

패턴 "^[a-z0-9](([a-z0-9]|-(?!-))*[a-z0-9])?$";

유형: 문자열
필수: True

isVerifiedAuthor

이 애플리케이션의 작성자가 확인되었는지 여부를 지정합니다. 즉, AWS 는 합리적이고 신중한 서비스 공급자로서 요청자가 제공한 정보를 성실하게 검토했으며 요청자의 자격 증명이 청구된 대로임을 확인했습니다.

유형: 부울
필수: False

verifiedAuthorUrl

확인된 작성자의 퍼블릭 프로필에 대한 URL입니다. 이 URL은 작성자가 제출합니다.

유형: 문자열
필수: False

spdxLicenseId

https://spdx.org/licenses/ 유효한 식별자입니다.

유형: 문자열
필수: False

licenseUrl

애플리케이션의 spdxLicenseID 값과 일치하는 앱의 라이선스 파일에 대한 링크입니다.

최대 크기 5MB

유형: 문자열
필수: False

readmeUrl

애플리케이션에 대한 자세한 설명과 작동 방식을 포함하는 마크다운 언어의 readme 파일에 대한 링크입니다.

최대 크기 5MB

유형: 문자열
필수: False

labels

검색 결과에서 앱 검색을 개선하기 위한 레이블입니다.

최소 길이: 1. 최대 길이: 127. 최대 레이블 수: 10

패턴: "^[a-zA-Z0-9+\\-_:\\/@]+$";

유형: 유형 string의 배열
필수: False

creationTime

이 리소스가 생성된 날짜 및 시간입니다.

유형: 문자열
필수: False

homePageUrl

애플리케이션에 대한 GitHub 리포지토리의 위치와 같은 애플리케이션에 대한 자세한 정보가 포함된 URL입니다.

유형: 문자열
필수: False

version

애플리케이션에 대한 버전 정보입니다.

유형: 버전
필수: 거짓

ApplicationPage

애플리케이션 세부 정보 목록입니다.

applications

애플리케이션 요약의 배열입니다.

유형: 유형 ApplicationSummary의 배열
필수: True

nextToken

결과의 다음 번 페이지를 요청하기 위한 토큰.

유형: 문자열
필수: False

ApplicationSummary

애플리케이션에 대한 세부 정보 요약입니다.

applicationId

애플리케이션 Amazon 리소스 이름(ARN).

유형: 문자열
필수: True

name

애플리케이션의 이름입니다.

최소 길이: 1. 최대 길이=140

패턴: "[a-zA-Z0-9\\-]+";

유형: 문자열
필수: True

description

애플리케이션에 대한 설명입니다.

최소 길이: 1. 최대 길이=256

유형: 문자열
필수: True

author

앱을 게시하는 작성자의 이름입니다.

최소 길이: 1. 최대 길이: 127.

패턴 "^[a-z0-9](([a-z0-9]|-(?!-))*[a-z0-9])?$";

유형: 문자열
필수: True

spdxLicenseId

https://spdx.org/licenses/ 유효한 식별자입니다.

유형: 문자열
필수: False

labels

검색 결과에서 앱 검색을 개선하기 위한 레이블입니다.

최소 길이: 1. 최대 길이: 127. 최대 레이블 수: 10

패턴: "^[a-zA-Z0-9+\\-_:\\/@]+$";

유형: 유형 string의 배열
필수: False

creationTime

이 리소스가 생성된 날짜 및 시간입니다.

유형: 문자열
필수: False

homePageUrl

애플리케이션에 대한 GitHub 리포지토리의 위치와 같은 애플리케이션에 대한 자세한 정보가 포함된 URL입니다.

유형: 문자열
필수: False

BadRequestException

요청의 파라미터 중 하나가 잘못되었습니다.

message

요청의 파라미터 중 하나가 잘못되었습니다.

유형: 문자열
필수: False

errorCode

400

유형: 문자열
필수: False

Capability

일부 애플리케이션을 배포하기 위해 지정해야 하는 값입니다.

CAPABILITY_IAM
CAPABILITY_NAMED_IAM
CAPABILITY_AUTO_EXPAND
CAPABILITY_RESOURCE_POLICY

ConflictException

리소스가 이미 존재합니다.

message

리소스가 이미 존재합니다.

유형: 문자열
필수: False

errorCode

409

유형: 문자열
필수: False

CreateApplicationInput

애플리케이션 요청을 생성합니다.

name

게시하려는 애플리케이션의 이름입니다.

최소 길이: 1. 최대 길이=140

패턴: "[a-zA-Z0-9\\-]+";

유형: 문자열
필수: True

description

애플리케이션에 대한 설명입니다.

최소 길이: 1. 최대 길이=256

유형: 문자열
필수: True

author

앱을 게시하는 작성자의 이름입니다.

최소 길이: 1. 최대 길이: 127.

패턴 "^[a-z0-9](([a-z0-9]|-(?!-))*[a-z0-9])?$";

유형: 문자열
필수: True

spdxLicenseId

https://spdx.org/licenses/ 유효한 식별자입니다.

유형: 문자열
필수: False

licenseBody

애플리케이션의 spdxLicenseID 값과 일치하는 앱의 라이선스가 포함된 로컬 텍스트 파일입니다. 파일의 형식은 입니다`file://<path>/<filename>`.

최대 크기 5MB

`licenseBody` 및 중 하나만 지정할 수 있습니다. 그렇지 `licenseUrl`않으면 오류가 발생합니다.

유형: 문자열
필수: False

licenseUrl

애플리케이션의 spdxLicenseID 값과 일치하는 앱의 라이선스가 포함된 S3 객체에 대한 링크입니다.

최대 크기 5MB

`licenseBody` 및 중 하나만 지정할 수 있습니다. 그렇지 `licenseUrl`않으면 오류가 발생합니다.

유형: 문자열
필수: False

readmeBody

애플리케이션에 대한 자세한 설명과 작동 방식을 포함하는 마크다운 언어의 로컬 텍스트 readme 파일입니다. 파일의 형식은 입니다`file://<path>/<filename>`.

최대 크기 5MB

`readmeBody` 및 중 하나만 지정할 수 있습니다. 그렇지 `readmeUrl`않으면 오류가 발생합니다.

유형: 문자열
필수: False

readmeUrl

애플리케이션에 대한 자세한 설명과 작동 방식을 포함하는 마크다운 언어의 S3 객체에 대한 링크입니다.

최대 크기 5MB

`readmeBody` 및 중 하나만 지정할 수 있습니다. 그렇지 `readmeUrl`않으면 오류가 발생합니다.

유형: 문자열
필수: False

labels

검색 결과에서 앱 검색을 개선하기 위한 레이블입니다.

최소 길이: 1. 최대 길이: 127. 최대 레이블 수: 10

패턴: "^[a-zA-Z0-9+\\-_:\\/@]+$";

유형: 유형 string의 배열
필수: False

homePageUrl

애플리케이션에 대한 GitHub 리포지토리의 위치와 같은 애플리케이션에 대한 자세한 정보가 포함된 URL입니다.

유형: 문자열
필수: False

semanticVersion

애플리케이션의 의미 체계 버전:

https://semver.org/

유형: 문자열
필수: False

templateBody

애플리케이션의 로컬 원시 패키지 AWS SAM 템플릿 파일입니다. 파일의 형식은 입니다`file://<path>/<filename>`.

`templateBody` 및 중 하나만 지정할 수 있습니다. 그렇지 `templateUrl`않으면 오류가 발생합니다.

유형: 문자열
필수: False

templateUrl

애플리케이션의 패키징된 AWS SAM 템플릿이 포함된 S3 객체에 대한 링크입니다.

`templateBody` 및 중 하나만 지정할 수 있습니다. 그렇지 `templateUrl`않으면 오류가 발생합니다.

유형: 문자열
필수: False

sourceCodeUrl

특정 GitHub 커밋의 URL과 같은 애플리케이션의 소스 코드에 대한 퍼블릭 리포지토리에 대한 링크입니다.

유형: 문자열
필수: False

sourceCodeArchiveUrl

이 버전의 애플리케이션에 대한 소스 코드의 ZIP 아카이브가 포함된 S3 객체에 대한 링크입니다.

최대 크기 50MB

유형: 문자열
필수: False

ForbiddenException

클라이언트가 인증되지 않았습니다.

message

클라이언트가 인증되지 않았습니다.

유형: 문자열
필수: False

errorCode

403

유형: 문자열
필수: False

InternalServerErrorException

AWS Serverless Application Repository 서비스에서 내부 오류가 발생했습니다.

message

AWS Serverless Application Repository 서비스에서 내부 오류가 발생했습니다.

유형: 문자열
필수: False

errorCode

500

유형: 문자열
필수: False

NotFoundException

요청에 지정된 리소스(예: 액세스 정책 설명)가 존재하지 않습니다.

message

요청에 지정된 리소스(예: 액세스 정책 설명)가 존재하지 않습니다.

유형: 문자열
필수: False

errorCode

404

유형: 문자열
필수: False

ParameterDefinition

애플리케이션에서 지원하는 파라미터입니다.

name

파라미터의 이름입니다.

유형: 문자열
필수: True

defaultValue

스택 생성 시 지정된 값이 없는 경우에 사용할 템플릿에 적합한 유형의 값입니다. 파라미터에 대한 제약을 정의하는 경우 이러한 제약을 준수하는 값을 지정해야 합니다.

유형: 문자열
필수: False

description

파라미터를 설명하는 최대 4,000자의 문자열입니다.

유형: 문자열
필수: False

type

파라미터의 유형입니다.

유효값: `String | Number | List<Number> | CommaDelimitedList`

`String`: 리터럴 문자열입니다.

예를 들어 사용자는를 지정할 수 있습니다`"MyUserName"`.

`Number`: 정수 또는 float. AWS CloudFormation valid는 파라미터 값을 숫자로 확인합니다. 그러나 템플릿의 다른 곳에서 파라미터를 사용하는 경우(예: `Ref` 내장 함수 사용) 파라미터 값은 문자열이 됩니다.

예를 들어 사용자는를 지정할 수 있습니다`"8888"`.

`List<Number>`: 쉼표로 구분된 정수 또는 부동 소수점의 배열입니다. AWS CloudFormation 는 파라미터 값을 숫자로 검증합니다. 그러나 템플릿의 다른 곳에서 파라미터를 사용하는 경우(예: `Ref` 내장 함수 사용) 파라미터 값은 문자열 목록이 됩니다.

예를 들어 사용자는 "80,20"을 지정한 다음를 `Ref` 생성할 수 있습니다`["80","20"]`.

`CommaDelimitedList`: 쉼표로 구분된 리터럴 문자열의 배열입니다. 총 문자열 수는 총 쉼표 수보다 하나 더 많아야 합니다. 또한 각 멤버 문자열은 공백으로 잘립니다.

예를 들어 사용자는 "test,dev,prod"를 지정한 다음 결과를 로 `Ref` 지정할 수 있습니다`["test","dev","prod"]`.

유형: 문자열
필수: False

noEcho

스택을 설명하는 호출을 할 때마다 파라미터 값을 마스킹할지 여부입니다. 값을 true로 설정하면 파라미터 값이 별표(*****)로 마스킹됩니다.

유형: 부울
필수: False

allowedPattern

`String` 유형에 허용할 패턴을 나타내는 정규식입니다.

유형: 문자열
필수: False

constraintDescription

제약 위반 시 해당 제약을 설명하는 문자열입니다. 예를 들어 제약 설명이 없으면 `[A-Za-z0-9]+` 패턴이 허용된 파라미터에 사용자가 유효하지 않은 값을 지정할 때 다음과 같은 오류 메시지가 표시됩니다.

```
Malformed input-Parameter MyParameter must match pattern [A-Za-z0-9]+
```

"대문자와 소문자 및 숫자만 포함해야 합니다"와 같은 제약 조건 설명을 추가하면 다음과 같은 사용자 지정 오류 메시지를 표시할 수 있습니다.

```
Malformed input-Parameter MyParameter must contain only uppercase and
lowercase letters and numbers.
```

유형: 문자열
필수: False

minValue

`Number` 유형에 허용할 가장 작은 숫자 값을 결정하는 숫자 값입니다.

유형: 정수
필수: 거짓

maxValue

`Number` 유형에 허용할 가장 큰 숫자 값을 결정하는 숫자 값입니다.

유형: 정수
필수: 거짓

minLength

`String` 유형에 허용할 최소 문자 수를 결정하는 정수 값입니다.

유형: 정수
필수: 거짓

maxLength

`String` 유형에 허용할 최대 문자 수를 결정하는 정수 값입니다.

유형: 정수
필수: 거짓

allowedValues

파라미터에 허용되는 값 목록을 포함하는 어레이입니다.

유형: 유형 string의 배열
필수: False

referencedByResources

이 파라미터를 사용하는 AWS SAM 리소스 목록입니다.

유형: 유형 string의 배열
필수: True

TooManyRequestsException

클라이언트가 시간 단위당 허용된 요청 수를 초과하여 전송하고 있습니다.

message

클라이언트가 시간 단위당 허용된 요청 수를 초과하여 전송하고 있습니다.

유형: 문자열
필수: False

errorCode

429

유형: 문자열
필수: False

Version

애플리케이션 버전 세부 정보입니다.

applicationId

애플리케이션 Amazon 리소스 이름(ARN).

유형: 문자열
필수: True

semanticVersion

애플리케이션의 의미 체계 버전:

https://semver.org/

유형: 문자열

필수: True

sourceCodeUrl

특정 GitHub 커밋의 URL과 같은 애플리케이션의 소스 코드에 대한 퍼블릭 리포지토리에 대한 링크입니다.

유형: 문자열
필수: False

sourceCodeArchiveUrl

이 버전의 애플리케이션에 대한 소스 코드의 ZIP 아카이브가 포함된 S3 객체에 대한 링크입니다.

최대 크기 50MB

유형: 문자열
필수: False

templateUrl

애플리케이션의 패키징된 AWS SAM 템플릿에 대한 링크입니다.

유형: 문자열
필수: True

creationTime

이 리소스가 생성된 날짜 및 시간입니다.

유형: 문자열
필수: True

parameterDefinitions

애플리케이션에서 지원하는 파라미터 유형의 배열입니다.

유형: 유형 ParameterDefinition의 배열

필수: True

requiredCapabilities

특정 애플리케이션을 배포하기 전에 지정해야 하는 값 목록입니다. 일부 애플리케이션에는 새 AWS Identity and Access Management (IAM) 사용자를 생성하는 등 AWS 계정의 권한에 영향을 미칠 수 있는 리소스가 포함될 수 있습니다. 이러한 애플리케이션의 경우이 파라미터를 지정하여 기능을 명시적으로 승인해야 합니다.

유일하게 유효한 값은 `CAPABILITY_IAM`, `CAPABILITY_RESOURCE_POLICY`, `CAPABILITY_NAMED_IAM`및 입니다`CAPABILITY_AUTO_EXPAND`.

다음 리소스에는 `CAPABILITY_IAM` 또는 `CAPABILITY_NAMED_IAM`을 지정해야 합니다. AWS::IAM::Group, AWS::IAM::InstanceProfile, AWS::IAM::Policy, and AWS::IAM::Role. 애플리케이션에 IAM 리소스가 포함된 경우 `CAPABILITY_IAM` 또는 중 하나를 지정할 수 있습니다`CAPABILITY_NAMED_IAM`. 애플리케이션에 사용자 지정 이름을 가진 IAM 리소스가 포함되어 있는 경우 `CAPABILITY_NAMED_IAM`을 지정해야 합니다.

AWS`CAPABILITY_RESOURCE_POLICY`AWS::Lambda::Permission, AWS::IAM:Policy, AWS::ApplicationAutoScaling::ScalingPolicy, AWS::S3::BucketPolicy, AWS::SQS::QueuePolicy 및 AWS::SNS::TopicPolicy 리소스를 지정해야 합니다.

중첩 애플리케이션을 한 개 이상 포함하는 애플리케이션은 `CAPABILITY_AUTO_EXPAND`를 지정해야 합니다.

애플리케이션 템플릿에 위의 리소스가 포함된 경우 배포하기 전에 애플리케이션과 연결된 모든 권한을 검토하는 것이 좋습니다. 기능이 필요한 애플리케이션에 대해이 파라미터를 지정하지 않으면 호출이 실패합니다.

유형: 유형 Capability의 배열

필수: True

resourcesSupported

이 애플리케이션에 포함된 모든 AWS 리소스가 검색되는 리전에서 지원되는지 여부입니다.

유형: 부울

필수: True

다음 사항도 참조하세요.

언어별 AWS SDKs 및 참조 중 하나에서이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하세요.

ListApplications

- AWS 명령줄 인터페이스
- AWS SDK for .NET
- AWS SDK for C++
- Go v2용 AWS SDK
- AWS SDK for Java V2
- JavaScript V3용 AWS SDK
- Kotlin용 AWS SDK
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

CreateApplication

- AWS 명령줄 인터페이스
- AWS SDK for .NET
- AWS SDK for C++
- Go v2용 AWS SDK
- AWS SDK for Java V2
- JavaScript V3용 AWS SDK
- Kotlin용 AWS SDK
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

애플리케이션 applicationId

URI

`/applications/applicationId`

HTTP 메소드

GET

작업 ID: `GetApplication`

지정된 애플리케이션을 가져옵니다.

경로 파라미터

명칭	유형	필수	설명
applicationId	String	True	애플리케이션의 Amazon 리소스 이름(ARN)입니다.

쿼리 파라미터

명칭	유형	필수	설명
semanticVersion	String	False	가져올 애플리케이션의 의미 체계 버전입니다.

응답

상태 코드	응답 모델	설명
200	Application	Success
400	BadRequestException	요청의 파라미터 중 하나가 잘못되었습니다.

상태 코드	응답 모델	설명
403	ForbiddenException	클라이언트가 인증되지 않았습니다.
404	NotFoundException	요청에 지정된 리소스(예: 액세스 정책 설명)가 존재하지 않습니다.
429	TooManyRequestsException	클라이언트가 시간 단위당 허용된 요청 수를 초과하여 전송하고 있습니다.
500	InternalServerErrorException	AWS Serverless Application Repository 서비스에서 내부 오류가 발생했습니다.

DELETE

작업 ID: DeleteApplication

지정된 애플리케이션을 삭제합니다.

경로 파라미터

명칭	유형	필수	설명
applicationId	String	True	애플리케이션의 Amazon 리소스 이름(ARN)입니다.

응답

상태 코드	응답 모델	설명
204	None	Success
400	BadRequestException	요청의 파라미터 중 하나가 잘못되었습니다.

상태 코드	응답 모델	설명
403	ForbiddenException	클라이언트가 인증되지 않았습니다.
404	NotFoundException	요청에 지정된 리소스(예: 액세스 정책 설명)가 존재하지 않습니다.
409	ConflictException	리소스가 이미 존재합니다.
429	TooManyRequestsException	클라이언트가 시간 단위당 허용된 요청 수를 초과하여 전송하고 있습니다.
500	InternalServerErrorException	AWS Serverless Application Repository 서비스에서 내부 오류가 발생했습니다.

OPTIONS

경로 파라미터

명칭	유형	필수	설명
applicationId	String	True	애플리케이션의 Amazon 리소스 이름(ARN)입니다.

응답

상태 코드	응답 모델	설명
200	None	응답 200개

PATCH

작업 ID: UpdateApplication

지정된 애플리케이션을 업데이트합니다.

경로 파라미터

명칭	유형	필수	설명
applicationId	String	True	애플리케이션의 Amazon 리소스 이름(ARN)입니다.

응답

상태 코드	응답 모델	설명
200	Application	Success
400	BadRequestException	요청의 파라미터 중 하나가 잘못되었습니다.
403	ForbiddenException	클라이언트가 인증되지 않았습니다.
404	NotFoundException	요청에 지정된 리소스(예: 액세스 정책 설명)가 존재하지 않습니다.
409	ConflictException	리소스가 이미 존재합니다.
429	TooManyRequestsException	클라이언트가 시간 단위당 허용된 요청 수를 초과하여 전송하고 있습니다.
500	InternalServerErrorException	AWS Serverless Application Repository 서비스에서 내부 오류가 발생했습니다.

스키마

요청 본문

PATCH 스키마

```
{
  "description": "string",
  "author": "string",
  "readmeBody": "string",
  "readmeUrl": "string",
  "labels": [
    "string"
  ],
  "homePageUrl": "string"
}
```

응답 본문

Application 스키마

```
{
  "applicationId": "string",
  "name": "string",
  "description": "string",
  "author": "string",
  "isVerifiedAuthor": boolean,
  "verifiedAuthorUrl": "string",
  "spdxLicenseId": "string",
  "licenseUrl": "string",
  "readmeUrl": "string",
  "labels": [
    "string"
  ],
  "creationTime": "string",
  "homePageUrl": "string",
  "version": {
    "applicationId": "string",
    "semanticVersion": "string",
    "sourceCodeUrl": "string",
    "sourceCodeArchiveUrl": "string",
    "templateUrl": "string",
    "creationTime": "string",
    "parameterDefinitions": [
      {
        "name": "string",
        "defaultValue": "string",
```

```
        "description": "string",
        "type": "string",
        "noEcho": boolean,
        "allowedPattern": "string",
        "constraintDescription": "string",
        "minValue": integer,
        "maxValue": integer,
        "minLength": integer,
        "maxLength": integer,
        "allowedValues": [
          "string"
        ],
        "referencedByResources": [
          "string"
        ]
      }
    ],
    "requiredCapabilities": [
      enum
    ],
    "resourcesSupported": boolean
  }
}
```

BadRequestException 스키마

```
{
  "message": "string",
  "errorCode": "string"
}
```

ForbiddenException 스키마

```
{
  "message": "string",
  "errorCode": "string"
}
```

NotFoundException 스키마

```
{
```

```
  "message": "string",
  "errorCode": "string"
}
```

ConflictException 스키마

```
{
  "message": "string",
  "errorCode": "string"
}
```

TooManyRequestsException 스키마

```
{
  "message": "string",
  "errorCode": "string"
}
```

InternalServerErrorException 스키마

```
{
  "message": "string",
  "errorCode": "string"
}
```

속성

Application

애플리케이션에 대한 세부 정보입니다.

applicationId

애플리케이션 Amazon 리소스 이름(ARN).

유형: 문자열
필수: True

name

애플리케이션의 이름입니다.

최소 길이: 1. 최대 길이=140

패턴: "[a-zA-Z0-9\\-]+";

유형: 문자열
필수: True

description

애플리케이션에 대한 설명입니다.

최소 길이: 1. 최대 길이=256

유형: 문자열
필수: True

author

앱을 게시하는 작성자의 이름입니다.

최소 길이: 1. 최대 길이: 127.

패턴 "^[a-z0-9](([a-z0-9]|-(?!-))*[a-z0-9])?$";

유형: 문자열
필수: True

isVerifiedAuthor

이 애플리케이션의 작성자가 확인되었는지 여부를 지정합니다. 즉, AWS 는 합리적이고 신중한 서비스 공급자로서 요청자가 제공한 정보를 성실하게 검토했으며 요청자의 자격 증명이 청구된 대로임을 확인했습니다.

유형: 부울
필수: False

verifiedAuthorUrl

확인된 작성자의 퍼블릭 프로필에 대한 URL입니다. 이 URL은 작성자가 제출합니다.

유형: 문자열
필수: False

spdxLicenseId

https://spdx.org/licenses/ 유효한 식별자입니다.

유형: 문자열
필수: False

licenseUrl

애플리케이션의 spdxLicenseID 값과 일치하는 앱의 라이선스 파일에 대한 링크입니다.

최대 크기 5MB

유형: 문자열
필수: False

readmeUrl

애플리케이션에 대한 자세한 설명과 작동 방식을 포함하는 마크다운 언어의 readme 파일에 대한 링크입니다.

최대 크기 5MB

유형: 문자열
필수: False

labels

검색 결과에서 앱 검색을 개선하기 위한 레이블입니다.

최소 길이: 1. 최대 길이: 127. 최대 레이블 수: 10

패턴: "^[a-zA-Z0-9+\\-_:\\/@]+$";

유형: 유형 string의 배열
필수: False

creationTime

이 리소스가 생성된 날짜 및 시간입니다.

유형: 문자열
필수: False

homePageUrl

애플리케이션에 대한 GitHub 리포지토리의 위치와 같은 애플리케이션에 대한 자세한 정보가 포함된 URL입니다.

유형: 문자열
필수: False

version

애플리케이션에 대한 버전 정보입니다.

유형: 버전
필수: 거짓

BadRequestException

요청의 파라미터 중 하나가 잘못되었습니다.

message

요청의 파라미터 중 하나가 잘못되었습니다.

유형: 문자열
필수: False

errorCode

400

유형: 문자열
필수: False

Capability

일부 애플리케이션을 배포하기 위해 지정해야 하는 값입니다.

```
CAPABILITY_IAM
CAPABILITY_NAMED_IAM
CAPABILITY_AUTO_EXPAND
CAPABILITY_RESOURCE_POLICY
```

ConflictException

리소스가 이미 존재합니다.

message

리소스가 이미 존재합니다.

유형: 문자열
필수: False

errorCode

409

유형: 문자열
필수: False

ForbiddenException

클라이언트가 인증되지 않았습니다.

message

클라이언트가 인증되지 않았습니다.

유형: 문자열
필수: False

errorCode

403

유형: 문자열
필수: False

InternalServerErrorException

AWS Serverless Application Repository 서비스에서 내부 오류가 발생했습니다.

message

AWS Serverless Application Repository 서비스에서 내부 오류가 발생했습니다.

유형: 문자열
필수: False

errorCode

500

유형: 문자열
필수: False

NotFoundException

요청에 지정된 리소스(예: 액세스 정책 설명)가 존재하지 않습니다.

message

요청에 지정된 리소스(예: 액세스 정책 설명)가 존재하지 않습니다.

유형: 문자열
필수: False

errorCode

404

유형: 문자열

필수: False

ParameterDefinition

애플리케이션에서 지원하는 파라미터입니다.

name

파라미터의 이름입니다.

유형: 문자열
필수: True

defaultValue

스택 생성 시 지정된 값이 없는 경우에 사용할 템플릿에 적합한 유형의 값입니다. 파라미터에 대한 제약을 정의하는 경우 이러한 제약을 준수하는 값을 지정해야 합니다.

유형: 문자열
필수: False

description

파라미터를 설명하는 최대 4,000자의 문자열입니다.

유형: 문자열
필수: False

type

파라미터의 유형입니다.

유효값: `String | Number | List<Number> | CommaDelimitedList`

`String`: 리터럴 문자열입니다.

예를 들어 사용자는를 지정할 수 있습니다`"MyUserName"`.

`Number`: 정수 또는 float. AWS CloudFormation valid는 파라미터 값을 숫자로 확인합니다. 그러나 템플릿의 다른 곳에서 파라미터를 사용하는 경우(예: `Ref` 내장 함수 사용) 파라미터 값은 문자열이 됩니다.

예를 들어 사용자는를 지정할 수 있습니다"8888".

List<Number>: 쉼표로 구분된 정수 또는 부동 소수점의 배열입니다. AWS CloudFormation 는 파라미터 값을 숫자로 검증합니다. 그러나 템플릿의 다른 곳에서 파라미터를 사용하는 경우(예: Ref 내장 함수 사용) 파라미터 값은 문자열 목록이 됩니다.

예를 들어 사용자는 "80,20"을 지정한 다음를 Ref 생성할 수 있습니다["80","20"].

CommaDelimitedList: 쉼표로 구분된 리터럴 문자열의 배열입니다. 총 문자열 수는 총 쉼표 수보다 하나 더 많아야 합니다. 또한 각 멤버 문자열은 공백으로 잘립니다.

예를 들어 사용자는 "test,dev,prod"를 지정한 다음 결과를 로 Ref 지정할 수 있습니다["test","dev","prod"].

유형: 문자열
필수: False

noEcho

스택을 설명하는 호출을 할 때마다 파라미터 값을 마스킹할지 여부입니다. 값을 true로 설정하면 파라미터 값이 별표(*****)로 마스킹됩니다.

유형: 부울
필수: False

allowedPattern

String 유형에 허용할 패턴을 나타내는 정규식입니다.

유형: 문자열
필수: False

constraintDescription

제약 위반 시 해당 제약을 설명하는 문자열입니다. 예를 들어 제약 설명이 없으면 [A-Za-z0-9]+ 패턴이 허용된 파라미터에 사용자가 유효하지 않은 값을 지정할 때 다음과 같은 오류 메시지가 표시됩니다.

```
Malformed input-Parameter MyParameter must match pattern [A-Za-z0-9]+
```

"대문자와 소문자 및 숫자만 포함해야 합니다"와 같은 제약 조건 설명을 추가하면 다음과 같은 사용자 지정 오류 메시지를 표시할 수 있습니다.

```
Malformed input-Parameter MyParameter must contain only uppercase and
lowercase letters and numbers.
```

유형: 문자열
필수: False

minValue

`Number` 유형에 허용할 가장 작은 숫자 값을 결정하는 숫자 값입니다.

유형: 정수
필수: 거짓

maxValue

`Number` 유형에 허용할 가장 큰 숫자 값을 결정하는 숫자 값입니다.

유형: 정수
필수: 거짓

minLength

`String` 유형에 허용할 최소 문자 수를 결정하는 정수 값입니다.

유형: 정수
필수: 거짓

maxLength

`String` 유형에 허용할 최대 문자 수를 결정하는 정수 값입니다.

유형: 정수
필수: 거짓

allowedValues

파라미터에 허용되는 값 목록을 포함하는 어레이입니다.

유형: 유형 string의 배열
필수: False

referencedByResources

이 파라미터를 사용하는 AWS SAM 리소스 목록입니다.

유형: 유형 string의 배열
필수: True

TooManyRequestsException

클라이언트가 시간 단위당 허용된 요청 수를 초과하여 전송하고 있습니다.

message

클라이언트가 시간 단위당 허용된 요청 수를 초과하여 전송하고 있습니다.

유형: 문자열
필수: False

errorCode

429

유형: 문자열
필수: False

UpdateApplicationInput

애플리케이션 요청을 업데이트합니다.

description

애플리케이션에 대한 설명입니다.

최소 길이: 1. 최대 길이=256

유형: 문자열
필수: False

author

앱을 게시하는 작성자의 이름입니다.

최소 길이: 1. 최대 길이: 127.

패턴 "^[a-z0-9](([a-z0-9]|-(?!-))*[a-z0-9])?$";

유형: 문자열
필수: False

readmeBody

애플리케이션에 대한 자세한 설명과 작동 방식을 포함하는 마크다운 언어의 텍스트 readme 파일입니다.

최대 크기 5MB

유형: 문자열
필수: False

readmeUrl

애플리케이션에 대한 자세한 설명과 작동 방식을 포함하는 마크다운 언어의 readme 파일에 대한 링크입니다.

최대 크기 5MB

유형: 문자열
필수: False

labels

검색 결과에서 앱 검색을 개선하기 위한 레이블입니다.

최소 길이: 1. 최대 길이: 127. 최대 레이블 수: 10

패턴: "^[a-zA-Z0-9+\\-_:\\/@]+$";

유형: 유형 string의 배열
필수: False

homePageUrl

애플리케이션에 대한 GitHub 리포지토리의 위치와 같은 애플리케이션에 대한 자세한 정보가 포함된 URL입니다.

유형: 문자열
필수: False

Version

애플리케이션 버전 세부 정보입니다.

applicationId

애플리케이션 Amazon 리소스 이름(ARN).

유형: 문자열
필수: True

semanticVersion

애플리케이션의 의미 체계 버전:

https://semver.org/

유형: 문자열
필수: True

sourceCodeUrl

특정 GitHub 커밋의 URL과 같은 애플리케이션의 소스 코드에 대한 퍼블릭 리포지토리에 대한 링크입니다.

유형: 문자열
필수: False

sourceCodeArchiveUrl

이 버전의 애플리케이션에 대한 소스 코드의 ZIP 아카이브가 포함된 S3 객체에 대한 링크입니다.

최대 크기 50MB

유형: 문자열
필수: False

templateUrl

애플리케이션의 패키징된 AWS SAM 템플릿에 대한 링크입니다.

유형: 문자열
필수: True

creationTime

이 리소스가 생성된 날짜와 시간입니다.

유형: 문자열
필수: True

parameterDefinitions

애플리케이션에서 지원하는 파라미터 유형의 배열입니다.

유형: 유형 ParameterDefinition의 배열
필수: True

requiredCapabilities

특정 애플리케이션을 배포하기 전에 지정해야 하는 값 목록입니다. 일부 애플리케이션에는 예를 들어 새 AWS Identity and Access Management (IAM) 사용자를 생성하여 AWS 계정의 권한에 영향을 미칠 수 있는 리소스가 포함될 수 있습니다. 이러한 애플리케이션의 경우이 파라미터를 지정하여 해당 기능을 명시적으로 승인해야 합니다.

유일하게 유효한 값은 CAPABILITY_IAM, CAPABILITY_RESOURCE_POLICY, CAPABILITY_NAMED_IAM및 입니다CAPABILITY_AUTO_EXPAND.

다음 리소스에는 CAPABILITY_IAM 또는 CAPABILITY_NAMED_IAM을 지정해야 합니다. AWS::IAM::Group, AWS::IAM::InstanceProfile, AWS::IAM::Policy, and AWS::IAM::Role. 애플리케이션에 IAM 리소스가 포함된 경우 CAPABILITY_IAM 또는 중 하나를 지정할 수 있습니다CAPABILITY_NAMED_IAM. 애플리케이션에 사용자 지정 이름을 가진 IAM 리소스가 포함되어 있는 경우 CAPABILITY_NAMED_IAM을 지정해야 합니다.

AWSCAPABILITY_RESOURCE_POLICYAWS::Lambda::Permission, AWS::IAM:Policy, AWS::ApplicationAutoScaling::ScalingPolicy, AWS::S3::BucketPolicy, AWS::SQS::QueuePolicy 및 AWS::SNS::TopicPolicy 리소스를 지정해야 합니다.

중첩 애플리케이션을 한 개 이상 포함하는 애플리케이션은 CAPABILITY_AUTO_EXPAND를 지정해야 합니다.

애플리케이션 템플릿에 위의 리소스가 포함된 경우 배포하기 전에 애플리케이션과 연결된 모든 권한을 검토하는 것이 좋습니다. 기능이 필요한 애플리케이션에 대해이 파라미터를 지정하지 않으면 호출이 실패합니다.

유형: 유형 Capability의 배열
필수: True

resourcesSupported

이 애플리케이션에 포함된 모든 AWS 리소스가 검색되는 리전에서 지원되는지 여부입니다.

유형: 부울
필수: True

다음 사항도 참조하세요.

언어별 AWS SDKs 및 참조 중 하나에서이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하세요.

GetApplication

- AWS 명령줄 인터페이스
- AWS SDK for .NET

- AWS SDK for C++
- Go v2용 AWS SDK
- AWS SDK for Java V2
- JavaScript V3용 AWS SDK
- Kotlin용 AWS SDK
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DeleteApplication

- AWS 명령줄 인터페이스
- AWS SDK for .NET
- AWS SDK for C++
- Go v2용 AWS SDK
- AWS SDK for Java V2
- JavaScript V3용 AWS SDK
- Kotlin용 AWS SDK
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

UpdateApplication

- AWS 명령줄 인터페이스
- AWS SDK for .NET
- AWS SDK for C++
- Go v2용 AWS SDK
- AWS SDK for Java V2
- JavaScript V3용 AWS SDK
- Kotlin용 AWS SDK

- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

애플리케이션 applicationId 변경 세트

URI

`/applications/applicationId/changesets`

HTTP 메소드

POST

작업 ID: `CreateCloudFormationChangeSet`

지정된 애플리케이션에 대한 AWS CloudFormation 변경 세트를 생성합니다.

경로 파라미터

명칭	유형	필수	설명
applicationId	String	True	애플리케이션의 Amazon 리소스 이름(ARN)입니다.

응답

상태 코드	응답 모델	설명
201	ChangeSetDetails	Success
400	BadRequestException	요청의 파라미터 중 하나가 잘못되었습니다.
403	ForbiddenException	클라이언트가 인증되지 않았습니다.

상태 코드	응답 모델	설명
429	TooManyRequestsException	클라이언트가 시간 단위당 허용된 요청 수를 초과하여 전송하고 있습니다.
500	InternalServerErrorException	AWS Serverless Application Repository 서비스에서 내부 오류가 발생했습니다.

OPTIONS

경로 파라미터

명칭	유형	필수	설명
applicationId	String	True	애플리케이션의 Amazon 리소스 이름(ARN)입니다.

응답

상태 코드	응답 모델	설명
200	None	응답 200개

스키마

요청 본문

POST 스키마

```
{
  "stackName": "string",
  "semanticVersion": "string",
  "templateId": "string",
  "parameterOverrides": [
    {
```

```
      "name": "string",
      "value": "string"
    }
  ],
  "capabilities": [
    "string"
  ],
  "changeSetName": "string",
  "clientToken": "string",
  "description": "string",
  "notificationArns": [
    "string"
  ],
  "resourceTypes": [
    "string"
  ],
  "rollbackConfiguration": {
    "rollbackTriggers": [
      {
        "arn": "string",
        "type": "string"
      }
    ],
    "monitoringTimeInMinutes": integer
  },
  "tags": [
    {
      "key": "string",
      "value": "string"
    }
  ]
}
```

응답 본문

ChangeSetDetails 스키마

```
{
  "applicationId": "string",
  "semanticVersion": "string",
  "changeSetId": "string",
  "stackId": "string"
}
```

BadRequestException 스키마

```
{
  "message": "string",
  "errorCode": "string"
}
```

ForbiddenException 스키마

```
{
  "message": "string",
  "errorCode": "string"
}
```

TooManyRequestsException 스키마

```
{
  "message": "string",
  "errorCode": "string"
}
```

InternalServerErrorException 스키마

```
{
  "message": "string",
  "errorCode": "string"
}
```

속성

BadRequestException

요청의 파라미터 중 하나가 잘못되었습니다.

message

요청의 파라미터 중 하나가 잘못되었습니다.

유형: 문자열
필수: False

errorCode

400

유형: 문자열
필수: False

ChangeSetDetails

변경 세트의 세부 정보입니다.

applicationId

애플리케이션 Amazon 리소스 이름(ARN).

유형: 문자열
필수: True

semanticVersion

애플리케이션의 의미 체계 버전:

https://semver.org/

유형: 문자열
필수: True

changeSetId

변경 세트의 Amazon 리소스 이름(ARN)입니다.

길이 제한: 최소 길이는 1입니다.

패턴: ARN:[-a-zA-Z0-9:/]*

유형: 문자열
필수: True

stackId

스택의 고유 ID입니다.

유형: 문자열
필수: True

CreateCloudFormationChangeSetInput

애플리케이션 변경 세트 요청을 생성합니다.

stackName

이 속성은 AWS CloudFormation CreateChangeSet API에 대해 동일한 이름의 파라미터에 해당합니다.

유형: 문자열
필수: True

semanticVersion

애플리케이션의 의미 체계 버전:

https://semver.org/

유형: 문자열
필수: False

templateId

CreateCloudFormationTemplate에서 반환한 UUID입니다.

패턴: [0-9a-fA-F]{8}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{12}

유형: 문자열
필수: False

parameterOverrides

애플리케이션의 파라미터에 대한 파라미터 값 목록입니다.

유형: 유형 ParameterValue의 배열
필수: False

capabilities

특정 애플리케이션을 배포하기 전에 지정해야 하는 값 목록입니다. 일부 애플리케이션에는 예를 들어 새 AWS Identity and Access Management (IAM) 사용자를 생성하여 AWS 계정의 권한에 영향을 미칠 수 있는 리소스가 포함될 수 있습니다. 이러한 애플리케이션의 경우이 파라미터를 지정하여 해당 기능을 명시적으로 승인해야 합니다.

유일하게 유효한 값은 CAPABILITY_IAM, CAPABILITY_RESOURCE_POLICY, CAPABILITY_NAMED_IAM및 입니다CAPABILITY_AUTO_EXPAND.

다음 리소스에는 CAPABILITY_IAM 또는 CAPABILITY_NAMED_IAM을 지정해야 합니다. AWS::IAM::Group, AWS::IAM::InstanceProfile, AWS::IAM::Policy, and AWS::IAM::Role. 애플리케이션에 IAM 리소스가 포함된 경우 CAPABILITY_IAM 또는 중 하나를 지정할 수 있습니다CAPABILITY_NAMED_IAM. 애플리케이션에 사용자 지정 이름을 가진 IAM 리소스가 포함되어 있는 경우 CAPABILITY_NAMED_IAM을 지정해야 합니다.

AWSCAPABILITY_RESOURCE_POLICYAWS::Lambda::Permission, AWS::IAM:Policy, AWS::ApplicationAutoScaling::ScalingPolicy, AWS::S3::BucketPolicy, AWS::SQS::QueuePolicy 및 AWS::SNS:TopicPolicy 리소스를 지정해야 합니다.

중첩 애플리케이션을 한 개 이상 포함하는 애플리케이션은 CAPABILITY_AUTO_EXPAND를 지정해야 합니다.

애플리케이션 템플릿에 위의 리소스가 포함된 경우 배포하기 전에 애플리케이션과 연결된 모든 권한을 검토하는 것이 좋습니다. 기능이 필요한 애플리케이션에 대해이 파라미터를 지정하지 않으면 호출이 실패합니다.

유형: 유형 string의 배열
필수: False

changeSetName

이 속성은 AWS CloudFormation CreateChangeSet API에 대해 동일한 이름의 파라미터에 해당합니다.

유형: 문자열
필수: False

clientToken

이 속성은 AWS CloudFormation CreateChangeSet API에 대해 동일한 이름의 파라미터에 해당합니다.

유형: 문자열
필수: False

description

이 속성은 AWS CloudFormation CreateChangeSet API에 대해 동일한 이름의 파라미터에 해당합니다.

유형: 문자열
필수: False

notificationArns

이 속성은 AWS CloudFormation CreateChangeSet API에 대해 동일한 이름의 파라미터에 해당합니다.

유형: 유형 string의 배열
필수: False

resourceTypes

이 속성은 AWS CloudFormation CreateChangeSet API에 대해 동일한 이름의 파라미터에 해당합니다.

유형: 유형 string의 배열
필수: False

rollbackConfiguration

이 속성은 AWS CloudFormation CreateChangeSet API에 대해 동일한 이름의 파라미터에 해당합니다.

유형: RollbackConfiguration
필수: 거짓

tags

이 속성은 AWS CloudFormation CreateChangeSet API에 대해 동일한 이름의 파라미터에 해당합니다.

유형: 유형 Tag의 배열
필수: False

ForbiddenException

클라이언트가 인증되지 않았습니다.

message

클라이언트가 인증되지 않았습니다.

유형: 문자열
필수: False

errorCode

403

유형: 문자열
필수: False

InternalServerErrorException

AWS Serverless Application Repository 서비스에서 내부 오류가 발생했습니다.

message

AWS Serverless Application Repository 서비스에서 내부 오류가 발생했습니다.

유형: 문자열
필수: False

errorCode

500

유형: 문자열
필수: False

ParameterValue

애플리케이션의 파라미터 값입니다.

name

파라미터와 연결된 키입니다. 특정 파라미터에 키와 값을 지정하지 않으면는 템플릿에 지정된 기본값을 AWS CloudFormation 사용합니다.

유형: 문자열
필수: True

value

파라미터와 연결된 입력 값입니다.

유형: 문자열
필수: True

RollbackConfiguration

이 속성은 AWS CloudFormation RollbackConfiguration 데이터 형식에 해당합니다.

rollbackTriggers

이 속성은 AWS CloudFormation RollbackConfiguration 데이터 형식에 대해 동일한 이름의 콘텐츠에 해당합니다.

유형: 유형 RollbackTrigger의 배열
필수: False

monitoringTimeInMinutes

이 속성은 AWS CloudFormation RollbackConfiguration 데이터 형식에 대해 동일한 이름의 콘텐츠에 해당합니다.

유형: 정수
필수: 거짓

RollbackTrigger

이 속성은 AWS CloudFormation RollbackTrigger 데이터 형식에 해당합니다.

arn

이 속성은 AWS CloudFormation RollbackTrigger 데이터 유형에 대해 동일한 이름의 콘텐츠에 해당합니다.

유형: 문자열
필수: True

type

이 속성은 AWS CloudFormation RollbackTrigger 데이터 형식에 대해 동일한 이름의 콘텐츠에 해당합니다.

유형: 문자열
필수: True

Tag

이 속성은 AWS CloudFormation 태그 데이터 유형에 해당합니다.

key

이 속성은 AWS CloudFormation 태그 데이터 유형에 대해 동일한 이름의 콘텐츠에 해당합니다.

유형: 문자열
필수: True

value

이 속성은 AWS CloudFormation 태그 데이터 유형에 대해 동일한 이름의 콘텐츠에 해당합니다.

유형: 문자열

필수: True

TooManyRequestsException

클라이언트가 시간 단위당 허용된 요청 수를 초과하여 전송하고 있습니다.

message

클라이언트가 시간 단위당 허용된 요청 수를 초과하여 전송하고 있습니다.

유형: 문자열
필수: False

errorCode

429

유형: 문자열
필수: False

다음 사항도 참조하세요.

언어별 AWS SDKs 및 참조 중 하나에서이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하세요.

CreateCloudFormationChangeSet

- AWS 명령줄 인터페이스
- AWS SDK for .NET
- AWS SDK for C++
- Go v2용 AWS SDK
- AWS SDK for Java V2
- JavaScript V3용 AWS SDK
- Kotlin용 AWS SDK
- AWS SDK for PHP V3
- AWS SDK for Python

- AWS SDK for Ruby V3

Applications applicationId Dependencies

URI

`/applications/applicationId/dependencies`

HTTP 메소드

GET

작업 ID: `ListApplicationDependencies`

포함된 애플리케이션에 중첩된 애플리케이션 목록을 검색합니다.

경로 파라미터

명칭	유형	필수	설명
applicationId	String	True	애플리케이션의 Amazon 리소스 이름(ARN)입니다.

쿼리 파라미터

명칭	유형	필수	설명
nextToken	String	False	페이지 매김을 시작할 위치를 지정하기 위한 토큰입니다.
maxItems	String	False	반환할 총 항목 수입니다.
semanticVersion	String	False	가져올 애플리케이션의 의미 체계 버전입니다.

응답

상태 코드	응답 모델	설명
200	ApplicationDependencyPage	Success
400	BadRequestException	요청의 파라미터 중 하나가 잘못되었습니다.
403	ForbiddenException	클라이언트가 인증되지 않았습니다.
404	NotFoundException	요청에 지정된 리소스(예: 액세스 정책 설명)가 존재하지 않습니다.
429	TooManyRequestsException	클라이언트가 시간 단위당 허용된 요청 수를 초과하여 전송하고 있습니다.
500	InternalServerErrorException	AWS Serverless Application Repository 서비스에서 내부 오류가 발생했습니다.

OPTIONS

경로 파라미터

명칭	유형	필수	설명
applicationId	String	True	애플리케이션의 Amazon 리소스 이름(ARN)입니다.

응답

상태 코드	응답 모델	설명
200	None	응답 200개

스키마

응답 본문

ApplicationDependencyPage 스키마

```
{
  "dependencies": [
    {
      "applicationId": "string",
      "semanticVersion": "string"
    }
  ],
  "nextToken": "string"
}
```

BadRequestException 스키마

```
{
  "message": "string",
  "errorCode": "string"
}
```

ForbiddenException 스키마

```
{
  "message": "string",
  "errorCode": "string"
}
```

NotFoundException 스키마

```
{
```

```
  "message": "string",
  "errorCode": "string"
}
```

TooManyRequestsException 스키마

```
{
  "message": "string",
  "errorCode": "string"
}
```

InternalServerErrorException 스키마

```
{
  "message": "string",
  "errorCode": "string"
}
```

속성

ApplicationDependencyPage

애플리케이션에 중첩된 애플리케이션 요약 목록입니다.

dependencies

애플리케이션에 중첩된 애플리케이션 요약 배열입니다.

유형: 유형 ApplicationDependencySummary의 배열
필수: True

nextToken

결과의 다음 번 페이지를 요청하기 위한 토큰.

유형: 문자열
필수: False

ApplicationDependencySummary

중첩된 애플리케이션 요약입니다.

applicationId

중첩된 애플리케이션의 Amazon 리소스 이름(ARN)입니다.

유형: 문자열
필수: True

semanticVersion

중첩 애플리케이션의 의미 체계 버전입니다.

유형: 문자열
필수: True

BadRequestException

요청의 파라미터 중 하나가 잘못되었습니다.

message

요청의 파라미터 중 하나가 잘못되었습니다.

유형: 문자열
필수: False

errorCode

400

유형: 문자열
필수: False

ForbiddenException

클라이언트가 인증되지 않았습니다.

message

클라이언트가 인증되지 않았습니다.

유형: 문자열
필수: False

errorCode

403

유형: 문자열
필수: False

InternalServerErrorException

AWS Serverless Application Repository 서비스에서 내부 오류가 발생했습니다.

message

AWS Serverless Application Repository 서비스에서 내부 오류가 발생했습니다.

유형: 문자열
필수: False

errorCode

500

유형: 문자열
필수: False

NotFoundException

요청에 지정된 리소스(예: 액세스 정책 설명)가 존재하지 않습니다.

message

요청에 지정된 리소스(예: 액세스 정책 설명)가 존재하지 않습니다.

유형: 문자열

필수: False

errorCode

404

유형: 문자열
필수: False

TooManyRequestsException

클라이언트가 시간 단위당 허용된 요청 수를 초과하여 전송하고 있습니다.

message

클라이언트가 시간 단위당 허용된 요청 수를 초과하여 전송하고 있습니다.

유형: 문자열
필수: False

errorCode

429

유형: 문자열
필수: False

다음 사항도 참조하세요.

언어별 AWS SDKs 및 참조 중 하나에서이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하세요.

ListApplicationDependencies

- AWS 명령줄 인터페이스
- AWS SDK for .NET
- AWS SDK for C++
- Go v2용 AWS SDK

- AWS SDK for Java V2
- JavaScript V3용 AWS SDK
- Kotlin용 AWS SDK
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

애플리케이션 applicationId 정책

URI

`/applications/applicationId/policy`

HTTP 메소드

GET

작업 ID: `GetApplicationPolicy`

애플리케이션에 대한 정책을 검색합니다.

경로 파라미터

명칭	유형	필수	설명
applicationId	String	True	애플리케이션의 Amazon 리소스 이름(ARN)입니다.

응답

상태 코드	응답 모델	설명
200	ApplicationPolicy	Success
400	BadRequestException	요청의 파라미터 중 하나가 잘못되었습니다.

상태 코드	응답 모델	설명
403	ForbiddenException	클라이언트가 인증되지 않았습니다.
404	NotFoundException	요청에 지정된 리소스(예: 액세스 정책 설명)가 존재하지 않습니다.
429	TooManyRequestsException	클라이언트가 시간 단위당 허용된 요청 수를 초과하여 전송하고 있습니다.
500	InternalServerErrorException	AWS Serverless Application Repository 서비스에서 내부 오류가 발생했습니다.

PUT

작업 ID: PutApplicationPolicy

애플리케이션에 대한 권한 정책을 설정합니다. 이 작업에 지원되는 작업 목록은 애플리케이션 권한을 참조하세요.

경로 파라미터

명칭	유형	필수	설명
applicationId	String	True	애플리케이션의 Amazon 리소스 이름(ARN)입니다.

응답

상태 코드	응답 모델	설명
200	ApplicationPolicy	Success

상태 코드	응답 모델	설명
400	BadRequestException	요청의 파라미터 중 하나가 잘못되었습니다.
403	ForbiddenException	클라이언트가 인증되지 않았습니다.
404	NotFoundException	요청에 지정된 리소스(예: 액세스 정책 설명)가 존재하지 않습니다.
429	TooManyRequestsException	클라이언트가 시간 단위당 허용된 요청 수를 초과하여 전송하고 있습니다.
500	InternalServerErrorException	AWS Serverless Application Repository 서비스에서 내부 오류가 발생했습니다.

OPTIONS

경로 파라미터

명칭	유형	필수	설명
applicationId	String	True	애플리케이션의 Amazon 리소스 이름(ARN)입니다.

응답

상태 코드	응답 모델	설명
200	None	응답 200개

스키마

요청 본문

PUT 스키마

```
{
  "statements": [
    {
      "statementId": "string",
      "principals": [
        "string"
      ],
      "actions": [
        "string"
      ],
      "principalOrgIDs": [
        "string"
      ]
    }
  ]
}
```

응답 본문

ApplicationPolicy 스키마

```
{
  "statements": [
    {
      "statementId": "string",
      "principals": [
        "string"
      ],
      "actions": [
        "string"
      ],
      "principalOrgIDs": [
        "string"
      ]
    }
  ]
```

```
}
```

BadRequestException 스키마

```
{
  "message": "string",
  "errorCode": "string"
}
```

ForbiddenException 스키마

```
{
  "message": "string",
  "errorCode": "string"
}
```

NotFoundException 스키마

```
{
  "message": "string",
  "errorCode": "string"
}
```

TooManyRequestsException 스키마

```
{
  "message": "string",
  "errorCode": "string"
}
```

InternalServerErrorException 스키마

```
{
  "message": "string",
  "errorCode": "string"
}
```

속성

ApplicationPolicy

애플리케이션에 적용되는 정책 설명입니다.

statements

애플리케이션에 적용된 정책 설명의 배열입니다.

유형: 유형 ApplicationPolicyStatement의 배열
필수: True

ApplicationPolicyStatement

애플리케이션에 적용되는 정책 설명입니다.

statementId

문의 고유 ID입니다.

유형: 문자열
필수: False

principals

애플리케이션을 공유할 AWS 계정 IDs 배열 또는 애플리케이션을 공개하려면 *.

유형: 유형 string의 배열
필수: True

actions

이 작업에 지원되는 작업 목록은 애플리케이션 권한을 참조하세요.

유형: 유형 string의 배열
필수: True

principalOrgIDs

애플리케이션을 공유할 AWS Organizations ID입니다.

유형: 유형 string의 배열
필수: False

BadRequestException

요청의 파라미터 중 하나가 잘못되었습니다.

message

요청의 파라미터 중 하나가 잘못되었습니다.

유형: 문자열
필수: False

errorCode

400

유형: 문자열
필수: False

ForbiddenException

클라이언트가 인증되지 않았습니다.

message

클라이언트가 인증되지 않았습니다.

유형: 문자열
필수: False

errorCode

403

유형: 문자열
필수: False

InternalServerErrorException

AWS Serverless Application Repository 서비스에서 내부 오류가 발생했습니다.

message

AWS Serverless Application Repository 서비스에서 내부 오류가 발생했습니다.

유형: 문자열
필수: False

errorCode

500

유형: 문자열
필수: False

NotFoundException

요청에 지정된 리소스(예: 액세스 정책 설명)가 존재하지 않습니다.

message

요청에 지정된 리소스(예: 액세스 정책 설명)가 존재하지 않습니다.

유형: 문자열
필수: False

errorCode

404

유형: 문자열
필수: False

TooManyRequestsException

클라이언트가 시간 단위당 허용된 요청 수를 초과하여 전송하고 있습니다.

message

클라이언트가 시간 단위당 허용된 요청 수를 초과하여 전송하고 있습니다.

유형: 문자열
필수: False

errorCode

429

유형: 문자열
필수: False

다음 사항도 참조하세요.

언어별 AWS SDKs 및 참조 중 하나에서이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하세요.

GetApplicationPolicy

- AWS 명령줄 인터페이스
- AWS SDK for .NET
- AWS SDK for C++
- Go v2용 AWS SDK
- AWS SDK for Java V2
- JavaScript V3용 AWS SDK
- Kotlin용 AWS SDK
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

PutApplicationPolicy

- AWS 명령줄 인터페이스
- AWS SDK for .NET
- AWS SDK for C++
- Go v2용 AWS SDK
- AWS SDK for Java V2
- JavaScript V3용 AWS SDK
- Kotlin용 AWS SDK
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

Applications applicationId Templates

URI

`/applications/applicationId/templates`

HTTP 메소드

POST

작업 ID: `CreateCloudFormationTemplate`

AWS CloudFormation 템플릿을 생성합니다.

경로 파라미터

명칭	유형	필수	설명
applicationId	String	True	애플리케이션의 Amazon 리소스 이름(ARN)입니다.

응답

상태 코드	응답 모델	설명
201	TemplateDetails	Success
400	BadRequestException	요청의 파라미터 중 하나가 잘못되었습니다.
403	ForbiddenException	클라이언트가 인증되지 않았습니다.
404	NotFoundException	요청에 지정된 리소스(예: 액세스 정책 설명)가 존재하지 않습니다.
429	TooManyRequestsException	클라이언트가 시간 단위당 허용된 요청 수를 초과하여 전송하고 있습니다.
500	InternalServerErrorException	AWS Serverless Application Repository 서비스에서 내부 오류가 발생했습니다.

OPTIONS

경로 파라미터

명칭	유형	필수	설명
applicationId	String	True	애플리케이션의 Amazon 리소스 이름(ARN)입니다.

응답

상태 코드	응답 모델	설명
200	None	응답 200개

스키마

요청 본문

POST 스키마

```
{
  "semanticVersion": "string"
}
```

응답 본문

TemplateDetails 스키마

```
{
  "templateId": "string",
  "templateUrl": "string",
  "applicationId": "string",
  "semanticVersion": "string",
  "status": enum,
  "creationTime": "string",
  "expirationTime": "string"
}
```

BadRequestException 스키마

```
{
  "message": "string",
  "errorCode": "string"
}
```

ForbiddenException 스키마

```
{
  "message": "string",
  "errorCode": "string"
}
```

NotFoundException 스키마

```
{
  "message": "string",
  "errorCode": "string"
}
```

TooManyRequestsException 스키마

```
{
  "message": "string",
  "errorCode": "string"
}
```

InternalServerErrorException 스키마

```
{
  "message": "string",
  "errorCode": "string"
}
```

속성

BadRequestException

요청의 파라미터 중 하나가 잘못되었습니다.

message

요청의 파라미터 중 하나가 잘못되었습니다.

유형: 문자열
필수: False

errorCode

400

유형: 문자열

필수: False

CreateCloudFormationTemplateInput

템플릿 요청을 생성합니다.

semanticVersion

애플리케이션의 의미 체계 버전:

https://semver.org/

유형: 문자열
필수: False

ForbiddenException

클라이언트가 인증되지 않았습니다.

message

클라이언트가 인증되지 않았습니다.

유형: 문자열
필수: False

errorCode

403

유형: 문자열
필수: False

InternalServerErrorException

AWS Serverless Application Repository 서비스에서 내부 오류가 발생했습니다.

message

AWS Serverless Application Repository 서비스에서 내부 오류가 발생했습니다.

유형: 문자열
필수: False

errorCode

500

유형: 문자열
필수: False

NotFoundException

요청에 지정된 리소스(예: 액세스 정책 설명)가 존재하지 않습니다.

message

요청에 지정된 리소스(예: 액세스 정책 설명)가 존재하지 않습니다.

유형: 문자열
필수: False

errorCode

404

유형: 문자열
필수: False

TemplateDetails

템플릿의 세부 정보입니다.

templateId

CreateCloudFormationTemplate에서 반환한 UUID입니다.

패턴: [0-9a-fA-F]{8}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{12}

유형: 문자열
필수: True

templateUrl

를 사용하여 애플리케이션을 배포하는 데 사용할 수 있는 템플릿에 대한 링크입니다 AWS CloudFormation.

유형: 문자열
필수: True

applicationId

애플리케이션 Amazon 리소스 이름(ARN).

유형: 문자열
필수: True

semanticVersion

애플리케이션의 의미 체계 버전:

https://semver.org/

유형: 문자열
필수: True

status

템플릿 생성 워크플로의 상태입니다.

가능한 값: `PREPARING | ACTIVE | EXPIRED`

유형: 문자열
필수: True
값: `PREPARING | ACTIVE | EXPIRED`

creationTime

이 리소스가 생성된 날짜와 시간입니다.

유형: 문자열

필수: True

expirationTime

이 템플릿이 만료되는 날짜 및 시간입니다. 템플릿은 생성 후 1시간 후에 만료됩니다.

유형: 문자열
필수: True

TooManyRequestsException

클라이언트가 시간 단위당 허용된 요청 수를 초과하여 전송하고 있습니다.

message

클라이언트가 시간 단위당 허용된 요청 수를 초과하여 전송하고 있습니다.

유형: 문자열
필수: False

errorCode

429

유형: 문자열
필수: False

다음 사항도 참조하세요.

언어별 AWS SDKs 및 참조 중 하나에서이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하세요.

CreateCloudFormationTemplate

- AWS 명령줄 인터페이스
- AWS SDK for .NET
- AWS SDK for C++
- Go v2용 AWS SDK

- AWS SDK for Java V2
- JavaScript V3용 AWS SDK
- Kotlin용 AWS SDK
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

Applications applicationId Templates templateId

URI

`/applications/applicationId/templates/templateId`

HTTP 메소드

GET

작업 ID: `GetCloudFormationTemplate`

지정된 AWS CloudFormation 템플릿을 가져옵니다.

경로 파라미터

명칭	유형	필수	설명
applicationId	String	True	애플리케이션의 Amazon 리소스 이름(ARN)입니다.
templateId	String	True	CreateCloudFormationTemplate에서 반환한 UUID입니다. 패턴: [0-9a-fA-F]{8}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{12}

응답

상태 코드	응답 모델	설명
200	TemplateDetails	Success
400	BadRequestException	요청의 파라미터 중 하나가 잘못되었습니다.
403	ForbiddenException	클라이언트가 인증되지 않았습니다.
404	NotFoundException	요청에 지정된 리소스(예: 액세스 정책 설명)가 존재하지 않습니다.
429	TooManyRequestsException	클라이언트가 시간 단위당 허용된 요청 수를 초과하여 전송하고 있습니다.
500	InternalServerErrorException	AWS Serverless Application Repository 서비스에서 내부 오류가 발생했습니다.

OPTIONS

경로 파라미터

명칭	유형	필수	설명
applicationId	String	True	애플리케이션의 Amazon 리소스 이름(ARN)입니다.
templateId	String	True	CreateCloudFormationTemplate에서 반환한 UUID입니다. 패턴: [0-9a-fA-F]{8}\-[0-9a-fA-F]{4}\-[0-9a-

명칭	유형	필수	설명
			fA-F]{4}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{12}

응답

상태 코드	응답 모델	설명
200	None	응답 200개

스키마

응답 본문

TemplateDetails 스키마

```
{
  "templateId": "string",
  "templateUrl": "string",
  "applicationId": "string",
  "semanticVersion": "string",
  "status": enum,
  "creationTime": "string",
  "expirationTime": "string"
}
```

BadRequestException 스키마

```
{
  "message": "string",
  "errorCode": "string"
}
```

ForbiddenException 스키마

```
{
```

```
  "message": "string",
  "errorCode": "string"
}
```

NotFoundException 스키마

```
{
  "message": "string",
  "errorCode": "string"
}
```

TooManyRequestsException 스키마

```
{
  "message": "string",
  "errorCode": "string"
}
```

InternalServerErrorException 스키마

```
{
  "message": "string",
  "errorCode": "string"
}
```

속성

BadRequestException

요청의 파라미터 중 하나가 잘못되었습니다.

message

요청의 파라미터 중 하나가 잘못되었습니다.

유형: 문자열
필수: False

errorCode

400

유형: 문자열
필수: False

ForbiddenException

클라이언트가 인증되지 않았습니다.

message

클라이언트가 인증되지 않았습니다.

유형: 문자열
필수: False

errorCode

403

유형: 문자열
필수: False

InternalServerErrorException

AWS Serverless Application Repository 서비스에서 내부 오류가 발생했습니다.

message

AWS Serverless Application Repository 서비스에서 내부 오류가 발생했습니다.

유형: 문자열
필수: False

errorCode

500

유형: 문자열
필수: False

NotFoundException

요청에 지정된 리소스(예: 액세스 정책 설명)가 존재하지 않습니다.

message

요청에 지정된 리소스(예: 액세스 정책 설명)가 존재하지 않습니다.

유형: 문자열
필수: False

errorCode

404

유형: 문자열
필수: False

TemplateDetails

템플릿의 세부 정보입니다.

templateId

CreateCloudFormationTemplate에서 반환한 UUID입니다.

패턴: [0-9a-fA-F]{8}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{12}

유형: 문자열
필수: True

templateUrl

를 사용하여 애플리케이션을 배포하는 데 사용할 수 있는 템플릿에 대한 링크입니다 AWS CloudFormation.

유형: 문자열

필수: True

applicationId

애플리케이션 Amazon 리소스 이름(ARN).

유형: 문자열
필수: True

semanticVersion

애플리케이션의 의미 체계 버전:

https://semver.org/

유형: 문자열
필수: True

status

템플릿 생성 워크플로의 상태입니다.

가능한 값: `PREPARING | ACTIVE | EXPIRED`

유형: 문자열
필수: True
값: `PREPARING | ACTIVE | EXPIRED`

creationTime

이 리소스가 생성된 날짜와 시간입니다.

유형: 문자열
필수: True

expirationTime

이 템플릿이 만료되는 날짜 및 시간입니다. 템플릿은 생성 후 1시간 후에 만료됩니다.

유형: 문자열
필수: True

TooManyRequestsException

클라이언트가 시간 단위당 허용된 요청 수를 초과하여 전송하고 있습니다.

message

클라이언트가 시간 단위당 허용된 요청 수를 초과하여 전송하고 있습니다.

유형: 문자열
필수: False

errorCode

429

유형: 문자열
필수: False

다음 사항도 참조하세요.

언어별 AWS SDKs 및 참조 중 하나에서이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하세요.

GetCloudFormationTemplate

- AWS 명령줄 인터페이스
- AWS SDK for .NET
- AWS SDK for C++
- Go v2용 AWS SDK
- AWS SDK for Java V2
- JavaScript V3용 AWS SDK
- Kotlin용 AWS SDK
- AWS SDK for PHP V3

- AWS SDK for Python
- AWS SDK for Ruby V3

Applications applicationId Unshare

URI

`/applications/applicationId/unshare`

HTTP 메소드

POST

작업 ID: `UnshareApplication`

AWS 조직에서 애플리케이션을 공유 해제합니다.

이 작업은 조직의 관리 계정에서만 호출할 수 있습니다.

경로 파라미터

명칭	유형	필수	설명
applicationId	String	True	애플리케이션의 Amazon 리소스 이름(ARN)입니다.

응답

상태 코드	응답 모델	설명
204	None	Success
400	BadRequestException	요청의 파라미터 중 하나가 잘못되었습니다.
403	ForbiddenException	클라이언트가 인증되지 않았습니다.

상태 코드	응답 모델	설명
404	NotFoundException	요청에 지정된 리소스(예: 액세스 정책 설명)가 존재하지 않습니다.
429	TooManyRequestsException	클라이언트가 시간 단위당 허용된 요청 수를 초과하여 전송하고 있습니다.
500	InternalServerErrorException	AWS Serverless Application Repository 서비스에서 내부 오류가 발생했습니다.

OPTIONS

경로 파라미터

명칭	유형	필수	설명
applicationId	String	True	애플리케이션의 Amazon 리소스 이름(ARN)입니다.

응답

상태 코드	응답 모델	설명
200	None	응답 200개

스키마

요청 본문

POST 스키마

```
{
```

```
  "organizationId": "string"
}
```

응답 본문

BadRequestException 스키마

```
{
  "message": "string",
  "errorCode": "string"
}
```

ForbiddenException 스키마

```
{
  "message": "string",
  "errorCode": "string"
}
```

NotFoundException 스키마

```
{
  "message": "string",
  "errorCode": "string"
}
```

TooManyRequestsException 스키마

```
{
  "message": "string",
  "errorCode": "string"
}
```

InternalServerErrorException 스키마

```
{
  "message": "string",
```

```
  "errorCode": "string"
}
```

속성

BadRequestException

요청의 파라미터 중 하나가 잘못되었습니다.

message

요청의 파라미터 중 하나가 잘못되었습니다.

유형: 문자열
필수: False

errorCode

400

유형: 문자열
필수: False

ForbiddenException

클라이언트가 인증되지 않았습니다.

message

클라이언트가 인증되지 않았습니다.

유형: 문자열
필수: False

errorCode

403

유형: 문자열
필수: False

InternalServerErrorException

AWS Serverless Application Repository 서비스에서 내부 오류가 발생했습니다.

message

AWS Serverless Application Repository 서비스에서 내부 오류가 발생했습니다.

유형: 문자열
필수: False

errorCode

500

유형: 문자열
필수: False

NotFoundException

요청에 지정된 리소스(예: 액세스 정책 설명)가 존재하지 않습니다.

message

요청에 지정된 리소스(예: 액세스 정책 설명)가 존재하지 않습니다.

유형: 문자열
필수: False

errorCode

404

유형: 문자열
필수: False

TooManyRequestsException

클라이언트가 시간 단위당 허용된 요청 수를 초과하여 전송하고 있습니다.

message

클라이언트가 시간 단위당 허용된 요청 수를 초과하여 전송하고 있습니다.

유형: 문자열
필수: False

errorCode

429

유형: 문자열
필수: False

UnshareApplicationInput

애플리케이션 요청 공유를 해제합니다.

organizationId

애플리케이션을 공유 해제할 AWS Organizations ID입니다.

유형: 문자열
필수: True

다음 사항도 참조하세요.

언어별 AWS SDKs 및 참조 중 하나에서이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하세요.

UnshareApplication

- AWS 명령줄 인터페이스
- AWS SDK for .NET
- AWS SDK for C++
- Go v2용 AWS SDK
- AWS SDK for Java V2
- JavaScript V3용 AWS SDK

- Kotlin용 AWS SDK
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

애플리케이션 applicationId 버전

URI

`/applications/applicationId/versions`

HTTP 메소드

GET

작업 ID: `ListApplicationVersions`

지정된 애플리케이션의 버전을 나열합니다.

경로 파라미터

명칭	유형	필수	설명
applicationId	String	True	애플리케이션의 Amazon 리소스 이름(ARN)입니다.

쿼리 파라미터

명칭	유형	필수	설명
maxItems	String	False	반환할 총 항목 수입니다.
nextToken	String	False	페이지 매김을 시작할 위치를 지정하기 위한 토큰입니다.

응답

상태 코드	응답 모델	설명
200	ApplicationVersionPage	Success
400	BadRequestException	요청의 파라미터 중 하나가 잘못되었습니다.
403	ForbiddenException	클라이언트가 인증되지 않았습니다.
404	NotFoundException	요청에 지정된 리소스(예: 액세스 정책 설명)가 존재하지 않습니다.
429	TooManyRequestsException	클라이언트가 시간 단위당 허용된 요청 수를 초과하여 전송하고 있습니다.
500	InternalServerErrorException	AWS Serverless Application Repository 서비스에서 내부 오류가 발생했습니다.

OPTIONS

경로 파라미터

명칭	유형	필수	설명
applicationId	String	True	애플리케이션의 Amazon 리소스 이름(ARN)입니다.

응답

상태 코드	응답 모델	설명
200	None	응답 200개

스키마

응답 본문

ApplicationVersionPage 스키마

```
{
  "versions": [
    {
      "applicationId": "string",
      "semanticVersion": "string",
      "sourceCodeUrl": "string",
      "creationTime": "string"
    }
  ],
  "nextToken": "string"
}
```

BadRequestException 스키마

```
{
  "message": "string",
  "errorCode": "string"
}
```

ForbiddenException 스키마

```
{
  "message": "string",
  "errorCode": "string"
}
```

NotFoundException 스키마

```
{
  "message": "string",
  "errorCode": "string"
}
```

TooManyRequestsException 스키마

```
{
  "message": "string",
  "errorCode": "string"
}
```

InternalServerErrorException 스키마

```
{
  "message": "string",
  "errorCode": "string"
}
```

속성

ApplicationVersionPage

애플리케이션에 대한 버전 요약 목록입니다.

versions

애플리케이션에 대한 버전 요약 배열입니다.

유형: 유형 VersionSummary의 배열
필수: True

nextToken

결과의 다음 번 페이지를 요청하기 위한 토큰.

유형: 문자열
필수: False

BadRequestException

요청의 파라미터 중 하나가 잘못되었습니다.

message

요청의 파라미터 중 하나가 잘못되었습니다.

유형: 문자열
필수: False

errorCode

400

유형: 문자열
필수: False

ForbiddenException

클라이언트가 인증되지 않았습니다.

message

클라이언트가 인증되지 않았습니다.

유형: 문자열
필수: False

errorCode

403

유형: 문자열
필수: False

InternalServerErrorException

AWS Serverless Application Repository 서비스에서 내부 오류가 발생했습니다.

message

AWS Serverless Application Repository 서비스에서 내부 오류가 발생했습니다.

유형: 문자열
필수: False

errorCode

500

유형: 문자열
필수: False

NotFoundException

요청에 지정된 리소스(예: 액세스 정책 설명)가 존재하지 않습니다.

message

요청에 지정된 리소스(예: 액세스 정책 설명)가 존재하지 않습니다.

유형: 문자열
필수: False

errorCode

404

유형: 문자열
필수: False

TooManyRequestsException

클라이언트가 시간 단위당 허용된 요청 수를 초과하여 전송하고 있습니다.

message

클라이언트가 시간 단위당 허용된 요청 수를 초과하여 전송하고 있습니다.

유형: 문자열

필수: False

errorCode

429

유형: 문자열
필수: False

VersionSummary

애플리케이션 버전 요약.

applicationId

애플리케이션 Amazon 리소스 이름(ARN).

유형: 문자열
필수: True

semanticVersion

애플리케이션의 의미 체계 버전:

https://semver.org/

유형: 문자열
필수: True

sourceCodeUrl

특정 GitHub 커밋의 URL과 같은 애플리케이션의 소스 코드에 대한 퍼블릭 리포지토리에 대한 링크입니다.

유형: 문자열
필수: False

creationTime

이 리소스가 생성된 날짜와 시간입니다.

유형: 문자열

필수: True

다음 사항도 참조하세요.

언어별 AWS SDKs 및 참조 중 하나에서이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하세요.

ListApplicationVersions

- AWS 명령줄 인터페이스
- AWS SDK for .NET
- AWS SDK for C++
- Go v2용 AWS SDK
- AWS SDK for Java V2
- JavaScript V3용 AWS SDK
- Kotlin용 AWS SDK
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

애플리케이션 applicationId 버전 semanticVersion

URI

`/applications/applicationId/versions/semanticVersion`

HTTP 메소드

PUT

작업 ID: `CreateApplicationVersion`

애플리케이션 버전을 생성합니다.

경로 파라미터

명칭	유형	필수	설명
applicationId	String	True	애플리케이션의 Amazon 리소스 이름(ARN)입니다.
semanticVersion	String	True	새 버전의 의미 체계 버전입니다.

응답

상태 코드	응답 모델	설명
201	Version	Success
400	BadRequestException	요청의 파라미터 중 하나가 잘못되었습니다.
403	ForbiddenException	클라이언트가 인증되지 않았습니다.
409	ConflictException	리소스가 이미 존재합니다.
429	TooManyRequestsException	클라이언트가 시간 단위당 허용된 요청 수를 초과하여 전송하고 있습니다.
500	InternalServerErrorException	AWS Serverless Application Repository 서비스에서 내부 오류가 발생했습니다.

OPTIONS

경로 파라미터

명칭	유형	필수	설명
applicationId	String	True	애플리케이션의 Amazon 리소스 이름(ARN)입니다.
semanticVersion	String	True	새 버전의 의미 체계 버전입니다.

응답

상태 코드	응답 모델	설명
200	None	응답 200개

스키마

요청 본문

PUT 스키마

```
{
  "templateBody": "string",
  "templateUrl": "string",
  "sourceCodeUrl": "string",
  "sourceCodeArchiveUrl": "string"
}
```

응답 본문

Version 스키마

```
{
  "applicationId": "string",
  "semanticVersion": "string",
  "sourceCodeUrl": "string",
```

```
  "sourceCodeArchiveUrl": "string",
  "templateUrl": "string",
  "creationTime": "string",
  "parameterDefinitions": [
    {
      "name": "string",
      "defaultValue": "string",
      "description": "string",
      "type": "string",
      "noEcho": boolean,
      "allowedPattern": "string",
      "constraintDescription": "string",
      "minValue": integer,
      "maxValue": integer,
      "minLength": integer,
      "maxLength": integer,
      "allowedValues": [
        "string"
      ],
      "referencedByResources": [
        "string"
      ]
    }
  ],
  "requiredCapabilities": [
    enum
  ],
  "resourcesSupported": boolean
}
```

BadRequestException 스키마

```
{
  "message": "string",
  "errorCode": "string"
}
```

ForbiddenException 스키마

```
{
  "message": "string",
  "errorCode": "string"
```

```
}
```

ConflictException 스키마

```
{
  "message": "string",
  "errorCode": "string"
}
```

TooManyRequestsException 스키마

```
{
  "message": "string",
  "errorCode": "string"
}
```

InternalServerErrorException 스키마

```
{
  "message": "string",
  "errorCode": "string"
}
```

속성

BadRequestException

요청의 파라미터 중 하나가 잘못되었습니다.

message

요청의 파라미터 중 하나가 잘못되었습니다.

유형: 문자열
필수: False

errorCode

400

유형: 문자열
필수: False

Capability

일부 애플리케이션을 배포하기 위해 지정해야 하는 값입니다.

```
CAPABILITY_IAM
CAPABILITY_NAMED_IAM
CAPABILITY_AUTO_EXPAND
CAPABILITY_RESOURCE_POLICY
```

ConflictException

리소스가 이미 존재합니다.

message

리소스가 이미 존재합니다.

유형: 문자열
필수: False

errorCode

409

유형: 문자열
필수: False

CreateApplicationVersionInput

버전 요청을 생성합니다.

templateBody

애플리케이션의 원시 패키지 AWS SAM 템플릿입니다.

유형: 문자열

필수: False

templateUrl

애플리케이션의 패키징된 AWS SAM 템플릿에 대한 링크입니다.

유형: 문자열
필수: False

sourceCodeUrl

특정 GitHub 커밋의 URL과 같은 애플리케이션의 소스 코드에 대한 퍼블릭 리포지토리에 대한 링크입니다.

유형: 문자열
필수: False

sourceCodeArchiveUrl

이 버전의 애플리케이션에 대한 소스 코드의 ZIP 아카이브가 포함된 S3 객체에 대한 링크입니다.

최대 크기 50MB

유형: 문자열
필수: False

ForbiddenException

클라이언트가 인증되지 않았습니다.

message

클라이언트가 인증되지 않았습니다.

유형: 문자열
필수: False

errorCode

403

유형: 문자열
필수: False

InternalServerErrorException

AWS Serverless Application Repository 서비스에서 내부 오류가 발생했습니다.

message

AWS Serverless Application Repository 서비스에서 내부 오류가 발생했습니다.

유형: 문자열
필수: False

errorCode

500

유형: 문자열
필수: False

ParameterDefinition

애플리케이션에서 지원하는 파라미터입니다.

name

파라미터의 이름입니다.

유형: 문자열
필수: True

defaultValue

스택 생성 시 지정된 값이 없는 경우에 사용할 템플릿에 적합한 유형의 값입니다. 파라미터에 대한 제약을 정의하는 경우 이러한 제약을 준수하는 값을 지정해야 합니다.

유형: 문자열
필수: False

description

파라미터를 설명하는 최대 4,000자의 문자열입니다.

유형: 문자열
필수: False

type

파라미터의 유형입니다.

유효값: `String | Number | List<Number> | CommaDelimitedList`

`String`: 리터럴 문자열입니다.

예를 들어 사용자는를 지정할 수 있습니다`"MyUserName"`.

`Number`: 정수 또는 float. AWS CloudFormation valid는 파라미터 값을 숫자로 확인합니다. 그러나 템플릿의 다른 곳에서 파라미터를 사용하는 경우(예: `Ref` 내장 함수 사용) 파라미터 값은 문자열이 됩니다.

예를 들어 사용자는를 지정할 수 있습니다`"8888"`.

`List<Number>`: 쉼표로 구분된 정수 또는 부동 소수점의 배열입니다. AWS CloudFormation 는 파라미터 값을 숫자로 검증합니다. 그러나 템플릿의 다른 곳에서 파라미터를 사용하는 경우(예: `Ref` 내장 함수 사용) 파라미터 값은 문자열 목록이 됩니다.

예를 들어 사용자는 "80,20"을 지정한 다음를 `Ref` 생성할 수 있습니다`["80","20"]`.

`CommaDelimitedList`: 쉼표로 구분된 리터럴 문자열의 배열입니다. 총 문자열 수는 총 쉼표 수보다 하나 더 많아야 합니다. 또한 각 멤버 문자열은 공백으로 잘립니다.

예를 들어 사용자는 "test,dev,prod"를 지정한 다음 결과를 로 `Ref` 지정할 수 있습니다`["test","dev","prod"]`.

유형: 문자열
필수: False

noEcho

스택을 설명하는 호출을 할 때마다 파라미터 값을 마스킹할지 여부입니다. 값을 true로 설정하면 파라미터 값이 별표(*****)로 마스킹됩니다.

유형: 부울
필수: False

allowedPattern

`String` 유형에 허용할 패턴을 나타내는 정규식입니다.

유형: 문자열
필수: False

constraintDescription

제약 위반 시 해당 제약을 설명하는 문자열입니다. 예를 들어 제약 설명이 없으면 `[A-Za-z0-9]+` 패턴이 허용된 파라미터에 사용자가 유효하지 않은 값을 지정할 때 다음과 같은 오류 메시지가 표시됩니다.

```
Malformed input-Parameter MyParameter must match pattern [A-Za-z0-9]+
```

"대문자와 소문자 및 숫자만 포함해야 합니다"와 같은 제약 조건 설명을 추가하면 다음과 같은 사용자 지정 오류 메시지를 표시할 수 있습니다.

```
Malformed input-Parameter MyParameter must contain only uppercase and
lowercase letters and numbers.
```

유형: 문자열
필수: False

minValue

`Number` 유형에 허용할 가장 작은 숫자 값을 결정하는 숫자 값입니다.

유형: 정수
필수: 거짓

maxValue

`Number` 유형에 허용할 가장 큰 숫자 값을 결정하는 숫자 값입니다.

유형: 정수

필수: 거짓

minLength

`String` 유형에 허용할 최소 문자 수를 결정하는 정수 값입니다.

유형: 정수
필수: 거짓

maxLength

`String` 유형에 허용할 최대 문자 수를 결정하는 정수 값입니다.

유형: 정수
필수: 거짓

allowedValues

파라미터에 허용되는 값 목록을 포함하는 어레이입니다.

유형: 유형 string의 배열
필수: False

referencedByResources

이 파라미터를 사용하는 AWS SAM 리소스 목록입니다.

유형: 유형 string의 배열
필수: True

TooManyRequestsException

클라이언트가 시간 단위당 허용된 요청 수를 초과하여 전송하고 있습니다.

message

클라이언트가 시간 단위당 허용된 요청 수를 초과하여 전송하고 있습니다.

유형: 문자열

필수: False

errorCode

429

유형: 문자열
필수: False

Version

애플리케이션 버전 세부 정보입니다.

applicationId

애플리케이션 Amazon 리소스 이름(ARN).

유형: 문자열
필수: True

semanticVersion

애플리케이션의 의미 체계 버전:

https://semver.org/

유형: 문자열
필수: True

sourceCodeUrl

특정 GitHub 커밋의 URL과 같은 애플리케이션의 소스 코드에 대한 퍼블릭 리포지토리에 대한 링크입니다.

유형: 문자열
필수: False

sourceCodeArchiveUrl

이 버전의 애플리케이션에 대한 소스 코드의 ZIP 아카이브가 포함된 S3 객체에 대한 링크입니다.

최대 크기 50MB

유형: 문자열
필수: False

templateUrl

애플리케이션의 패키징된 AWS SAM 템플릿에 대한 링크입니다.

유형: 문자열
필수: True

creationTime

이 리소스가 생성된 날짜 및 시간입니다.

유형: 문자열
필수: True

parameterDefinitions

애플리케이션에서 지원하는 파라미터 유형의 배열입니다.

유형: 유형 ParameterDefinition의 배열
필수: True

requiredCapabilities

특정 애플리케이션을 배포하기 전에 지정해야 하는 값 목록입니다. 일부 애플리케이션에는 새 AWS Identity and Access Management (IAM) 사용자를 생성하는 등 AWS 계정의 권한에 영향을 미칠 수 있는 리소스가 포함될 수 있습니다. 이러한 애플리케이션의 경우이 파라미터를 지정하여 기능을 명시적으로 승인해야 합니다.

유일하게 유효한 값은 `CAPABILITY_IAM`, `CAPABILITY_RESOURCE_POLICY`, `CAPABILITY_NAMED_IAM`및 입니다`CAPABILITY_AUTO_EXPAND`.

다음 리소스에는 `CAPABILITY_IAM` 또는 `CAPABILITY_NAMED_IAM`을 지정해야 합니다. AWS::IAM::Group, AWS::IAM::InstanceProfile, AWS::IAM::Policy, and AWS::IAM::Role. 애플리케이션에 IAM 리소스가 포함된 경우 `CAPABILITY_IAM` 또는 중 하나를 지정할 수 있습니

다CAPABILITY_NAMED_IAM. 애플리케이션에 사용자 지정 이름을 가진 IAM 리소스가 포함되어 있는 경우 CAPABILITY_NAMED_IAM을 지정해야 합니다.

AWSCAPABILITY_RESOURCE_POLICYAWS::Lambda::Permission, AWS::IAM:Policy, AWS::ApplicationAutoScaling::ScalingPolicy, AWS::S3::BucketPolicy, AWS::SQS::QueuePolicy 및 AWS::SNS::TopicPolicy 리소스를 지정해야 합니다.

중첩 애플리케이션을 한 개 이상 포함하는 애플리케이션은 CAPABILITY_AUTO_EXPAND를 지정해야 합니다.

애플리케이션 템플릿에 위의 리소스가 포함된 경우 배포하기 전에 애플리케이션과 연결된 모든 권한을 검토하는 것이 좋습니다. 기능이 필요한 애플리케이션에 대해이 파라미터를 지정하지 않으면 호출이 실패합니다.

유형: 유형 Capability의 배열
필수: True

resourcesSupported

이 애플리케이션에 포함된 모든 AWS 리소스가 검색되는 리전에서 지원되는지 여부입니다.

유형: 부울
필수: True

다음 사항도 참조하세요.

언어별 AWS SDKs 및 참조 중 하나에서이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하세요.

CreateApplicationVersion

- AWS 명령줄 인터페이스
- AWS SDK for .NET
- AWS SDK for C++
- Go v2용 AWS SDK
- AWS SDK for Java V2
- JavaScript V3용 AWS SDK

- Kotlin용 AWS SDK
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

문서 기록

- API 버전: 최신
- 설명서 최종 업데이트: 2020년 3월 10일

다음 표는 AWS Serverless Application Repository 개발자 안내서의 각 릴리스에서 변경된 중요 사항에 대해 설명합니다. 이 설명서에 대한 업데이트 알림을 받으려면 RSS 피드를 구독하면 됩니다.

변경 사항	설명	날짜
애플리케이션에 대한 액세스 공유 및 제한 업데이트	AWS 조직의 계정에 애플리케이션을 공유하고 계정 및 조직의 퍼블릭 애플리케이션에 대한 AWS 액세스를 제한하는 지원이 추가되었습니다 AWS . 조직의 사용자에게 애플리케이션을 공유하는 자세한 예는 AWS Serverless Application Repository 애플리케이션 정책 예제를 참조하세요. 공개 애플리케이션에 대한 액세스 제한 방법의 예는 AWS Serverless Application Repository ID 기반 정책 예를 참조하십시오.	2020년 3월 10일
새로 지원되는 리소스	여러 가지 리소스에 대한 지원이 추가되었습니다. 지원되는 리소스의 전체 목록은 지원되는 AWS 리소스 목록을 참조하세요.	2020년 1월 17일
중국 리전	이제 중국 리전, 베이징 및 닝샤에서 AWS Serverless Application Repository 를 사용할 수 있습니다. AWS Serverless Application Repository 리전 및	2020년 1월 15일

	엔드포인트에 대한 자세한 내용은의 리전 및 엔드포인트를 참조하세요AWS 일반 참조.	
다른 AWS 서비스와의 일관성을 위해 보안 섹션을 업데이트했습니다.	자세한 내용은 보안을 참조하세요.	2020년 1월 2일
애플리케이션 게시를 위한 간소화된 프로세스	AWS SAM CLI의 새 `sam publish` 명령은에서 서버리스 애플리케이션을 게시하는 프로세스를 간소화합니다 AWS Serverless Application Repository. 샘플 애플리케이션 다운로드 및 게시에 대한 종합적인 자습서는 빠른 시작: 애플리케이션 게시를 참조하십시오. AWS 클라우드에서 이미 개발하고 테스트한 애플리케이션을 게시하는 방법에 대한 지침은 AWS SAM CLI를 통해 애플리케이션 게시를 참조하세요.	2018년 12월 21일
중첩 애플리케이션 및 계층 지원	중첩 애플리케이션과 계층 지원 추가 여기에는 지원되는 AWS 리소스에 대한 업데이트 및 애플리케이션 기능 확인이 포함됩니다.	2018년 11월 29일

사용자 지정 IAM 역할 및 리소스 정책을 사용하여 애플리케이션 게시	사용자 지정 IAM 역할 및 리소스 정책을 사용하여 애플리케이션 게시를 위한 지원 추가 여기에는 사용 애플리케이션 및 게시 애플리케이션 워크플로에 대한 업데이트와 AWS Serverless Application Repository 개발자 안내서의 지원되는 AWS 리소스 및 API 참조에 대한 업데이트가 포함됩니다.	2018년 11월 16일
정책 템플릿 업데이트	AWS Serverless Application Repository 개발자 안내서의 지원되는 정책 템플릿 업데이트.	2018년 9월 26일
설명서 업데이트	AWS Serverless Application Repository 개발자 안내서에 인증 및 액세스 제어 주제가 추가되었습니다.	2018년 7월 2일
공개 릴리스	이제 14개 AWS 리전에서 사용할 수 AWS Serverless Application Repository있는의 공개 릴리스입니다. 를 AWS Serverless Application Repository 사용할 수 있는 AWS 리전 및 AWS Serverless Application Repository 엔드포인트에 대한 자세한 내용은의 리전 및 엔드포인트를 참조하세요AWS 일반 참조.	2018년 2월 20일
새 안내서	개발자 AWS Serverless Application Repository 안내서의 첫 번째 미리 보기 릴리스입니다.	2017년 11월 30일

AWS 용어집

최신 AWS 용어는 AWS 용어집 참조의 AWS 용어집을 참조하세요.

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