aws

참조 안내서

AWS SDKs 및 도구

AWS SDKs 및 도구: 참조 안내서

Table of Contents

AWS SDKs 및 도구 참조 가이드에서 다루는 내용

많은 SDK와 도구는 설계 사양 공유나 공유 라이브러리를 통해 몇 가지 일반 기능을 공유합니다.

이 안내서에는 다음과 관련된 정보가 포함되어 있습니다.

- 글로벌 구성 AWS SDKs 및 도구 - 공유 `config` 및 `credentials` 파일 또는 환경 변수를 사용하여 AWS SDKs 및 도구를 구성하는 방법.
- AWS SDKs 및 도구를 사용한 인증 및 액세스 -를 개발할 AWS 때 코드 또는 도구가를 인증하는 방법을 설정합니다 AWS 서비스.
- AWS SDKs 및 도구 설정 참조 - 인증 및 구성에 사용할 수 있는 모든 표준 설정에 대한 참조
- AWS 공통 런타임(CRT) 라이브러리 - 거의 모든 SDK에서 사용할 수 있는 공유 AWS 공통 런타임(CRT) 라이브러리의 개요입니다. SDKs
- AWS SDKs 및 도구 유지 관리 정책 에서는 모바일 및 사물 인터넷(IoT) SDKs를 비롯한 AWS 소프트웨어 개발 키트(SDKs) 및 도구에 대한 유지 관리 정책 및 버전 관리와 기본 종속성을 다룹니다.

이 AWS SDKs 및 도구 참조 가이드는 여러 SDKs 및 도구에 적용할 수 있는 정보의 기반이 되기 위한 것입니다. 여기에 제공된 모든 정보 외에도 사용자가 사용하고 있는 SDK 또는 도구에 대한 특정 가이드도 사용해야 합니다. 다음은 이 가이드의 관련 자료 섹션이 포함된 SDK 및 도구입니다.

사용 중인 제품:	이 가이드의 관련 섹션은 다음과 같습니다.
• 모든 SDK 또는 도구	AWS SDKs 및 도구 유지 관리 정책
• AWS Cloud9 • AWS CDK • AWS Toolkit for Azure DevOps • AWS Toolkit for JetBrains • AWS Toolkit for Visual Studio • AWS Toolkit for Visual Studio Code • AWS Serverless Application Model • AWS CodeArtifact • AWS CodeBuild	글로벌 구성 AWS SDKs 및 도구 AWS SDKs 및 도구를 사용한 인증 및 액세스 AWS SDKs 및 도구 유지 관리 정책

사용 중인 제품:	이 가이드의 관련 섹션은 다음과 같습니다.
• Amazon CodeCatalyst • AWS CodeCommit • AWS CodeDeploy • AWS CodePipeline	
• AWS CLI • AWS SDK for C++ • AWS SDK for Go • AWS SDK for Java • AWS SDK for JavaScript • AWS SDK for Kotlin • AWS SDK for .NET • AWS SDK for PHP • AWS SDK for Python (Boto3) • AWS SDK for Ruby • AWS SDK for Rust • AWS SDK for Swift • AWS Tools for Windows PowerShell	글로벌 구성 AWS SDKs 및 도구 AWS SDKs 및 도구를 사용한 인증 및 액세스 AWS SDKs 및 도구 설정 참조 AWS 공통 런타임(CRT) 라이브러리 AWS SDKs 및 도구 유지 관리 정책 AWS SDKs 및 도구 버전 수명 주기

- 애플리케이션을 개발하는 데 도움이 될 수 있는 도구에 대한 개요는 빌드 기반 도구를 AWS AWS참조하세요.
- 지원에 대한 자세한 내용은 AWS 지식 센터를 참조하십시오.
- AWS 용어에 대해서는 AWS 용어집 참조의 AWS 용어집을 참조하세요.

개발자 리소스

Amazon Q Developer는 AWS 애플리케이션을 이해, 구축, 확장 및 운영하는 데 도움이 되는 생성형 AI 기반 대화형 어시스턴트입니다. 빌드 속도를 높이기 위해 Amazon Q AWS를 지원하는 모델을 고품질 AWS 콘텐츠로 보강하여 더 완전하고 실행 가능하며 참조된 답변을 생성합니다. 자세한 내용은 Amazon Q Developer 사용 설명서의 Amazon Q Developer란 무엇인가요?를 참조하세요.

도구 키트 원격 측정 알림

AWS 통합 개발 환경(IDE) 도구 키트는 IDE의 AWS 서비스에 액세스할 수 있는 플러그인 및 확장 프로그램입니다. Amazon Q IDE 플러그인 및 확장을 사용하면 IDE에서 생성형 AI 지원을 사용할 수 있습니다. 각 IDE 도구 키트에 대한 자세한 내용은 이전 표의 도구 키트 사용 설명서를 참조하세요. IDE에서 Amazon Q를 사용하는 방법에 대한 자세한 내용은 Amazon Q 개발자 안내서의 IDE에서 Amazon Q 사용을 참조하세요.

AWS IDE Toolkits 및 Amazon Q는 클라이언트 측 원격 측정 데이터를 수집하고 저장하여 향후 AWS Toolkit 및 Amazon Q 릴리스에 대한 결정을 알릴 수 있습니다. 수집된 데이터는 AWS 도구 키트 및 Amazon Q의 사용을 정량화합니다.

모든 AWS IDE 도구 키트 및 Amazon Q에서 수집된 원격 측정 데이터에 대한 자세한 내용은 `aws-toolkit-common` Github 리포지토리의 commonDefinitions.json 문서를 참조하세요.

각 AWS IDE 도구 키트 및 Amazon Q 확장에서 수집한 원격 측정 데이터에 대한 자세한 내용은 다음 AWS 도구 키트 GitHub 리포지토리의 리소스 문서를 참조하세요.

- AWS Amazon Q를 사용한 Visual Studio Toolkit
- AWS Toolkit for Visual Studio Code 및 VS Code용 Amazon Q 확장
- AWS Toolkit for JetBrains 및 JetBrains용 Amazon Q 플러그인
- Amazon Q for Eclipse

AWS 도구 키트에서 액세스할 수 있는 특정 AWS 서비스는 추가 클라이언트 측 원격 측정 데이터를 수집할 수 있습니다. 각 개별 AWS 서비스에서 수집하는 데이터 유형에 대한 자세한 내용은 관심 있는 특정 서비스의 AWS 설명서 주제를 참조하세요.

글로벌 구성 AWS SDKs 및 도구

AWS SDKs 및 AWS Command Line Interface (AWS CLI)와 같은 기타 AWS 개발자 도구를 사용하면 AWS 서비스 APIs. 하지만 이를 시도하기 전에 요청 작업을 수행하는 데 필요한 정를 사용하여 SDK 또는 도구를 구성해야 합니다.

이 정보에는 다음과 같은 항목이 포함됩니다.

- API를 호출자를 식별하는 보안 인증 정보. 자격 증명은 AWS 서버에 대한 요청을 암호화하는 데 사용됩니다. 이 정보를 사용하여는 자격 증명을 AWS 확인하고 자격 증명과 연결된 권한 정책을 검색할 수 있습니다. 그러면 사용자에게 허용된 작업을 결정할 수 있습니다.
- AWS CLI 또는 SDK에 요청을 처리하는 방법, 요청을 보낼 위치(서비스 AWS 엔드포인트), 응답을 해석하거나 표시하는 방법을 알리는 데 사용하는 기타 구성 세부 정보입니다.

각 SDK 또는 도구는 필요한 보안 인증 및 구성 정보를 제공하는 데 사용할 수 있는 여러 소스를 지원합니다. 일부 소스는 SDK 또는 도구에만 적용되며, 해당 방법을 사용하는 방법에 대한 자세한 사항은 해당 도구 또는 SDK 설명서를 참조해야 합니다.

그러나 AWS SDKs 및 도구는 코드 자체 이외의 기본 소스의 공통 설정을 지원합니다. 이 섹션은 다음 주제를 포함합니다.

주제

공유 `config` 및 `credentials` 파일을 사용하여 AWS SDKs 및 도구 전역 구성

공유 AWS `config` 및 `credentials` 파일은 AWS SDK 또는 도구에 대한 인증 및 구성을 지정할 수 있는 가장 일반적인 방법입니다.

공유 `config` 및 `credentials` 파일에는 프로필 세트가 포함되어 있습니다. 프로필은 키-값 페어로 구성된 구성 설정 세트로, AWS SDKs, AWS Command Line Interface (AWS CLI) 및 기타 도구에서 사용됩니다. 구성 값은 프로파일이 사용될 때 SDK/도구의 일부 측면을 구성하기 위해 프로파일에 첨부됩

니다. 이러한 파일은 값이 사용자의 로컬 환경에 있는 모든 애플리케이션, 프로세스 또는 SDK에 적용된다는 점에서 “공유”됩니다.

공유 `config` 및 `credentials` 파일 모두 ASCII 문자만 포함된 일반 텍스트 파일입니다(UTF-8 인코딩). 이들은 일반적으로 INI 파일이라고 하는 형식을 취합니다.

프로파일

공유 `config` 및 `credentials` 파일 내의 설정은 특정 프로파일과 연결됩니다. 파일 내에 여러 프로필을 정의하여 다양한 개발 환경에 적용할 다양한 설정 구성을 생성할 수 있습니다.

`[default]` 프로파일에는 이름이 지정된 특정 프로파일이 지정되지 않은 경우 SDK 또는 도구 작업에 사용되는 값이 포함됩니다. 이름으로 명시적으로 참조할 수 있는 별도의 프로파일을 만들 수도 있습니다. 각 프로필은 애플리케이션 및 시나리오에서 필요에 따라 다양한 설정과 값을 사용할 수 있습니다.

> **Note**
>
> `[default]`은(는)단순히 이름이 지정되지 않은 프로파일입니다. 이 프로파일은 사용자가 프로파일을 지정하지 않을 경우 SDK에서 사용하는 기본 프로파일이기 때문에 `default`(이)라는 이름이 지정되었습니다. 상속된 기본값을 다른 프로파일에 제공하지 않습니다. `[default]` 프로파일에 무언가를 설정하고 명명된 프로파일에 설정하지 않으면 명명된 프로파일을 사용할 때 값이 설정되지 않습니다.

명명된 프로필 설정

`[default]` 프로파일과 이름이 지정된 여러 프로파일이 동일한 파일에 존재할 수 있습니다. 다음 설정을 사용하여 코드를 실행할 때 SDK 또는 도구에서 사용할 프로필의 설정을 선택합니다. 로 작업할 때 코드 내에서 또는 명령별로 프로파일을 선택할 수도 있습니다 AWS CLI.

다음 중 하나를 설정하여이 기능을 구성합니다.

`AWS_PROFILE` - 환경 변수

이 환경 변수가 명명된 프로파일 또는 "기본값"으로 설정된 경우 모든 SDK 코드 및 AWS CLI 명령은 해당 프로파일의 설정을 사용합니다.

명령 행을 통한 환경 변수 설정의 Linux/macOS 예제:

```
export AWS_PROFILE="my_default_profile_name";
```

명령줄을 통해 환경 변수를 설정하는 Windows 예제:

```
setx AWS_PROFILE "my_default_profile_name"
```

`aws.profile` - JVM 시스템 속성

JVM의 SDK for Kotlin과 Java 2.x용 SDK의 경우 `aws.profile` 시스템 속성을 설정할 수 있습니다. SDK는 서비스 클라이언트를 생성할 때 코드에서 설정이 재정의되지 않는 한 명명된 프로파일의 설정을 사용합니다. Java 1.x용 SDK는이 시스템 속성을 지원하지 않습니다.

> **Note**
>
> 애플리케이션이 여러 애플리케이션을 실행하는 서버에 있는 경우 기본 프로필 대신 항상 명명된 프로필을 사용하는 것이 좋습니다. 기본 프로필은 환경의 모든 AWS 애플리케이션에서 자동으로 선택되며 이들 간에 공유됩니다. 따라서 다른 사용자가 애플리케이션의 기본 프로필을 업데이트하면 다른 사용자에게 의도하지 않게 영향을 미칠 수 있습니다. 이를 방지하려면 공유 `config` 파일에서 명명된 프로파일을 정의한 다음 코드에서 명명된 프로파일을 설정하여 애플리케이션에서 해당 명명된 프로파일을 사용합니다. 범위가 애플리케이션에만 영향을 미친다는 것을 알고 있는 경우 환경 변수 또는 JVM 시스템 속성을 사용하여 명명된 프로파일을 설정할 수 있습니다.

구성 파일 형식

`config` 파일은 섹션으로 구성됩니다. 섹션은 이름이 지정된 설정 모음이며 다른 섹션 정의 라인을 찾을 때까지 계속됩니다.

`config` 파일은 다음 형식을 사용하는 일반 텍스트 파일입니다.

- 섹션의 모든 항목은 `setting-name=value`와 같은 일반적인 형식을 취합니다.
- 줄은 해시태그 문자(#)로 시작하여 주석 처리할 수 있습니다.

섹션 유형

섹션 정의는 설정 모음에 이름을 적용하는 줄입니다. 섹션 정의 줄은 대괄호(`[]`)로 시작하고 끝납니다. 대괄호 안에는 섹션 유형 식별자와 섹션의 사용자 지정 이름이 있습니다. 문자, 숫자, 하이픈(`-`)및 밑줄(`_`)은 사용할 수 있지만 공백은 사용할 수 없습니다.

섹션 유형: **default**

섹션 정의 줄의 예: [default]

[default]는 profile 섹션 식별자가 필요하지 않은 유일한 프로필입니다.

다음은 [default] 프로파일이 있는 기본 config 파일을 보여주는 예입니다. region 설정값을 설정합니다. 다른 섹션 정의가 발생할 때까지이 줄을 따르는 모든 설정은이 프로파일의 일부입니다.

```
[default]
#Full line comment, this text is ignored.
region = us-east-2
```

섹션 유형: **profile**

섹션 정의 줄의 예: [profile *dev*]

profile 섹션 정의 줄은 다양한 개발 시나리오에 적용할 수 있는 명명된 구성 그룹입니다. 명명된 프로파일의 이해를 높이려면 프로파일의 이전 섹션을 참조하십시오.

다음 예제에서는 profile 섹션 정의 줄과 이름이 지정된 프로파일이 있는 config 파일을 보여줍니다foo. 다른 섹션 정의가 발생할 때까지이 줄을 따르는 모든 설정은이 명명된 프로파일의 일부입니다.

```
[profile foo]
...settings...
```

일부 설정에는 다음 예제의 s3 설정 및 하위 설정과 같은 중첩된 자체 하위 설정 그룹이 있습니다. 하나 이상의 공백으로 들여쓰기하여 하위 설정을 그룹과 연결합니다.

```
[profile test]
region = us-west-2
s3 =
    max_concurrent_requests=10
    max_queue_size=1000
```

섹션 유형: **sso-session**

섹션 정의 줄의 예: [sso-session *my-sso*]

sso-session 섹션 정의 줄은를 사용하여 AWS 자격 증명을 확인하도록 프로필을 구성하는 데 사용하는 설정 그룹의 이름을 지정합니다 AWS IAM Identity Center. Single Sign-On 인증 구성에 대한 자세

한 내용은 IAM Identity Center를 사용하여 AWS SDK 및 도구 인증 섹션을 참조하십시오. 프로파일은 키-값 쌍으로 sso-session 섹션에 연결됩니다. 여기서 sso-session 값은 키이고 sso-session 섹션 이름은 sso-session = <name-of-sso-session-section> 같은 값입니다.

1다음 예에서는 'my-sso'의 토큰을 사용하여 '111122223333' 계정의 'SampleRole' IAM 역할에 대한 단기 AWS 보안 인증을 가져올 프로파일을 구성합니다. "my-sso" sso-session 섹션은 sso-session 키를 사용하여 profile 섹션에서 이름으로 참조됩니다.

```
[profile dev]
sso_session = my-sso
sso_account_id = 111122223333
sso_role_name = SampleRole

[sso-session my-sso]
sso_region = us-east-1
sso_start_url = https://my-sso-portal.awsapps.com/start
```

섹션 유형: **services**

섹션 정의 줄의 예: [services dev]

> **Note**
>
> 이 services 섹션은 서비스별 엔드포인트 사용자 정의를 지원하며 이 기능이 포함된 SDK 및 도구에서만 사용할 수 있습니다. SDK에서 이 기능을 사용할 수 있는지 확인하려면 서비스별 엔드포인트에 대한 AWS SDKs 도구 지원을(를)참조하십시오.

services 섹션 정의 줄은 AWS 서비스 요청에 대한 사용자 지정 엔드포인트를 구성하는 설정 그룹의 이름을 지정합니다. 프로파일은 키-값 쌍으로 services 섹션에 연결됩니다. 여기서 services 값은 키이고 services 섹션 이름은 services = <name-of-services-section> 같은 값입니다.

services 섹션은 <SERVICE> = 줄별로 하위 섹션으로 더 구분되며, 여기서는 AWS 서비스 식별자 키<SERVICE>입니다. AWS 서비스 식별자는 모든 공백을 밑줄로 바꾸고 모든 문자를 소문자로 대체serviceId하여 API 모델의를 기반으로 합니다. services 섹션에서 사용할 모든 서비스 식별자 키 목록은 서비스별 엔드포인트 식별자을 참조하세요. 서비스 식별자 키 뒤에는 각각 고유한 줄에 공백 두 개로 들여쓰기하여 중첩된 설정이 이어집니다.

다음 예에서는 services 정의를 사용하여 Amazon DynamoDB 서비스에 대한 요청에만 사용할 엔드포인트를 구성합니다. "local-dynamodb" services 섹션은 services 키를 사용하여 profile

섹션에서 이름으로 참조됩니다. AWS 서비스 식별자 키는 입니다dynamodb. Amazon DynamoDB 서비스 하위 섹션은 줄에서 시작됩니다dynamodb = . 들여쓰기된 바로 다음 줄은 해당 하위 섹션에 포함되며 해당 서비스에 적용됩니다.

```
[profile dev]
services = local-dynamodb

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

사용자 지정 엔드포인트 구성에 대한 자세한 내용은 서비스별 엔드포인트 섹션을 참조하십시오.

보안 인증 파일의 형식

프로파일 섹션이 profile 단어로 시작하지 않는다는 점을 제외하면 credentials 파일의 규칙은 일반적으로 config 파일 규칙과 동일합니다. 대괄호 사이에는 프로필 이름 자체만 사용합니다. 다음 예제는 라는 명명된 프로파일 섹션이 있는 credentials 파일을 보여줍니다foo.

```
[foo]
...credential settings...
```

'비밀' 또는 민감한 설정으로 간주되는 aws_access_key_id, aws_secret_access_key및 만 credentials 파일에 저장할 수 있습니다aws_session_token. 이러한 설정을 공유 config 파일에 배치할 수도 있지만 이러한 민감한 값을 별도의 credentials 파일에 보관하는 것이 좋습니다. 이렇게 하면 필요한 경우 각 파일에 별도의 권한을 제공할 수 있습니다.

다음은 [default] 프로파일이 있는 기본 credentials 파일을 보여주는 예입니다. aws_access_key_id, aws_secret_access_key및 aws_session_token 전역 설정을 설정합니다.

```
[default]
aws_access_key_id=AKIAIOSFODNN7EXAMPLE
aws_secret_access_key=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
aws_session_token=IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZVERYLONGSTRI
```

credentials 파일에서 명명된 프로필을 사용하든 "default"를 사용하든 상관없이 여기에 있는 모든 설정은 동일한 프로필 이름을 사용하는 config 파일의 모든 설정과 결합됩니다. 동일한 이름을 공유하는 프로파일에 대한 보안 인증이 두 파일 모두에 있는 경우 보안 인증 파일의 키가 우선합니다.

AWS SDKs 및 도구의 공유 **config** 및 **credentials** 파일의 위치 찾기 및 변경

공유 AWS `config` 및 `credentials` 파일은 AWS SDKs. 파일은 환경에 로컬로 상주하며 SDK 코드 또는 해당 환경에서 실행하는 AWS CLI 명령에 의해 자동으로 사용됩니다. 예를 들어 자체 컴퓨터에서 또는 Amazon Elastic Compute Cloud 인스턴스에서 개발할 때.

SDK 또는 도구가 실행되면 이러한 파일을 확인하고 사용 가능한 구성 설정을 로드합니다. 파일이 아직 없는 경우 SDK 또는 도구에 의해 기본 파일이 자동으로 생성됩니다.

기본적으로 파일은 home 또는 사용자 폴더에 있는 라는 폴더에 `.aws` 있습니다.

운영 체제	기본 위치 및 파일 이름

Linux 및 macOS	`~/.aws/config`
	
	`~/.aws/credentials`

Windows	`%USERPROFILE%\.aws\config`
	
	`%USERPROFILE%\.aws\credentials`

홈 디렉터리 해상도

~는 다음과 같은 경우에만 홈 디렉터리 확인에 사용됩니다.

- 경로를 시작합니다.
- 바로 뒤에 / 또는 플랫폼별 구분자가 옵니다. 창에서 ~/ 및는 ~\ 홈 디렉터리로 확인됩니다.

홈 디렉터리를 결정할 때 다음 변수를 확인합니다.

- (모든 플랫폼)`HOME` 환경 변수
- (Windows 플랫폼)`USERPROFILE` 환경 변수
- (Windows 플랫폼) `HOMEDRIVE` 및 `HOMEPATH` 환경 변수의 연결(`$HOMEDRIVE$HOMEPATH`)
- (SDK 또는 도구별 선택 사항)SDK 또는 도구별 홈 경로 확인 기능 또는 변수

가능한 경우, 경로의 시작 부분에 사용자의 홈 디렉터리(예:~username/)가 지정되어 있으면 그것은 요청된 사용자 이름의 홈 디렉터리(예: /home/username/.aws/config)로 확정됩니다.

이러한 파일의 기본 위치 변경

다음 중 하나를 사용하여 SDK 또는 도구에서 이러한 파일을 로드하는 위치를 재정의할 수 있습니다.

환경 변수 사용

다음 환경 변수는 이러한 파일의 위치 또는 이름을 기본값에서 사용자 지정 값으로 변경할 수 있습니다.

- config 파일 환경 변수:**AWS_CONFIG_FILE**
- credentials 파일 환경 변수:**AWS_SHARED_CREDENTIALS_FILE**

Linux/macOS

Linux 또는 macOS에서 다음의 내보내기 명령을 실행하여 대체 위치를 지정할 수 있습니다.

```
$ export AWS_CONFIG_FILE=/some/file/path/on/the/system/config-file-name
$ export AWS_SHARED_CREDENTIALS_FILE=/some/other/file/path/on/the/system/
credentials-file-name
```

Windows

Windows에서 다음의 setx 명령을 실행하여 대체 위치를 지정할 수 있습니다.

```
C:\> setx AWS_CONFIG_FILE c:\some\file\path\on\the\system\config-file-name
C:\> setx AWS_SHARED_CREDENTIALS_FILE c:\some\other\file\path\on\the\system
\credentials-file-name
```

환경 변수를 사용하여 시스템을 구성하는 방법에 대한 자세한 내용은 섹션을 참조하세요환경 변수를 사용하여 AWS SDKs 및 도구 전역 구성.

JVM 시스템 속성 사용

JVM에서 실행되는 SDK for Kotlin 및 SDK for Java 2.x의 경우 다음 JVM 시스템 속성을 설정하여 이러한 파일의 위치 또는 이름을 기본값에서 사용자 지정 값으로 변경할 수 있습니다.

- config 파일 JVM 시스템 속성: **aws.configFile**

- `credentials` 파일 환경 변수:**`aws.sharedCredentialsFile`**

JVM 시스템 속성을 설정하는 방법에 대한 지침은 섹션을 참조하세요the section called “JVM 시스템 속성을 설정하는 방법”. Java 1.x용 SDK는 이러한 시스템 속성을 지원하지 않습니다.

환경 변수를 사용하여 AWS SDKs 및 도구 전역 구성

환경 변수는 AWS SDKs 및 도구를 사용할 때 구성 옵션과 자격 증명을 지정하는 또 다른 방법을 제공합니다. 환경 변수는 이름이 지정된 프로파일을 스크립팅하거나 일시적으로 기본값으로 설정하는 데 유용할 수 있습니다. 대부분의 SDK에서 지원하는 환경 변수 목록은 환경 변수 목록 섹션을 참조하십시오.

옵션의 우선 순위

- 환경 변수를 사용하여 설정을 지정하면 공유 AWS `config` 및 `credentials` 파일의 프로파일에서 로드된 모든 값이 재정의됩니다.
- AWS CLI 명령줄에서 파라미터를 사용하여 설정을 지정하면 해당 환경 변수 또는 구성 파일의 프로파일에서 모든 값을 재정의합니다.

환경 변수를 설정하는 방법

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

Linux, macOS, or Unix

```
$ export AWS_ACCESS_KEY_ID=AKIAIOSFODNN7EXAMPLE
$ export AWS_SECRET_ACCESS_KEY=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
$ export
 AWS_SESSION_TOKEN=AQoEXAMPLEH4aoAH0gNCAPy...truncated...zrkuWJOgQs8IZZaIv2BXIa2R4Olgk
$ export AWS_REGION=us-west-2
```

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

Windows Command Prompt

```
C:\> setx AWS_ACCESS_KEY_ID AKIAIOSFODNN7EXAMPLE
C:\> setx AWS_SECRET_ACCESS_KEY wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
```

```
C:\> setx
 AWS_SESSION_TOKEN AQoEXAMPLEH4aoAH0gNCAPy...truncated...zrkuWJOgQs8IZZaIv2BXIa2R40lgk
C:\> setx AWS_REGION us-west-2
```

환경 변수를 설정하는 데 set를 사용하면 사용되는 값이 변경되어 현재 명령 프롬프트 세션이 종료될 때까지 또는 변수를 다른 값으로 설정할 때까지 유지됩니다. 환경 변수를 설정하는 데 setx를 사용하면 현재 명령 프롬프트 세션과 명령 실행 후 생성한 모든 명령 프롬프트 세션에서 사용되는 값이 변경됩니다. 명령을 실행하는 시점에 이미 실행 중인 다른 명령 쉘에는 영향을 주지 않습니다.

PowerShell

```
PS C:\> $Env:AWS_ACCESS_KEY_ID="AKIAIOSFODNN7EXAMPLE"
PS C:\> $Env:AWS_SECRET_ACCESS_KEY="wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY"
PS C:
\> $Env:AWS_SESSION_TOKEN="AQoEXAMPLEH4aoAH0gNCAPy...truncated...zrkuWJOgQs8IZZaIv2BXIa2R40l
PS C:\> $Env:AWS_REGION="us-west-2"
```

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

서버리스 환경 변수 설정

개발에 서버리스 아키텍처를 사용하는 경우 환경 변수를 설정할 수 있는 다른 옵션이 있습니다. 컨테이너에 따라 클라우드가 아닌 환경과 마찬가지로 해당 컨테이너에서 실행되는 코드에 대해 다양한 전략을 사용하여 환경 변수를 보고 액세스할 수 있습니다.

예를 들어를 AWS Lambda사용하면 환경 변수를 직접 설정할 수 있습니다. 자세한 내용은 AWS Lambda 개발자 안내서의 AWS Lambda 환경 변수 사용을 참조하세요.

서버리스 프레임워크에서는 환경 설정 아래의 공급자 키 아래에 있는 `serverless.yml` 파일에 SDK 환경 변수를 설정할 수 있는 경우가 많습니다. `serverless.yml` 파일에 대한 자세한 내용은 서버리스 프레임워크 설명서의 일반 함수 설정을 참조하십시오.

컨테이너 환경 변수를 설정하는 데 사용하는 메커니즘에 관계없이 정의된 런타임 환경 변수에 Lambda에 대해 문서화된 것과 같이 컨테이너에서 예약한 것도 있습니다. 항상 사용 중인 컨테이너의 공식 설명서를 참조하여 환경 변수가 처리되는 방식과 제한이 있는지 확인합니다.

JVM 시스템 속성을 사용하여 전역 구성 AWS SDK for Java 및 AWS SDK for Kotlin

JVM 시스템 속성은 AWS SDK for Java 및와 같이 JVM에서 실행되는 SDKs에 대한 구성 옵션 및 자격 증명을 지정하는 또 다른 방법을 제공합니다 AWS SDK for Kotlin. SDKs. ???

옵션의 우선 순위

- JVM 시스템 속성을 사용하여 설정을 지정하는 경우 환경 변수에서 찾거나 공유 AWS `config` 및 `credentials` 파일의 프로파일에서 로드된 값을 재정의합니다.
- 환경 변수를 사용하여 설정을 지정하면 공유 AWS `config` 및 `credentials` 파일의 프로파일에서 로드된 모든 값이 재정의됩니다.

JVM 시스템 속성을 설정하는 방법

JVM 시스템 속성은 여러 가지 방법으로 설정할 수 있습니다.

명령줄에서

`-D` 스위치를 사용하여 명령을 호출할 때 `java` 명령줄에 JVM 시스템 속성을 설정합니다. 다음 명령은 코드의 값을 명시적으로 재정의하지 않는 한 모든 서비스 클라이언트에 대해 AWS 리전 전역적으로를 구성합니다.

```
java -Daws.region=us-east-1 -jar <your_application.jar> <other_arguments>
```

여러 JVM 시스템 속성을 설정해야 하는 경우 `-D` 스위치를 여러 번 지정합니다.

환경 변수 사용

명령줄에 액세스하여 JVM을 호출하여 애플리케이션을 실행할 수 없는 경우 `JAVA_TOOL_OPTIONS` 환경 변수를 사용하여 명령줄 옵션을 구성할 수 있습니다. 이 접근 방식은 Java 런타임에서 AWS Lambda 함수를 실행하거나 임베디드 JVM에서 코드를 실행하는 등의 상황에서 유용합니다.

다음 예제에서는 코드의 값을 명시적으로 재정의하지 않는 한 모든 서비스 클라이언트에 대해 AWS 리전 전역적으로를 구성합니다.

Linux, macOS, or Unix

```
$ export JAVA_TOOL_OPTIONS="-Daws.region=us-east-1"
```

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

Windows Command Prompt

```
C:\> setx JAVA_TOOL_OPTIONS -Daws.region=us-east-1
```

환경 변수를 설정하는 데 `set`를 사용하면 사용되는 값이 변경되어 현재 명령 프롬프트 세션이 종료될 때까지 또는 변수를 다른 값으로 설정할 때까지 유지됩니다. 환경 변수를 설정하는 데 `setx`를 사용하면 현재 명령 프롬프트 세션과 명령 실행 후 생성한 모든 명령 프롬프트 세션에서 사용되는 값이 변경됩니다. 명령을 실행하는 시점에 이미 실행 중인 다른 명령 쉘에는 영향을 주지 않습니다.

런타임 시

다음 예제와 같이 `System.setProperty` 메서드를 사용하여 런타임 시 코드로 JVM 시스템 속성을 설정할 수도 있습니다.

```
System.setProperty("aws.region", "us-east-1");
```

> **Important**
>
> SDK 서비스 클라이언트를 초기화하기 전에 JVM 시스템 속성을 설정합니다. 그렇지 않으면 서비스 클라이언트가 다른 값을 사용할 수 있습니다.

AWS SDKs 및 도구를 사용한 인증 및 액세스

AWS SDK 애플리케이션을 개발하거나 사용할 AWS 도구를 사용하는 AWS 서비스경우 코드 또는 도구가 인증되는 방법을 설정해야 합니다 AWS. 코드가 실행되는 환경과 사용 가능한 액세스에 따라 다양한 방식으로 AWS 리소스에 대한 프로그래밍 방식 AWS 액세스를 구성할 수 있습니다.

아래 옵션은 자격 증명 공급자 체인의 일부입니다. 즉, 그에 따라 공유 AWS `config` 및 `credentials` 파일을 구성하면 AWS SDK 또는 도구가 해당 인증 방법을 자동으로 검색하고 사용합니다.

애플리케이션 코드를 인증할 메서드 선택

애플리케이션에서 수행한 호출을 인증할 메서 AWS 드를 선택합니다.

코드 INSIDE a AWS 서비스 (예: Amazon EC2, Lambda, Amazon ECS, Amazon EKS, CodeBuild)를 실행하고 있습니까?

코드가 실행되면 애플리케이션에서 자격 AWS증명을 자동으로 사용할 수 있습니다. 예를 들어 애플리케이션이 Amazon Elastic Compute Cloud에서 호스팅되고 해당 리소스와 연결된 IAM 역할이 있는 경우 자격 증명이 애플리케이션에 자동으로 제공됩니다. 마찬가지로 Amazon ECS 또는 Amazon EKS 컨테이너를 사용하는 경우 SDK의 자격 증명 공급자 체인을 통해 컨테이너 내에서 실행되는 코드에서 IAM 역할에 대해 설정된 자격 증명을 자동으로 가져올 수 있습니다.

코드가 Amazon Elastic Compute Cloud 인스턴스에 있습니까?

IAM 역할을 사용하여 Amazon EC2에 배포된 애플리케이션 인증 - Amazon EC2 인스턴스에서 IAM 역할을 사용하여 사용자 애플리케이션을 안전하게 실행합니다.

코드가 AWS Lambda 함수에 있습니까?

Lambda 함수를 생성할 때 Lambda는 최소한의 권한으로 실행 역할을 생성합니다. 그런 다음 AWS SDK 또는 도구는 Lambda 실행 환경을 통해 런타임 시 Lambda에 연결된 IAM 역할을 자동으로 사용합니다.

코드가 Amazon Elastic Container Service(Amazon EC2 또는 Amazon ECS) AWS Fargate 에 있습니까?

작업에 IAM 역할을 사용합니다. 태스크 역할을 생성하고 Amazon ECS 태스크 정의에서 해당 역할을 지정해야 합니다. 그러면 AWS SDK 또는 도구는 Amazon ECS 메타데이터를 통해 런타임에 작업에 할당된 IAM 역할을 자동으로 사용합니다.

코드가 Amazon Elastic Kubernetes Service에 있습니까?

Amazon EKS Pod Identity를 사용하는 것이 좋습니다.

참고: 서비스 계정에 대한 IAM 역할(IRSA)이 고유한 요구 사항에 더 적합할 수 있다고 생각되면 Amazon EKS 사용 설명서의 EKS Pod Identity 및 IRSA 비교를 참조하세요.

코드가에서 실행되고 있습니까? AWS CodeBuild

CodeBuild에 대한 자격 증명 기반 정책 사용을 참조하세요.

코드가 다른에 있습니까 AWS 서비스?

에 대한 전용 가이드를 참조하세요 AWS 서비스. 에서 코드를 실행하면 SDK 자격 증명 공급자 체인 AWS이 자동으로 자격 증명을 가져오고 새로 고칠 수 있습니다.

모바일 애플리케이션 또는 클라이언트 기반 웹 애플리케이션을 생성 중입니까?

액세스가 필요한 모바일 애플리케이션 또는 클라이언트 기반 웹 애플리케이션을 생성하는 경우 AWS 웹 자격 증명 연동을 사용하여 임시 AWS 보안 자격 증명을 동적으로 요청하도록 앱을 빌드합니다.

웹 ID 페더레이션을 사용하면 사용자 정의 로그인 코드를 생성하거나 자신의 사용자 보안 인증을 관리할 필요가 없습니다. 대신에, 앱 사용자는 Login with Amazon, Facebook, Google 또는 다른 OpenID Connect(OIDC)호환 IdP와 같은 널리 알려진 외부 ID 제공업체(idP)를 사용해 로그인할 수 있습니다. 인증 토큰을 받은 다음 해당 토큰을의 리소스를 사용할 권한이 있는 IAM 역할에 매핑 AWS 되는의 임시 보안 자격 증명으로 교환할 수 있습니다 AWS 계정.

SDK 또는 도구에 맞게 이를 구성하는 방법을 알아보려면 웹 자격 증명 또는 OpenID Connect를 사용하여 역할을 수임하여 인증 AWS SDKs 및 도구을 참조하십시오.

모바일 애플리케이션의 경우 Amazon Cognito를 사용하는 것이 좋습니다. Amazon Cognito는 ID 브로커로 활동하며 사용자를 대신하여 상당한 페더레이션을 합니다. 자세한 정보는 IAM 사용 설명서의 모바일 앱용 Amazon Cognito를 참조하십시오.

코드를 로컬로 개발하고 실행하고 있습니까?

를 사용하는 것이 좋습니다IAM Identity Center를 사용하여 AWS SDK 및 도구 인증.

보안 모범 사례로 IAM Identity Center AWS Organizations 와 함께를 사용하여 모든에서 액세스를 관리하는 것이 좋습니다 AWS 계정. 에서 사용자를 생성하거나 AWS IAM Identity Center, Microsoft Active Directory를 사용하거나, SAML 2.0 ID 제공업체(IdP)를 사용하거나, IdP를에 개별적으로 페더

레이션할 수 있습니다 AWS 계정. 사용자 리전에서 Identity Center를 지원하는지 확인하려면 Amazon Web Services 일반 참조.의 AWS IAM Identity Center 엔드포인트 및 할당량을 참조하십시오.

를 제어하지 않고 (사람 개발자로서) AWS OrganizationsAWS IAM Identity Center 를 활성화할 권한이 AWS 계정 있는 경우:

(권장) 대상 역할에에 `sts:AssumeRole` 대한 권한이 있는 최소 권한의 IAM 사용자를 생성합니다. 그런 다음 해당 사용자에 대한 `source_profile` 설정을 사용하여 역할을 수임하도록 프로필을 구성합니다.

환경 변수 또는 공유 AWS `credentials` 파일을 통해 임시 IAM 자격 증명을 사용할 수도 있습니다. 단기 자격 증명을 사용하여 AWS SDKs 및 도구 인증을(를) 참조하세요.

참고: 샌드박스 또는 학습 환경에서만를 고려할 수 있습니다 장기 자격 증명을 사용하여 AWS SDKs 및 도구 인증.

이 코드는 온프레미스 또는 하이브리드/온디맨드 VM(예: Amazon S3에서 읽거나 쓰는 서버 또는 클라우드에 배포하는 Jenkins)에서 실행됩니까?

X.509 클라이언트 인증서를 사용하고 있습니까?

예: 단원을 참조하십시오IAM Roles Anywhere를 사용하여 AWS SDKs 및 도구 인증. IAM Roles Anywhere를 사용하여 외부에서 실행되는 서버, 컨테이너 및 애플리케이션과 같은 워크로드에 대해 IAM에서 임시 보안 자격 증명을 얻을 수 있습니다 AWS. IAM Roles Anywhere를 사용하려면 워크로드에 X.509 인증서를 사용해야 합니다.

환경에서 페더레이션 자격 증명 공급자(예: Microsoft Entra 또는 Okta)에 안전하게 연결하여 임시 AWS 자격 증명을 요청할 수 있습니까?

예: 사용 프로세스 보안 인증 제공자

프로세스 보안 인증 제공자를 사용하여 런타임에 자격 증명을 자동으로 검색합니다. 이러한 시스템은 헬퍼 도구 또는 플러그인을 사용하여 자격 증명을 얻고를 사용하여 백그라운드에서 IAM 역할을 수임할 수 있습니다`sts:AssumeRole`.

아니요:를 통해 주입된 임시 자격 증명 사용 AWS Secrets Manager

를 통해 주입된 임시 자격 증명을 사용합니다 AWS Secrets Manager. 수명이 짧은 액세스 키를 얻는 옵션은 IAM 사용 설명서의 임시 보안 자격 증명 요청을 참조하세요. 이러한 임시 자격 증명을 저장하는 옵션은 섹션을 참조하세요AWS 액세스 키.

이러한 자격 증명을 사용하여 프로덕션 보안 암호 또는 수명이 긴 역할 기반 자격 증명을 저장할 수 있는 Secrets Manager에서 더 광범위한 애플리케이션 권한을 안전하게 검색할 수 있습니다.

에 없는 타사 도구를 사용하고 있습니까 AWS?

자격 증명 획득에 대한 최상의 지침은 타사 공급자가 작성한 설명서를 참조하세요.

타사에서 설명서를 제공하지 않은 경우 임시 자격 증명을 안전하게 주입할 수 있습니까?

예: 환경 변수와 임시 AWS STS 자격 증명을 사용합니다.

아니요: 암호화된 보안 암호 관리자(마지막 수단)에 저장된 정적 액세스 키를 사용합니다.

인증 방법

AWS 환경 내에서 실행되는 코드에 대한 인증 방법

코드가 실행되면 애플리케이션에서 자격 AWS증명을 자동으로 사용할 수 있습니다. 예를 들어 애플리케이션이 Amazon Elastic Compute Cloud에서 호스팅되고 해당 리소스와 연결된 IAM 역할이 있는 경우 자격 증명이 애플리케이션에 자동으로 제공됩니다. 마찬가지로 Amazon ECS 또는 Amazon EKS 컨테이너를 사용하는 경우 SDK의 자격 증명 공급자 체인을 통해 컨테이너 내에서 실행되는 코드에서 IAM 역할에 대해 설정된 자격 증명을 자동으로 가져올 수 있습니다.

- IAM 역할을 사용하여 Amazon EC2에 배포된 애플리케이션 인증 - Amazon EC2 인스턴스에서 IAM 역할을 사용하여 사용자 애플리케이션을 안전하게 실행합니다.
- 다음과 같은 방법으로 IAM Identity Center를 AWS 사용하여 프로그래밍 방식으로와 상호 작용할 수 있습니다.
  - AWS CloudShell를 사용하여 콘솔에서 AWS CLI 명령을 실행합니다.
  - 소프트웨어 개발 팀을 위한 클라우드 기반 협업 공간은 Amazon CodeCatalyst를 고려하십시오.

웹 기반 ID 제공자를 통한 인증 - 모바일 혹은 클라이언트 기반 웹 애플리케이션

액세스가 필요한 모바일 애플리케이션 또는 클라이언트 기반 웹 애플리케이션을 생성하는 경우 AWS 웹 자격 증명 연동을 사용하여 임시 AWS 보안 자격 증명을 동적으로 요청하도록 앱을 빌드합니다.

웹 ID 페더레이션을 사용하면 사용자 정의 로그인 코드를 생성하거나 자신의 사용자 보안 인증을 관리할 필요가 없습니다. 대신에, 앱 사용자는 Login with Amazon, Facebook, Google 또는 다른 OpenID Connect(OIDC)호환 IdP와 같은 널리 알려진 외부 ID 제공업체(idP)를 사용해 로그인할 수 있습니다. 인증 토큰을 받은 다음 해당 토큰을의 리소스를 사용할 권한이 있는 IAM 역할에 매핑 AWS 되는의 임시 보안 자격 증명으로 교환할 수 있습니다 AWS 계정.

SDK 또는 도구에 맞게 이를 구성하는 방법을 알아보려면 웹 자격 증명 또는 OpenID Connect를 사용하여 역할을 수임하여 인증 AWS SDKs 및 도구을 참조하십시오.

모바일 애플리케이션의 경우 Amazon Cognito를 사용하는 것이 좋습니다. Amazon Cognito는 ID 브로커로 활동하며 사용자를 대신하여 상당한 페더레이션을 합니다. 자세한 정보는 IAM 사용 설명서의 모바일 앱용 Amazon Cognito를 참조하십시오.

로컬에서 실행되는 코드에 대한 인증 방법(에 없음 AWS)

- IAM Identity Center를 사용하여 AWS SDK 및 도구 인증 - 보안 모범 사례로 IAM Identity Center AWS Organizations 와 함께를 사용하여 모든에서 액세스를 관리하는 것이 좋습니다 AWS 계정. 에서 사용자를 생성하거나 AWS IAM Identity Center, Microsoft Active Directory를 사용하거나, SAML 2.0 ID 제공업체(IdP)를 사용하거나, IdP를 개별적으로 페더레이션할 수 있습니다 AWS 계정. 사용자 리전에서 Identity Center를 지원하는지 확인하려면 Amazon Web Services 일반 참조.의 AWS IAM Identity Center 엔드포인트 및 할당량을 참조하십시오.
- IAM Roles Anywhere를 사용하여 AWS SDKs 및 도구 인증 - IAM Roles Anywhere를 사용하여 외부에서 실행되는 서버, 컨테이너 및 애플리케이션과 같은 워크로드에 대해 IAM에서 임시 보안 자격 증명을 얻을 수 있습니다 AWS. IAM Roles Anywhere를 사용하려면 워크로드에 X.509 인증서를 사용해야 합니다.
- 자격 AWS 증명이 있는 역할을 수임하여 AWS SDKs 및 도구 인증 - IAM 역할을 수임하여 액세스 권한이 없는 AWS 리소스에 일시적으로 액세스할 수 있습니다.
- AWS 액세스 키를 사용하여 AWS SDKs 및 도구 인증 - 덜 편리하거나 AWS 리소스에 대한 보안 위험을 증가시킬 수 있는 다른 옵션.

액세스 관리에 대한 추가 정보

IAM 사용 설명서에는 AWS 리소스에 대한 액세스를 안전하게 제어하는 방법에 대한 다음 정보가 나와 있습니다.

- IAM 자격 증명(사용자, 사용자 그룹 및 역할) -의 자격 증명 기본 사항을 이해합니다 AWS.
- IAM의 보안 모범 사례 — 공유 책임 모델에 따라 AWS 애플리케이션을 개발할 때 따라야 할 보안 권장 사항.

Amazon Web Services 일반 참조에는 다음에 대한 기본 사항이 있습니다.

- AWS 보안 인증 이해 및 취득 — 콘솔 및 프로그래밍 방식 액세스 모두에 대한 액세스 키 옵션 및 관리 관행.

액세스를 위한 IAM Identity Center 신뢰할 수 있는 자격 증명 전파(TIP) 플러그인 AWS 서비스

- TIP 플러그인을 사용하여 액세스 AWS 서비스 - 신뢰할 수 있는 자격 증명 전파를 지원하는 Amazon Q Business 또는 기타 서비스용 애플리케이션을 생성하고 AWS SDK for Java 또는를 사용하는 경우 AWS SDK for JavaScript간소화된 권한 부여 환경을 위해 TIP 플러그인을 사용할 수 있습니다.

AWS Builder ID

는 이미 소유하거나 생성하려는 모든 AWS 계정 를 AWS Builder ID 보완합니다. 는 사용자가 생성하는 AWS 리소스의 컨테이너 AWS 계정 역할을 하고 해당 리소스에 대한 보안 경계를 제공하지만는 사용자를 개인으로 AWS Builder ID 나타냅니다. 로 로그인하여 Amazon Q 및 Amazon CodeCatalyst와 같은 개발자 도구 및 서비스에 AWS Builder ID 액세스할 수 있습니다.

- AWS 로그인 사용 설명서의 로 로그인 AWS Builder ID -를 생성 및 사용하는 방법과 Builder ID가 제공하는 내용을 AWS Builder ID 알아봅니다.
- CodeCatalyst 개념 - Amazon CodeCatalyst 사용 설명서에서 AWS Builder ID - CodeCatalyst에서 AWS Builder ID를 사용하는 방법을 알아보십시오.

IAM Identity Center를 사용하여 AWS SDK 및 도구 인증

AWS IAM Identity Center 는AWS 컴퓨팅이 아닌 서비스에서 애플리케이션을 개발할 AWS 때 자격 AWS 증명을 제공하는 권장 방법입니다. 예를 들어 이는 사용자의 로컬 개발 환경과 같은 것입니다. Amazon Elastic Compute Cloud(Amazon EC2) 또는 같은 AWS 리소스에서 개발하는 경우 대신 해당 서비스에서 자격 증명을 가져오는 AWS Cloud9것이 좋습니다.

이 자습서에서는 IAM Identity Center 액세스를 설정하고 AWS 액세스 포털 및를 사용하여 SDK 또는 도구에 맞게 구성합니다 AWS CLI.

- AWS 액세스 포털은 IAM Identity Center에 수동으로 로그인하는 웹 위치입니다. URL 형식은 `d-xxxxxxxxxx.awsapps.com/start` 또는 *`your_subdomain`*`.awsapps.com/start`입니다. AWS 액세스 포털에 로그인하면 해당 사용자에 대해 구성된 AWS 계정 및 역할을 볼 수 있습니다. 이 절차에서는 AWS 액세스 포털을 사용하여 SDK/도구 인증 프로세스에 필요한 구성 값을 가져옵니다.
- AWS CLI 는 코드에 의한 API 호출에 IAM Identity Center 인증을 사용하도록 SDK 또는 도구를 구성하는 데 사용됩니다. 이 일회성 프로세스는 공유 AWS `config` 파일을 업데이트한 다음 코드를 실행할 때 SDK 또는 도구에서 사용합니다.

사전 조건

이 절차를 시작하기 전에 다음을 완료해야 합니다.

- 이 없는 경우 AWS 계정에 가입합니다 AWS 계정.
- IAM Identity Center를 아직 활성화하지 않은 경우 AWS IAM Identity Center 사용 설명서의 지침에 따라 IAM Identity Center를 활성화합니다.

IAM Identity Center를 사용하여 프로그래밍 방식 액세스 구성

1 단계: 액세스 설정 및 적절한 권한 세트 선택

자격 AWS 증명에 액세스하려면 다음 방법 중 하나를 선택합니다.

IAM ID 센터를 통한 액세스 권한을 설정하지 않았습니다.

1. 사용 AWS IAM Identity Center 설명서의 기본 IAM Identity Center 디렉터리를 사용하여 사용자 액세스 구성 절차에 따라 사용자를 추가하고 관리 권한을 추가합니다.
2. `AdministratorAccess` 권한 세트는 정기적인 개발에 사용해서는 안 됩니다. 대신 사전 정의된 `PowerUserAccess` 권한 세트를 사용하는 것이 좋습니다. 단, 고용주가 이를 위해 사용자 지정 권한 세트를 생성한 경우는 예외입니다.

   기본 IAM Identity Center 디렉터리 절차를 다시 사용하여 동일한 사용자 액세스 구성 절차를 따르되 이번에는 다음과 같이 합니다.

   - *`Admin team`* 그룹을 생성하는 대신 *`Dev team`* 그룹을 생성하고 지침에서 이를 대체합니다.
   - 기존 사용자를 사용할 수 있지만 사용자를 새 *`Dev team`* 그룹에 추가해야 합니다.
   - *`AdministratorAccess`* 권한 세트를 생성하는 대신 *`PowerUserAccess`* 권한 세트를 생성하고 지침에서 이를 대체합니다.

   작업을 마치면 다음이 필요합니다.

   - `Dev team` 그룹입니다.
   - `Dev team` 그룹에 연결된 `PowerUserAccess` 권한 세트입니다.
   - 사용자가 `Dev team` 그룹에를 추가했습니다.
3. 포털을 종료하고 다시 로그인하여 `Administrator` 또는에 대한 AWS 계정 및 옵션을 확인합니다`PowerUserAccess`. 도구/SDK로 작업할 `PowerUserAccess` 때를 선택합니다.

이미 고용주가 관리하는 페더레이션 ID 제공업체(예: Microsoft Entra 또는 Okta)를 AWS 통해에 액세스할 수 있습니다.

ID 제공업체의 포털을 AWS 통해에 로그인합니다. 클라우드 관리자가 사용자PowerUserAccess(개발자)에게 권한을 부여한 경우 액세스 권한이 AWS 계정 있는 및 권한 세트가 표시됩니다. 권한 집합 이름 옆에는 해당 권한 집합을 사용하여 수동으로 또는 프로그래밍 방식으로 계정에 액세스할 수 있는 옵션이 표시됩니다.

사용자 지정 구현으로 인해 사용 권한 집합 이름이 달라지는 등 다양한 경험이 발생할 수 있습니다. 어떤 권한 세트를 사용할지 확실하지 않은 경우 IT 팀에 문의하세요.

이미 고용주가 관리하는 액세스 포털을 AWS 통해에 AWS 액세스할 수 있습니다.

AWS 액세스 포털을 AWS 통해에 로그인합니다. 클라우드 관리자가 사용자 PowerUserAccess(개발자)에게 권한을 부여한 경우 액세스 권한이 있는 AWS 계정 와 권한 집합이 표시됩니다. 권한 집합 이름 옆에는 해당 권한 집합을 사용하여 수동으로 또는 프로그래밍 방식으로 계정에 액세스할 수 있는 옵션이 표시됩니다.

이미 고용주가 관리하는 페더레이션 사용자 지정 ID 제공업체를 AWS 통해에 액세스할 수 있습니다.

IT 팀에 문의하십시오.

2 단계: IAM Identity Center를 사용할 SDK 및 도구 구성

1. 사용자 개발 시스템에 최신 AWS CLI을 설치합니다.

   a. 자세한 내용은 AWS Command Line Interface 사용 설명서의 AWS CLI최신 버전의 설치 또는 업데이트를 참조하십시오.

   b. (선택 사항) AWS CLI 가 작동하는지 확인하려면 명령 프롬프트를 열고 `aws --version` 명령을 실행합니다.

2. AWS 액세스 포털에 로그인합니다. 사용자의 고용주가 이 URL을 제공할 수도 있으며 1단계: 액세스 설정 후 이메일로 받을 수도 있습니다. 그렇지 않은 경우 https://console.aws.amazon.com/singlesignon/ 대시보드에서 AWS 액세스 포털 URL을 찾습니다.

   a. AWS 액세스 포털의 계정 탭에서 관리할 개별 계정을 선택합니다. 사용자의 역할이 표시됩니다. 액세스 키를 선택하여 적절한 권한 세트에 대한 명령줄 또는 프로그래밍 방식의 액세스에 대한 자격 증명을 가져옵니다. 사전 정의된 PowerUserAccess 권한 세트를 사용하거나, 사용자 또는 고용주가 개발을 위한 최소 권한을 적용하기 위해 생성한 권한 세트를 사용할 수 있습니다.

b. 보안 인증 가져오기 대화 상자에서 운영 체제에 따라 MacOS 및 Linux 또는 Windows를 선택합니다.

c. IAM Identity Center 보안 인증 방법을 선택하여 다음 단계에 필요한 `Issuer URL` 및 `SSO Region` 값을 가져옵니다. 참고:는와 상호 교환하여 사용할 `SSO Start URL` 수 있습니다`Issuer URL`.

3. AWS CLI 명령 프롬프트에서 `aws configure sso` 명령을 실행합니다. 메시지가 표시되면 이전 단계에서 수집한 구성 값을 입력합니다. 이 AWS CLI 명령에 대한 자세한 내용은 `aws configure sso` 마법사를 사용하여 프로필 구성을 참조하세요.

   a. 프롬프트에에 대해 얻은 값을 `SSO Start URL`입력합니다`Issuer URL`.

   b. CLI 프로파일 이름의 경우 시작할 때 `###`을 입력하는 것이 좋습니다. 기본이 아닌 (명명된)프로파일 및 관련 환경 변수를 설정하는 방법에 대한 자세한 내용은 프로파일을 참조하십시오.

4. (선택 사항) AWS CLI 명령 프롬프트에서 `aws sts get-caller-identity` 명령을 실행하여 활성 세션 자격 증명을 확인합니다. 응답에는 구성한 IAM Identity Center 권한 세트가 표시되어야 합니다.

5. AWS SDK를 사용하는 경우 개발 환경에서 SDK용 애플리케이션을 생성합니다.

   a. 일부 SDK의 경우 IAM Identity Center 인증을 사용하려면 먼저 `SSO`과 `SSOOIDC`와 같은 추가 패키지를 애플리케이션에 추가해야 합니다. 자세한 내용은 특정 SDK를 참조하십시오.

   b. 이전에에 대한 액세스를 구성한 경우 공유 AWS `credentials` 파일에서를 AWS검토합니다AWS 액세스 키. 자격 증명 공급자 체인 이해 우선 순위로 인해 SDK 또는 도구에서 IAM Identity Center 보안 인증을 사용하려면 먼저 정적 보안 인증을 반드시 제거해야 합니다.

SDK 및 도구가 이 구성을 사용하여 보안 인증을 사용하고 새로 고치는 방법에 대한 자세한 내용은 AWS SDKs 및 도구에 대한 IAM Identity Center 인증 확인 방법을 참조하십시오.

공유 `config` 파일에서 직접 IAM Identity Center 공급자 설정을 구성하려면이 가이드IAM 아이덴티티 센터 보안 인증 공급자의 섹션을 참조하세요.

포털 액세스 세션 새로 고침

액세스가 결국 만료되고 SDK 또는 도구에 인증 오류가 발생합니다. 이 만료가 발생하는 시기는 구성된 세션 길이에 따라 달라집니다. 필요한 경우 액세스 포털 세션을 다시 새로 고치려면 AWS CLI 를 사용하여 `aws sso login` 명령을 실행합니다.

IAM Identity Center 액세스 포털 세션 기간과 권한 설정 세션 기간을 모두 연장할 수 있습니다. 이렇게 하면 코드를 실행하여 수동으로 다시 로그인해야 하는 시간이 길어집니다. AWS CLI자세한 내용은 AWS IAM Identity Center 사용 설명서에서 다음 주제를 참조하세요.

- IAM Identity Center 세션 기간 - 사용자의 AWS 액세스 포털 세션 기간을 구성합니다
- 권한 설정 세션 기간 - 세션 기간을 설정합니다

AWS SDKs 및 도구에 대한 IAM Identity Center 인증 확인 방법

IAM Identity Center 관련 용어

다음 용어는 기본 AWS IAM Identity Center프로세스와 구성을 이해하는 데 도움이 됩니다. AWS SDK APIs 설명서는 이러한 인증 개념 중 일부에 대해 IAM Identity Center와 다른 이름을 사용합니다. 두 이름을 모두 알고 있으면 도움이 됩니다.

다음 표에서는 이름이 서로 연결되는 방식을 보여줍니다.

IAM Identity Center 이름	SDK API 이름	설명
Identity Center	`sso`	AWS Single Sign-On의 이름이 변경되더라도 `sso` API 네임스페이스는 이전 버전과의 호환성을 위해 원래 이름을 유지합니다. 자세한 내용은 AWS IAM Identity Center 사용 설명서의 IAM Identity Center 이름 변경을 참조하십시오.
IAM Identity Center 콘솔 관리자 콘솔		싱글 사인온을 구성하는 데 사용하는 콘솔입니다.
AWS 액세스 포털 URL		사용자 IAM ID 센터 계정의 고유한 URL(예`https://xxx.awsapps.com/start` :). 사용자 IAM ID 센터 로그인 보안 인증을 사

IAM Identity Center 이름	SDK API 이름	설명
		용하여 이 포털에 로그인합니다.
IAM Identity Center 액세스 포털 세션	인증 세션	발신자에게 베어러 액세스 토큰을 제공합니다.
권한 집합 세션		SDK가 내부적으로 AWS 서비스 호출하는 데 사용하는 IAM 세션입니다. 비공식 토론에서는 이를 '역할 세션'이라고 잘못 지칭하는 것을 볼 수 있습니다.
권한 세트 보안 인증	AWS 자격 증명 sigv4 보안 인증	SDK가 실제로 대부분의 AWS 서비스 호출(특히 모든 sigv4 AWS 서비스 호출)에 사용하는 자격 증명입니다. 비공식 토론에서는 이를 '역할 보안 인증'이라고 잘못 지칭하는 것을 볼 수 있습니다.
IAM 아이덴티티 센터 보안 인증 공급자	SSO 보안 인증 공급자	기능을 제공하는 클래스 또는 모듈 같은 보안 인증을 얻는 방법.

에 대한 SDK 자격 증명 확인 이해 AWS 서비스

IAM ID 센터 API는 베어러 토큰 보안 인증을 sigv4 보안 인증으로 교환합니다. 대부분은 AWS 서비스 및와 같은 몇 가지 예외를 제외하고 Amazon CodeWhisperer sigv4 APIs Amazon CodeCatalyst. 다음은 애플리케이션 코드에 대한 대부분의 AWS 서비스 호출을 지원하기 위한 자격 증명 확인 프로세스를 설명합니다 AWS IAM Identity Center.

AWS 액세스 포털 세션 시작

- 사용자 보안 인증으로 세션에 로그인하여 프로세스를 시작하십시오.
  - AWS Command Line Interface ()에서 `aws sso login` 명령을 사용합니다AWS CLI. 아직 활성 세션이 없는 경우 새 IAM Identity Center 세션이 시작됩니다.

- 새 세션을 시작하면 IAM ID 센터로부터 새로 고침 토큰과 액세스 토큰을 받습니다. AWS CLI 또한는 SSO 캐시 JSON 파일을 새 액세스 토큰 및 새로 고침 토큰으로 업데이트하고 SDKs.
- 이미 활성 세션이 있는 경우 AWS CLI 명령은 기존 세션을 재사용하고 기존 세션이 만료될 때마다 만료됩니다. IAM Identity Center 세션의 길이를 설정하는 방법을 알아보려면 AWS IAM Identity Center 사용 설명서의 사용자의 AWS 액세스 포털 세션 기간 구성을 참조하세요.
  - 자주 로그인해야 하는 필요성을 줄이기 위해 최대 세션 길이가 90일로 연장되었습니다.

SDK가 AWS 서비스 호출에 대한 자격 증명을 가져오는 방법

SDKs 서비스당 클라이언트 객체를 인스턴스화할 AWS 서비스 때에 대한 액세스를 제공합니다. 공유 AWS `config` 파일의 선택한 프로필이 IAM Identity Center 자격 증명 확인을 위해 구성된 경우 IAM Identity Center는 애플리케이션의 자격 증명을 확인하는 데 사용됩니다.

- 보안 인증 확인 프로세스는 클라이언트가 생성되면 런타임 중에 완료됩니다.

IAM ID 센터 싱글 사인온을 사용하여 sigv4 API의 보안 인증을 가져오기 위해 SDK는 IAM ID 센터 액세스 토큰을 사용하여 IAM 세션을 가져옵니다. 이 IAM 세션을 권한 세트 세션이라고 하며, IAM 역할을 수임하여 SDK에 대한 AWS 액세스를 제공합니다.

- 권한 세트 세션 기간은 IAM ID 센터 세션 기간과 별개로 설정됩니다.
  - 권한 설정 세션 기간을 설정하는 방법을 알아보려면 사용 AWS IAM Identity Center 설명서의 세션 기간 설정을 참조하십시오.
- 권한 세트 자격 증명은 대부분의 AWS SDK API 설명서에서 AWS 자격 증명 및 sigv4 자격 증명이라고도 합니다.

IAM ID 센터 API의 GetRoleCredentials를 호출하면 권한 세트 보안 인증이 SDK로 반환됩니다. SDK의 클라이언트 객체는 수임한 IAM 역할을 사용하여 Amazon S3에 계정의 버킷을 나열하도록 요청하는 AWS 서비스등를 호출합니다. 클라이언트 객체는 권한 설정 세션이 만료될 때까지 해당 권한 집합 보안 인증을 사용하여 계속 작동할 수 있습니다.

세션 만료 및 새로 고침

SSO 토큰 공급자 구성를 사용하는 경우 IAM Identity Center에서 가져온 시간별 액세스 토큰은 새로 고침 토큰을 사용하여 자동으로 새로 고쳐집니다.

- SDK가 액세스 토큰을 사용하려고 할 때 액세스 토큰이 만료되면 SDK는 새로 고침 토큰을 사용하여 새 액세스 토큰을 가져오려고 합니다. IAM ID 센터는 새로 고침 토큰을 IAM ID 센터 액세스 포털 세

션 기간과 비교합니다. 새로 고침 토큰이 만료되지 않은 경우 IAM ID 센터는 다른 액세스 토큰으로 응답합니다.

- 이 액세스 토큰은 기존 클라이언트의 권한 설정 세션을 새로 고치거나 새 클라이언트의 보안 인증을 확인하는 데 사용할 수 있습니다.

하지만 IAM Identity Center 액세스 포털 세션이 만료되면 새 액세스 토큰이 부여되지 않습니다. 따라서 권한 세트 기간은 갱신할 수 없습니다. 캐시된 권한 세트 세션 기간이 초과되면 기존 클라이언트가 만료되고 액세스를 하지 못합니다.

새 클라이언트를 생성하는 모든 코드는 IAM Identity Center 세션이 만료되는 즉시 인증에 실패합니다. 권한 세트 보안 인증이 캐시되지 않기 때문입니다. 유효한 액세스 토큰을 확보하기 전까지는 코드를 사용하여 새 클라이언트를 만들고 보안 인증 확인 프로세스를 완료할 수 없습니다.

요약하자면, SDK에 새 권한 집합 보안 인증이 필요한 경우 SDK는 먼저 유효한 기존 보안 인증을 확인하고 이를 사용합니다. 이는 보안 인증이 새 클라이언트용이든 보안 인증이 만료된 기존 클라이언트용이든 상관없이 적용됩니다. 보안 인증을 찾을 수 없거나 유효하지 않은 경우 SDK는 IAM Identity Center API를 호출하여 새 보안 인증을 가져옵니다. API를 호출하려면 액세스 토큰이 필요합니다. 액세스 토큰이 만료되면 SDK는 새로 고침 토큰을 사용하여 IAM Identity Center 서비스로부터 새 액세스 토큰을 가져오려고 시도합니다. 이 토큰은 IAM Identity Center 액세스 포털 세션이 만료되지 않은 경우 부여됩니다.

IAM Roles Anywhere를 사용하여 AWS SDKs 및 도구 인증

IAM Roles Anywhere를 사용하여 외부에서 실행되는 서버, 컨테이너 및 애플리케이션과 같은 워크로드에 대해 IAM에서 임시 보안 자격 증명을 가져올 수 있습니다 AWS. IAM Roles Anywhere를 사용하려면 워크로드에 X.509 인증서를 사용해야 합니다. 클라우드 관리자는 IAM Roles Anywhere를 보안 인증 공급자로 구성하는 데 필요한 인증서와 프라이빗 키를 제공해야 합니다.

1단계: IAM Roles Anywhere 구성

IAM Roles Anywhere는 외부에서 실행되는 워크로드 또는 프로세스에 대한 임시 자격 증명을 가져오는 방법을 제공합니다 AWS. 인증 권한에 트러스트 앵커를 설정하여 관련 IAM 역할에 대한 임시 보안 인증을 가져옵니다. 역할에서 코드가 IAM Roles Anywhere로 인증될 때 워크로드가 갖게 될 권한을 설정합니다.

신뢰 앵커, IAM 역할 및 IAM Roles Anywhere 프로파일을 설정하는 단계는 IAM Roles Anywhere 사용 설명서의 AWS Identity and Access Management Roles Anywhere에서 신뢰 앵커 및 프로파일 생성을 참조하세요.

> **Note**
>
> IAM Roles Anywhere 사용 설명서의 프로파일은 IAM Roles Anywhere 서비스 내의 고유한 개념을 말합니다. 공유 AWS `config` 파일 내의 프로필과는 관련이 없습니다.

2단계: IAM Roles Anywhere 사용

IAM Roles Anywhere에서 임시 보안 인증을 가져오려면 IAM Roles Anywhere에서 제공하는 보안 인증 도우미 도구를 사용합니다. 보안 인증 도구는 IAM Roles Anywhere의 서명 프로세스를 구현합니다.

자격 증명 헬퍼 도구를 다운로드하는 지침은 IAM AWS Identity and Access Management Roles Anywhere 사용 설명서의 Roles Anywhere에서 임시 보안 자격 증명 받기를 참조하세요.

IAM Roles Anywhere의 임시 보안 자격 증명을 AWS SDKs 및와 함께 사용하려면 공유 AWS `config` 파일에서 `credential_process` 설정을 구성할 AWS CLI수 있습니다. SDKs 및는가 인증에 사용하는 프로세스 자격 증명 공급자`credential_process`를 AWS CLI 지원합니다. 다음은 `credential_process` 설정의 일반 구조를 보여줍니다.

```
credential_process = [path to helper tool] [command] [--parameter1 value] [--
parameter2 value] [...]
```

도우미 도구의 `credential-process` 명령은 `credential_process` 설정과 호환되는 표준 JSON 형식의 임시 보안 인증을 반환합니다. 명령 이름에는 하이픈이 포함되지만 설정 이름에는 밑줄이 포함된다는 점에 유의하십시오. 명령에는 다음과 같은 파라미터를 요구합니다.

- `private-key` - 요청에 서명한 개인 키의 경로.
- `certificate` - 보안 인증의 경로.
- `role-arn` - 임시 보안 인증을 가져올 역할의 ARN.
- `profile-arn` - 지정된 역할에 대한 매핑을 제공하는 프로파일의 ARN.
- `trust-anchor-arn` - 트러스트 앵커의 ARN.

클라우드 관리자가 인증서와 프라이빗 키를 제공해야 합니다 AWS Management Console에서 세 개의 ARN 값을 모두 복사할 수 있습니다. 다음 예는 도우미 도구에서 임시 보안 인증을 검색하도록 구성하는 공유 `config` 파일을 보여줍니다.

```
[profile dev]
```

```
credential_process = ./aws_signing_helper credential-process --certificate /
path/to/certificate --private-key /path/to/private-key --trust-anchor-
arn arn:aws:rolesanywhere:region:account:trust-anchor/TA_ID --profile-
arn arn:aws:rolesanywhere:region:account:profile/PROFILE_ID --role-
arn arn:aws:iam::account:role/ROLE_ID
```

선택적 파라미터 및 추가 도우미 도구 세부 정보는 GitHub의 IAM Roles Anywhere 보안 인증 도우미를 참조하십시오.

SDK 구성 설정 자체 및 프로세스 보안 인증 공급자에 대한 자세한 내용은 이 설명서의 프로세스 보안 인증 제공자을 참조하십시오.

자격 AWS 증명이 있는 역할을 수임하여 AWS SDKs 및 도구 인증

역할 수임에는 액세스 권한이 없을 수 있는 AWS 리소스에 액세스하기 위해 일련의 임시 보안 보안 인증을 사용하는 것이 포함됩니다. 이러한 임시 보안 인증은 액세스 키 ID, 보안 액세스 키 및 보안 토큰으로 구성됩니다. AWS Security Token Service (AWS STS) API 요청에 대해 자세히 알아보려면AWS Security Token Service API 참조의 작업을 참조하세요.

역할을 수임하도록 SDK 또는 도구를 설정하려면 먼저 수임할 특정 역할을 만들거나 식별해야 합니다. IAM 역할은 Amazon 리소스 이름(ARN)역할로 고유하게 식별됩니다. 역할은 다른 엔티티와 신뢰 관계를 구축합니다. 역할을 사용하는 신뢰할 수 있는 엔터티는 AWS 서비스 또는 다른 엔터티일 수 있습니다 AWS 계정. IAM 역할에 대한 자세한 내용은 IAM 사용 설명서의 IAM 역할 섹션을 참조하십시오.

IAM 역할을 식별한 후 해당 역할을 신뢰할 수 있는 경우 해당 역할에서 부여한 권한을 사용하도록 SDK 또는 도구를 구성할 수 있습니다.

> **Note**
>
> 가능하면 리전 엔드포인트를 사용하고를 구성하는 것이 AWS 가장 좋습니다AWS 리전.

IAM 역할 수임

역할을 수임할 때는 임시 보안 자격 증명 세트를 AWS STS 반환합니다. 이러한 보안 인증은 다른 프로파일이나 코드가 실행되는 인스턴스 또는 컨테이너에서 제공됩니다. 일반적으로 이러한 유형의 역할 수임은 한 계정에 대한 AWS 자격 증명이 있지만 애플리케이션에서 다른 계정의 리소스에 액세스해야 할 때 사용됩니다.

1단계: IAM 역할 설정

역할을 수임하도록 SDK 또는 도구를 설정하려면 먼저 수임할 특정 역할을 만들거나 식별해야 합니다. IAM 역할은 역할 ARN을 사용하여 고유하게 식별됩니다. 역할은 일반적으로 계정 내에서 또는 크로스 계정 액세스를 위해 다른 엔티티와 신뢰 관계를 구축합니다. 자세한 내용은 IAM 사용 설명서의 IAM 역할 생성을 참조하십시오.

2단계: SDK 또는 도구 구성

`credential_source` 또는 `source_profile`에서 보안 인증을 소싱하도록 SDK 또는 도구를 구성합니다.

Amazon ECS 컨테이너, Amazon EC2 인스턴스 또는 환경 변수에서 보안 인증을 소싱하는 데 `credential_source`을(를) 사용합니다.

다른 프로파일에서 보안 인증을 소싱하는 데 `source_profile`을(를) 사용합니다. `source_profile`은 또한 수임된 역할을 사용하여 다른 역할을 수임하는 프로파일 계층 구조인 역할 체인을 지원합니다.

프로필에서 이를 지정하면 SDK 또는 도구가 자동으로 해당 AWS STS `AssumeRole` API 호출을 수행합니다. 역할을 수임하여 임시 자격 증명을 검색하고 사용하려면 공유 AWS `config` 파일에 다음 구성 값을 지정합니다. 이러한 설정에 대한 자세한 내용은 역할 보안 인증 제공자 수임 설정 섹션을 참조하십시오.

- `role_arn` - 1단계에서 생성한 IAM 역할에서
- `credential_source` 또는 `source_profile` 중 하나를 구성
- (선택 사항) `duration_seconds`
- (선택 사항) `external_id`
- (선택 사항) `mfa_serial`
- (선택 사항) `role_session_name`

다음 예는 공유 `config` 파일에서 두 역할 수임 옵션 모두의 구성을 보여줍니다.

```
role_arn = arn:aws:iam::123456789012:role/my-role-name
credential_source = Ec2InstanceMetadata
```

```
[profile-with-user-that-can-assume-role]
aws_access_key_id=AKIAIOSFODNN7EXAMPLE
```

```
aws_secret_access_key=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
aws_session_token=IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZVERYLONGSTRI

[profile dev]
region = us-east-1
output = json
role_arn = arn:aws:iam::123456789012:role/my-role-name
source_profile = profile-with-user-that-can-assume-role
role_session_name = my_session
```

모든 역할 수임 보안 인증 공급자 설정에 대한 자세한 내용은 이 안내서의 역할 보안 인증 제공자 수임를 참조하세요.

웹 자격 증명 또는 OpenID Connect를 사용하여 역할을 수임하여 인증 AWS SDKs 및 도구

역할 수임에는 액세스 권한이 없을 수 있는 AWS 리소스에 액세스하기 위해 일련의 임시 보안 보안 인증을 사용하는 것이 포함됩니다. 이러한 임시 보안 인증은 액세스 키 ID, 보안 액세스 키 및 보안 토큰으로 구성됩니다. AWS Security Token Service (AWS STS) API 요청에 대해 자세히 알아보려면AWS Security Token Service API 참조의 작업을 참조하세요.

역할을 수임하도록 SDK 또는 도구를 설정하려면 먼저 수임할 특정 역할을 만들거나 식별해야 합니다. IAM 역할은 Amazon 리소스 이름(ARN)역할로 고유하게 식별됩니다. 역할은 다른 엔티티와 신뢰 관계를 구축합니다. 역할을 사용하는 신뢰할 수 있는 엔터티는 웹 자격 증명 공급자 또는 OpenID Connect(OIDC) 또는 SAML 페더레이션일 수 있습니다. IAM 역할에 대한 자세한 내용은 IAM 사용 설명서의 역할을 수임하는 방법을 참조하세요.

SDK에서 IAM 역할을 구성한 후 해당 역할이 자격 증명 공급자를 신뢰하도록 구성된 경우 임시 AWS 자격 증명을 얻기 위해 해당 역할을 수임하도록 SDK를 추가로 구성할 수 있습니다.

> **Note**
>
> 가능하면 리전 엔드포인트를 사용하고를 구성하는 것이 AWS 가장 좋습니다AWS 리전.

웹 자격 증명 또는 OpenID Connect와 페더레이션

Login With Amazon, Facebook, Google과 같은 퍼블릭 자격 증명 공급자의 JSON 웹 토큰(JWTs)을 사용하여를 사용하여 임시 AWS 자격 증명을 가져올 수 있습니다`AssumeRoleWithWebIdentity`.

사용 방법에 따라 이러한 JWTs ID 토큰 또는 액세스 토큰이라고 할 수 있습니다. EntraId 또는 PingFederate와 같은 OIDC의 검색 프로토콜과 호환되는 ID 제공업체(IdPs)에서 발급한 JWTs를 사용할 수도 있습니다.

Amazon Elastic Kubernetes Service를 사용하는 경우이 기능은 Amazon EKS 클러스터의 각 서비스 계정에 대해 서로 다른 IAM 역할을 지정하는 기능을 제공합니다. 이 Kubernetes 기능은 JWTs 포드에 배포한 다음이 자격 증명 공급자가 임시 AWS 자격 증명을 얻는 데 사용합니다. 이 Amazon EKS 구성에 대한 자세한 내용은 Amazon EKS 사용 설명서의 서비스 계정에 대한 IAM 역할을 참조하세요. 그러나 더 간단한 옵션을 원하면 SDK에서 지원하는 경우 Amazon EKS Pod Identitie를 대신 사용하는 것이 좋습니다.

1단계: 보안 인증 공급자 및 IAM 역할 설정

외부 IdP와의 페더레이션을 구성하려면 IAM 자격 증명 공급자를 사용하여 외부 IdP 및 해당 구성에 AWS 대해 알립니다. 이렇게 하면 AWS 계정 와 외부 IdP 간에 신뢰가 설정됩니다. 인증에 JSON 웹 토큰(JWT)을 사용하도록 SDK를 구성하기 전에 먼저 자격 증명 공급자(IdP)와 여기에 액세스하는 데 사용되는 IAM 역할을 설정해야 합니다. 이러한 설정을 하려면 IAM 사용 설명서의 웹 보안 인증 또는 OpenID Connect 페더레이션을 위한 역할 생성(콘솔)을 참조하십시오.

2단계: SDK 또는 도구 구성

AWS STS 인증을 위해의 JSON 웹 토큰(JWT)을 사용하도록 SDK 또는 도구를 구성합니다.

프로필에서 이를 지정하면 SDK 또는 도구가 자동으로 해당 AWS STS `AssumeRoleWithWebIdentity` API를 직접 호출합니다. 웹 자격 증명 연동을 사용하여 임시 자격 증명을 검색하고 사용하려면 공유 AWS `config` 파일에서 다음 구성 값을 지정합니다. 이러한 설정에 대한 자세한 내용은 역할 보안 인증 제공자 수임 설정 섹션을 참조하십시오.

- `role_arn` - 1단계에서 생성한 IAM 역할에서
- `web_identity_token_file`- 외부 IdP에서
- (선택 사항) `duration_seconds`
- (선택 사항) `role_session_name`

다음은 웹 ID를 사용하여 역할을 수임하는 공유 `config` 파일 구성의 예입니다.

```
[profile web-identity]
role_arn=arn:aws:iam::123456789012:role/my-role-name
web_identity_token_file=/path/to/a/token
```

> **Note**
>
> 모바일 애플리케이션의 경우 Amazon Cognito를 사용하는 것이 좋습니다. Amazon Cognito는 ID 브로커로 활동하며 사용자를 대신하여 상당한 페더레이션을 합니다. 하지만 Amazon Cognito 보안 인증 공급자는 다른 보안 인증 공급자처럼 SDK 및 도구 코어 라이브러리에 포함되지 않습니다. Amazon Cognito API에 액세스하려면 SDK 또는 도구용 빌드 또는 라이브러리에 Amazon Cognito 서비스 클라이언트를 포함시키십시오. AWS SDKs와 함께 사용하려면 Amazon Cognito 개발자 안내서의 코드 예제를 참조하세요.

모든 역할 수임 보안 인증 공급자 설정에 대한 자세한 내용은 이 안내서의 역할 보안 인증 제공자 수임를 참조하세요.

AWS 액세스 키를 사용하여 AWS SDKs 및 도구 인증

AWS 액세스 키 사용은 AWS SDKs.

단기 보안 인증 정보를 사용합니다

연장된 세션 기간 옵션 사용에 IAM Identity Center를 사용하여 AWS SDK 및 도구 인증을 사용하도록 SDK 또는 도구를 구성하는 것이 좋습니다.

하지만 SDK 또는 도구의 임시 보안 인증을 직접 설정하려면 단기 자격 증명을 사용하여 AWS SDKs 및 도구 인증을 참조하십시오.

장기 보안 인증 정보 사용

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

에서 액세스 관리 AWS 계정

보안 모범 사례로 IAM Identity Center와 AWS Organizations 함께를 사용하여 모든에서 액세스를 관리하는 것이 좋습니다 AWS 계정. 자세한 내용은 IAM 사용 설명서의 IAM의 보안 모범 사례를 참조하십시오.

IAM Identity Center에서 사용자를 생성하거나, Microsoft Active Directory를 사용하거나, SAML 2.0 자격 증명 공급자(IdP)를 사용하거나, IdP를 개별적으로 페더레이션할 수 있습니다 AWS 계정. 이러한 접근 방식 중 하나를 사용하면 사용자에게 Single Sign-On 경험을 제공할 수 있습니다. 다중 인증(MFA)을 적용하고 AWS 계정 액세스에 임시 자격 증명을 사용할 수도 있습니다. 이는 공유할 수 있는 장기 보안 인증 정보이며 AWS 리소스에 대한 보안 위험을 증가시킬 수 있는 IAM 사용자와는 다릅니다.

샌드박스 환경에서만 사용할 IAM 사용자 생성

를 처음 사용하는 경우 테스트 IAM 사용자를 AWS생성한 다음 이를 사용하여 자습서를 실행하고 제공해야 할 사항을 살펴볼 수 AWS 있습니다. 학습 중에는 이러한 유형의 보안 인증 정보를 사용해도 괜찮지만 샌드박스 환경 밖에서는 사용하지 않는 것이 좋습니다.

다음 사용 사례의 경우에서 IAM 사용자를 시작하는 것이 합리적일 수 있습니다. AWS

- AWS SDK 또는 도구를 시작하고 AWS 서비스 샌드박스 환경에서 탐색합니다.
- 사람이 직접 진행하는 로그인 프로세스를 지원하지 않는 예약된 스크립트, 작업 및 기타 자동화된 프로세스를 학습의 일부로 실행하세요.

이러한 사용 사례 외부에서 IAM 사용자를 사용하는 경우 가능한 한 AWS 계정 빨리 IAM Identity Center로 전환하거나 자격 증명 공급자를 로 페더레이션합니다. 자세한 내용은 AWS에서 자격 증명 공급자 및 페더레이션을 참조하세요.

보안 IAM 사용자 액세스 키

IAM 사용자 액세스 키는 정기적으로 교체해야 합니다. IAM 사용자 설명서의 액세스 키 교체에 있는 지침을 따르세요. 실수로 IAM 사용자 액세스 키를 공유했다고 생각되면 액세스 키를 교체하세요.

IAM 사용자 액세스 키는 로컬 시스템의 공유 AWS `credentials` 파일에 저장해야 합니다. 코드에 IAM 사용자 액세스 키를 저장하지 마세요. IAM 사용자 액세스 키가 포함된 구성 파일을 소스 코드 관리 소프트웨어에 포함시키지 마세요. 오픈 소스 프로젝트 git-secrets와 같은 외부 도구를 사용하면 중요한 정보를 실수로 Git 리포지토리에 커밋하는 것을 방지할 수 있습니다. 자세한 내용은 IAM 사용자 설명서의 IAM ID(사용자, 그룹 및 역할)을 참조하십시오.

IAM 사용자 시작을 설정하려면 장기 자격 증명을 사용하여 AWS SDKs 및 도구 인증을 참조하십시오.

단기 자격 증명을 사용하여 AWS SDKs 및 도구 인증

확장 세션 기간 옵션과 IAM Identity Center를 사용하여 AWS SDK 및 도구 인증 함께 사용하도록 AWS SDK 또는 도구를 구성하는 것이 좋습니다. 그러나 AWS 액세스 포털에서 사용할 수 있는 임시 자격 증

명을 복사하고 사용할 수 있습니다. 보안 인증이 만료되면 새 보안 인증을 복사해야 합니다. 프로파일에서 임시 보안 인증을 사용하거나 이를 시스템 속성 및 환경 변수의 값으로 사용할 수 있습니다.

모범 사례: 보안 인증 파일에서 액세스 키와 토큰을 수동으로 관리하는 대신 애플리케이션에서 다음에서 전달되는 임시 보안 인증을 사용하는 것이 좋습니다.

- Amazon Elastic Compute Cloud 또는에서 애플리케이션을 실행하는 것과 같은 컴퓨팅 AWS 서비스입니다 AWS Lambda.
- 와 같은 자격 증명 공급자 체인의 또 다른 옵션입니다IAM Identity Center를 사용하여 AWS SDK 및 도구 인증.
- 또는 프로세스 보안 인증 제공자를 사용하여 임시 자격 증명을 검색합니다.

AWS 액세스 포털에서 검색된 단기 보안 인증 정보를 사용하여 보안 인증 파일 설정

1. 공유 보안 인증 파일 생성.
2. 보안 인증 파일에 임시 동작 보안 인증을 붙여 넣을 때까지 다음 자리 표시자 텍스트를 붙여 넣습니다.

```
[default]
aws_access_key_id=<value from AWS access portal>
aws_secret_access_key=<value from AWS access portal>
aws_session_token=<value from AWS access portal>
```

3. 파일을 저장합니다. 이제 `~/.aws/credentials`파일이 로컬 개발 시스템에 존재해야 합니다. 이 파일에는 이름이 지정된 특정 프로파일이 지정되지 않은 경우 SDK 또는 도구에서 사용하는 [기본] 프로파일이 들어 있습니다.
4. AWS 액세스 포털에 로그인합니다.
5. 수동 자격 증명 새로 고침에 대한 다음 지침에 따라 AWS 액세스 포털에서 IAM 역할 자격 증명을 복사합니다.

   a. 링크된 지침의 4단계에서 개발 요건에 액세스 권한을 부여하는 IAM 역할 이름을 선택합니다. 이 역할은 일반적으로 PowerUserAccess 또는 Developer와 같은 이름으로 되어 있습니다.

   b. 링크된 지침 7 단계에서 AWS 보안 인증 파일에 수동으로 프로파일 추가 옵션을 선택하고 내용을 복사합니다.

6. 복사한 보안 인증을 로컬 `credentials` 파일에 붙여 넣습니다. `default` 프로파일을 사용하는 경우 생성된 프로파일 이름은 필요하지 않습니다. 파일은 다음과 유사해야 합니다.

```
[default]
aws_access_key_id=AKIAIOSFODNN7EXAMPLE
aws_secret_access_key=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
aws_session_token=IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZVERYLONG
```

7. `credentials` 파일을 저장합니다.

SDK는 서비스 클라이언트를 생성할 때 이러한 임시 보안 인증에 액세스하여 각 요청에 사용합니다. 5a단계에서 선택한 IAM 역할 설정에 따라 임시 보안 인증의 유효 기간이 결정됩니다. 최대 유효 기간은 12시간입니다.

임시 보안 인증이 만료되면 4~7단계를 반복합니다.

장기 자격 증명을 사용하여 AWS SDKs 및 도구 인증

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

IAM 사용자를 사용하여 코드를 실행하는 경우 개발 환경의 SDK 또는 도구는 공유 AWS `credentials` 파일의 장기 IAM 사용자 자격 증명을 사용하여 인증합니다. IAM의 보안 모범 사례 주제를 검토하고 가능한 한 빨리 IAM Identity Center 또는 기타 임시 보안 인증으로 전환하십시오.

보안 인증에 대한 중요 경고 및 지침

보안 인증에 대한 경고

- 금지 사항. AWS 리소스에 액세스할 때는 계정의 루트 보안 인증을 사용해서는 안 됩니다. 이 보안 인증은 계정 액세스에 제한이 없고 취소하기 어렵습니다.
- 금지 사항. 애플리케이션 파일에 리터럴 액세스 키나 보안 인증 정보를 넣지 않습니다. 이를 어기는 경우, 예를 들어 프로젝트를 퍼블릭 리포지토리에 업로드하면 뜻하지 않게 보안 인증이 노출될 위험이 있습니다.
- 금지 사항. 프로젝트 영역에 보안 인증이 포함된 파일을 포함하지 마십시오.
- 공유 AWS `credentials` 파일에 저장된 모든 자격 증명은 일반 텍스트로 저장됩니다.

보안 인증 정보를 안전하게 관리하기 위한 추가 지침

보안 AWS 인증을 안전하게 관리하는 방법에 대한 일반적인 설명은의 AWS 액세스 키 관리 모범 사례를 참조하세요AWS 일반 참조. 해당 설명과 더불어 다음 사항을 고려하십시오.

- Amazon Elastic Container Service(Amazon ECS) 작업에 작업용 IAM 역할을 사용하십시오.
- Amazon EC2 인스턴스에서 실행 중인 애플리케이션에 IAM 역할을 사용하십시오.

사전 조건: AWS 계정 생성

IAM 사용자를 사용하여 AWS 서비스에 액세스하려면 AWS 계정과 AWS 자격 증명이 필요합니다.

1. 계정을 생성합니다.

   AWS 계정을 생성하려면 AWS Account Management 참조 안내서의 시작하기: 처음 AWS 사용하십니까?를 참조하세요.
2. 관리 사용자를 생성합니다.

   관리 콘솔 및 서비스에 액세스하기 위해 루트 사용자 계정(사용자가 생성하는 초기 계정)을 사용하지 않습니다. 대신 IAM 사용 설명서의 관리 사용자 생성에 설명된 대로 관리 사용자 계정을 생성합니다.

   관리 사용자 계정을 만들고 로그인 세부 정보를 기록한 후 반드시 루트 사용자 계정에서 로그아웃하고 관리 계정을 사용하여 다시 로그인합니다.

이러한 계정 중 어느 것도에서 개발을 수행 AWS 하거나에서 애플리케이션을 실행하는 데 적합하지 않습니다 AWS. 모범 사례로서 이러한 작업에 적합한 사용자, 권한 집합 및 서비스 역할을 만들어야 합니다. 자세한 정보는 IAM 사용 설명서의 최소 권한 적용을 참조하십시오.

1단계: IAM 사용자 생성

- IAM 사용 설명서의 IAM 사용자 생성(콘솔)절차에 따라 IAM 사용자를 생성합니다. IAM 사용자를 생성할 때:
  - 에 대한 사용자 액세스 권한 제공을 AWS Management Console 선택하는 것이 좋습니다. 이를 통해 진단 로그 확인 AWS CloudTrail 또는 Amazon Simple Storage Service에 파일 업로드와 같이 시각적 환경에서 실행 중인 코드와 AWS 서비스 관련된를 볼 수 있습니다. 이는 코드를 디버깅할 때 유용합니다.

- 권한 설정 - 권한 옵션에서이 사용자에게 권한을 할당할 방법에 대한 정책 직접 연결을 선택합니다.
  - 대부분의 “시작하기” SDK 자습서에서는 Amazon S3 서비스를 예로 사용합니다. 애플리케이션에 Amazon S3에 대한 전체 액세스 권한을 제공하려면 이 사용자에게 연결할 `AmazonS3FullAccess` 정책을 선택하십시오.
- 권한 경계 또는 태그 설정과 관련하여 해당 절차의 선택적 단계를 무시할 수 있습니다.

2단계: 액세스 키 가져오기

1. IAM 콘솔의 탐색 창에서 사용자를 선택한 다음 이전에 생성한 사용자의 **`User name`**를 선택합니다.
2. 사용자 페이지에서 보안 보안 인증 페이지를 선택합니다. 그런 다음 액세스 키에서 액세스 키 생성을 선택합니다.
3. 액세스 키 생성 1단계에서 명령줄 인터페이스(CLI)또는 로컬 코드를 선택합니다. 두 옵션 모두 AWS CLI 및 SDKs 모두에 사용할 동일한 유형의 키를 생성합니다.
4. 액세스 키 만들기 2단계에서 선택적 태그를 입력하고 다음을 선택합니다.
5. 액세스 키 생성 3단계에서 .csv 파일 다운로드를 선택하여 IAM 사용자의 액세스 키 및 보안 액세스 키와 함께 `.csv` 파일을 저장합니다. 나중에 이 정보가 필요합니다.

   > **Warning**
   >
   > 적절한 보안 조치를 사용하여 이러한 보안 인증을 안전하게 유지합니다.

6. 완료(Done)를 선택합니다.

3단계: **`credentials`** 파일 업데이트

1. 공유 AWS `credentials` 파일을 생성하거나 엽니다. 이 파일은 Linux 및 macOS 시스템의 경우 `~/.aws/credentials`이며, Windows의 경우 `%USERPROFILE%\.aws\credentials`입니다. 자세한 내용은 보안 인증 파일 위치를 참조하십시오.
2. 다음 텍스트를 공유 `credentials` 파일에 추가합니다. 예제 ID 값 및 예제 키 값을 이전에 다운로드한 `.csv` 파일의 값으로 바꾸십시오.

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

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

3. 파일을 저장합니다.

공유 `credentials` 파일은 보안 인증을 저장하는 가장 일반적인 방법입니다. 환경 변수로 설정할 수도 있습니다. 환경 변수 이름은 AWS 액세스 키 섹션을 참조하십시오. 이 방법으로 시작할 수 있지만 가능한 한 빨리 IAM Identity Center 또는 기타 임시 보안 인증으로 전환하는 것이 좋습니다. 장기 보안 인증을 사용하지 않도록 전환한 후에는 공유 `credentials` 파일에서 해당 보안 인증을 삭제해야 합니다.

IAM 역할을 사용하여 Amazon EC2에 배포된 애플리케이션 인증

이 예제에서는 Amazon Elastic Compute Cloud 인스턴스에 배포된 애플리케이션에서 사용할 Amazon S3 액세스 권한이 있는 AWS Identity and Access Management 역할을 설정하는 방법을 다룹니다.

Amazon Elastic Compute Cloud 인스턴스에서 AWS SDK 애플리케이션을 실행하려면 IAM 역할을 생성한 다음 Amazon EC2 인스턴스에 해당 역할에 대한 액세스 권한을 부여합니다. 자세한 내용은 Amazon EC2 사용 설명서의 Amazon EC2의 IAM 역할을 참조하세요.

IAM 역할 생성

개발하는 AWS SDK 애플리케이션은 작업을 수행하기 AWS 서비스 위해 하나 이상에 액세스할 수 있습니다. 애플리케이션을 실행하는 데 필요한 권한을 부여하는 IAM 역할을 생성합니다.

이 절차에서는 Amazon S3에 대한 읽기 전용 액세스 권한을 예제로 부여하는 역할을 생성합니다. 많은 AWS SDK 가이드에는 Amazon S3에서 읽는 "시작" 자습서가 있습니다.

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/iam/ IAM 콘솔을 엽니다.
2. 탐색 창에서 역할을 선택한 다음 역할 생성을 선택합니다.
3. 신뢰할 수 있는 엔터티 선택을 위해 신뢰할 수 있는 엔터티 유형 아래에서 AWS 서비스 서비스를 선택합니다.
4. 사용 사례에서 Amazon EC2를 선택한 후 다음을 선택합니다.
5. 권한 추가의 경우 정책 목록에서 Amazon S3 읽기 전용 액세스 확인란을 선택한 후 다음을 선택합니다.
6. 역할 이름을 입력한 다음 역할 생성을 선택합니다. Amazon EC2 인스턴스를 생성할 때 필요하므로이 이름을 기억하세요.

Amazon EC2 인스턴스 시작과 IAM 역할 지정

다음을 수행하여 IAM 역할을 사용하여 Amazon EC2 인스턴스를 생성하고 시작할 수 있습니다.

- Amazon EC2 사용 설명서의 인스턴스 빠른 시작을 따릅니다. 그러나 최종 제출 단계 전에 다음 작업도 수행합니다.
  - 고급 세부 정보의 IAM 인스턴스 프로파일에서 이전 단계에서 생성한 역할을 선택합니다.

이 IAM 및 Amazon EC2 설정을 사용하면 애플리케이션을 Amazon EC2 인스턴스에 배포할 수 있으며 애플리케이션은 Amazon S3 서비스에 대한 읽기 액세스 권한을 갖게 됩니다.

EC2 인스턴스에 연결

애플리케이션을 Amazon EC2 인스턴스로 전송한 다음 애플리케이션을 실행할 수 있도록 Amazon EC2 인스턴스에 연결합니다. 인스턴스를 생성할 때 키 페어(로그인)에서 사용한 키 페어의 프라이빗 부분, 즉 PEM 파일이 포함된 파일이 필요합니다.

인스턴스 유형에 대한 지침인 Linux 인스턴스에 연결 또는 Windows 인스턴스에 연결을 따라이 작업을 수행할 수 있습니다. 연결할 때는 개발 머신에서 인스턴스로 파일을 전송할 수 있는 방식으로 연결합니다.

> **Note**
>
> Linux 또는 macOS 터미널에서 보안 복사 명령을 사용하여 애플리케이션을 복사할 수 있습니다. 키 페어와 `scp` 함께 `scp -i path/to/key file/to/copy ec2-user@ec2-xx-xx-xxx-xxx.compute.amazonaws.com:~`를 사용하려면 명령을 사용할 수 있습니다. Windows에 대한 자세한 내용은 Windows 인스턴스로 파일 전송을 참조하세요.

AWS 도구 키트를 사용하는 경우 도구 키트를 사용하여 인스턴스에 연결할 수도 있습니다. 자세한 내용은 사용하는 툴킷의 특정한 사용 설명서를 참조하십시오.

EC2 인스턴스에서 애플리케이션 실행

1. 로컬 드라이브에서 Amazon EC2 인스턴스로 애플리케이션 파일을 복사합니다.
2. 애플리케이션을 시작하고 개발 시스템에서 동일한 결과로 실행되는지 확인합니다.
3. (선택 사항) 애플리케이션이 IAM 역할에서 제공하는 보안 인증을 사용하는지 확인합니다.

a. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/ec2/ Amazon EC2 콘솔을 엽니다.

b. 인스턴스를 선택합니다.

c. 작업, 보안을 선택한 다음 IAM 역할 수정을 선택합니다.

d. IAM 역할의 경우 IAM 역할 없음을 선택하여 IAM 역할을 분리합니다.

e. IAM 역할 업데이트를 선택합니다.

f. 애플리케이션을 다시 실행하고 인증 오류가 반환되는지 확인합니다.

TIP 플러그인을 사용하여 액세스 AWS 서비스

신뢰할 수 있는 자격 증명 전파(TIP)는 관리자가 그룹 연결과 같은 사용자 속성을 기반으로 권한을 부여 AWS 서비스 할 수 AWS IAM Identity Center 있는의 기능입니다. 신뢰할 수 있는 자격 증명 전파를 통해 자격 증명 컨텍스트가 IAM 역할에 추가되어 AWS 리소스에 대한 액세스를 요청하는 사용자를 식별합니다. 이 컨텍스트는 다른에 전파됩니다 AWS 서비스.

자격 증명 컨텍스트는가 액세스 요청을 수신할 때 권한 부여 결정을 내리는 데 AWS 서비스 사용하는 정보로 구성됩니다. 이 정보에는 요청자(예: IAM Identity Center 사용자), 액세스가 요청 AWS 서비스 되는 (예: Amazon Redshift) 및 액세스 범위(예: 읽기 전용 액세스)를 식별하는 메타데이터가 포함됩니다. 수신는이 컨텍스트와 사용자에게 할당된 모든 권한을 AWS 서비스 사용하여 리소스에 대한 액세스를 승인합니다. 자세한 내용은 AWS IAM Identity Center 사용 설명서의 신뢰할 수 있는 자격 증명 전파 개요의 섹션을 참조하세요.

TIP 플러그인은 신뢰할 수 AWS 서비스 있는 자격 증명 전파를 지원하는와 함께 사용할 수 있습니다. 참조 사용 사례로 Amazon Q Business 사용 설명서의를 사용하여 Amazon Q Business 애플리케이션 구성을 AWS IAM Identity Center 참조하세요.

> **Note**
>
> Amazon Q Business를 사용하는 경우 서비스별 지침은 를 사용하여 Amazon Q Business 애플리케이션 구성을 AWS IAM Identity Center 참조하세요.

TIP 플러그인을 사용하기 위한 사전 조건

플러그인이 작동하려면 다음 리소스가 필요합니다.

1. AWS SDK for Java 또는를 사용해야 합니다 AWS SDK for JavaScript.
2. 사용 중인 서비스가 신뢰할 수 있는 자격 증명 전파를 지원하는지 확인합니다.

   AWS IAM Identity Center 사용 설명서의 IAM Identity Center와 통합되는 관리형 애플리케이션의 IAM Identity Center를 통해 신뢰할 수 있는 ID 전파 활성화 열을 참조하세요. AWS
3. IAM Identity Center 및 신뢰할 수 있는 자격 증명 전파를 활성화합니다.

   AWS IAM Identity Center 사용 설명서의 TIP 사전 조건 및 고려 사항을 참조하세요.
4. Identity-Center-integrated 애플리케이션이 있어야 합니다.

   AWS IAM Identity Center 사용 설명서의 AWS 관리형 애플리케이션 또는 고객 관리형 애플리케이션을 참조하세요.
5. 신뢰할 수 있는 토큰 발급자(TTI)를 설정하고 서비스를 IAM Identity Center에 연결해야 합니다.

   AWS IAM Identity Center 사용 설명서의 신뢰할 수 있는 토큰 발급자에 대한 사전 조건 및 신뢰할 수 있는 토큰 발급자 설정을 위한 작업을 참조하세요.

코드에서 TIP 플러그인을 사용하려면

1. 신뢰할 수 있는 자격 증명 전파 플러그인의 인스턴스를 생성합니다.
2. 와 상호 작용하기 위한 서비스 클라이언트 인스턴스를 생성하고 신뢰할 수 있는 자격 증명 전파 플러그인을 추가하여 서비스 클라이언트를 AWS 서비스 사용자 지정합니다.

TIP 플러그인은 다음 입력 파라미터를 사용합니다.

- **`webTokenProvider`**: 고객이 외부 자격 증명 공급자로부터 OpenID 토큰을 얻기 위해 구현하는 함수입니다.
- **`accessRoleArn`**: 자격 증명 강화 자격 증명을 가져오기 위해 사용자의 자격 증명 컨텍스트가 있는 플러그인에서 수임할 IAM 역할 ARN입니다.
- **`applicationArn`**: 클라이언트 또는 애플리케이션의 고유 식별자 문자열입니다. 이 값은 OAuth 권한 부여가 구성된 애플리케이션 ARN입니다.
- **`ssoOidcClient`**: (선택 사항) 고객 정의 구성을 사용하는 `SsoOidcClient` for Java 또는 `client-sso-oidc` for JavaScript와 같은 SSO OIDC 클라이언트입니다. 제공하지 않으면를 사용하는 OIDC 클라이언트`applicationRoleArn`가 인스턴스화되고 사용됩니다.

- **stsClient**: (선택 사항) AWS STS 사용자의 자격 증명 컨텍스트로 수임하는 데 사용되는 고객 정의 구성accessRoleArn의 클라이언트입니다. 제공되지 않으면를 사용하는 AWS STS 클라이언트applicationRoleArn가 인스턴스화되고 사용됩니다.
- **applicationRoleArn**: (선택 사항) OIDC 및 AWS STS 클라이언트를 부트스트래핑할 수 AssumeRoleWithWebIdentity 있도록 로 수임할 IAM 역할 ARN입니다.
  - 제공되지 않은 경우 ssoOidcClient 및 stsClient 파라미터를 모두 제공해야 합니다.
  - 제공된 경우는 accessRoleArn 파라미터와 동일한 값이 될 applicationRoleArn 수 없습니다. applicationRoleArn는 accessRole을 수임하는 데 사용되는 stsClient를 빌드하는 데 사용됩니다. applicationRole 및 모두에 동일한 역할을 사용하는 경우 역할을 사용하여 자신을 수임(자체 역할 가정)하는 accessRole것을 의미하므로 권장되지 않습니다 AWS. 자세한 내용은 공지 사항을 참조하세요.

ssoOidcClient, stsClient및 applicationRoleArn 파라미터에 대한 고려 사항

TIP 플러그인을 구성할 때 제공하는 파라미터에 따라 다음 권한 요구 사항을 고려하세요.

- ssoOidcClient 및를 제공하는 경우stsClient:
  - 의 자격 증명에는 자격 증명 센터를 호출하여 자격 증명 센터별 사용자 컨텍스트를 가져올 수 있는 oauth:CreateTokenWithIAM 권한이 ssoOidcClient 있어야 합니다.
  - 의 자격 증명에는 sts:AssumeRole,에 대한 sts:SetContext 권한이 stsClient 있어야 합니다accessRole. accessRole 또한의 자격 증명과 신뢰 관계로를 구성해야 합니다stsClient.
- 를 제공하는 경우applicationRoleArn:
  - applicationRole 에는 OIDC oauth:CreateTokenWithIAM sts:AssumeRole 및 STS 클라이언트를 빌드하는 데 사용되므로 필요한 리소스(IdC 인스턴스, accessRole)에 대한 및 sts:SetContext 권한이 있어야 합니다.IdC
  - applicationRole는 플러그인의 AssumeRoleWithWebIdentity 호출을 통해 applicationRole을 수임하는 데 사용되므로 webTokenwebToken는를 생성하는 데 사용되는 자격 증명 공급자와 신뢰 관계가 있어야 합니다.

ApplicationRole 구성의 예:

웹 토큰 공급자를 사용한 신뢰 정책:

```
{
    "Version": "2012-10-17",
    "Statement": [
```

```
        {
            "Effect": "Allow",
            "Principal": {
                "Federated": "arn:aws:iam::ACCOUNT_ID:oidc-provider/
IDENTITY_PROVIDER_URL"
            },
            "Action": "sts:AssumeRoleWithWebIdentity",
            "Condition": {
                "StringEquals": {
                    "IDENTITY_PROVIDER_URL:aud": "CLIENT_ID_TO_BE_TRUSTED"
                }
            }
        }
    ]
}
```

권한 정책:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sts:AssumeRole",
                "sts:SetContext"
            ],
            "Resource": [
                "accessRoleArn"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "sso-oauth:CreateTokenWithIAM"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

TIP를 사용한 코드 예제

아래 예제에서는 AWS SDK for Java 또는를 사용하여 코드에 TIP 플러그인을 구현하는 방법을 보여줍니다 AWS SDK for JavaScript.

Java

AWS SDK for Java 프로젝트에서 TIP 플러그인을 사용하려면 프로젝트 `pom.xml` 파일의 종속성으로 선언해야 합니다.

```
<dependency>
<groupId>software.amazon.awsidentity.trustedIdentityPropagation</groupId>
<artifactId>aws-sdk-java-trustedIdentityPropagation-java-plugin</artifactId>
   <version>2.0.0</version>
</dependency>
```

소스 코드에에 필요한 패키지 문을 포함합니다`software.amazon.awssdk.trustedidentitypropagation`.

다음 예제에서는 신뢰할 수 있는 자격 증명 전파 플러그인의 인스턴스를 생성하고 이를 서비스 클라이언트에 추가하는 두 가지 방법을 보여줍니다. 두 예제 모두 Amazon S3를 서비스로 사용하고 `S3AccessGrantsPlugin`를 사용하여 사용자별 권한을 관리하지만 신뢰할 수 AWS 서비스 있는 자격 증명 전파(TIP)를 지원하는 모든에 적용할 수 있습니다.

> **Note**
>
> 이 예제에서는 S3 Access Grants에서 사용자별 권한을 설정해야 합니다. 자세한 내용은 S3 Access Grants 설명서를 참조하세요.

옵션 1: OIDC 및 STS 클라이언트 빌드 및 전달

```
SsoOidcClient oidcClient = SsoOidcClient.builder()
    .region(Region.US_EAST_1)
    .credentialsProvider(credentialsProvider).build();

StsClient stsClient = StsClient.builder()
    .region(Region.US_EAST_1)
    .credentialsProvider(credentialsProvider).build();
```

```
TrustedIdentityPropagationPlugin trustedIdentityPropagationPlugin =
 TrustedIdentityPropagationPlugin.builder()
        .webTokenProvider(() -> webToken)
        .applicationArn(idcApplicationArn)
        .accessRoleArn(accessRoleArn)
        .ssoOidcClient(oidcClient)
        .stsClient(stsClient)
        .build();

S3AccessGrantsPlugin accessGrantsPlugin = S3AccessGrantsPlugin.builder()
        .build();

S3Client s3Client =
        S3Client.builder().region(Region.US_EAST_1)
                .crossRegionAccessEnabled(true)
                .addPlugin(trustedIdentityPropagationPlugin)
                .addPlugin(accessGrantsPlugin)
                .build();

final var resp = s3Client.getObject(GetObjectRequest.builder()
        .key("path/to/object/fileName")
        .bucket("bucketName")
        .build());
```

옵션 2: applicationRoleArn 전달 및 플러그인으로 클라이언트 생성 연기

```
TrustedIdentityPropagationPlugin trustedIdentityPropagationPlugin =
 TrustedIdentityPropagationPlugin.builder()
        .webTokenProvider(() -> webToken)
        .applicationArn(idcApplicationArn)
        .accessRoleArn(accessRoleArn)
        .applicationRoleArn(applicationRoleArn)
        .build();

S3AccessGrantsPlugin accessGrantsPlugin = S3AccessGrantsPlugin.builder()
        .build();

S3Client s3Client =
        S3Client.builder().region(Region.US_EAST_1)
                .crossRegionAccessEnabled(true)
                .addPlugin(trustedIdentityPropagationPlugin)
                .addPlugin(accessGrantsPlugin)
                .build();
```

```
final var resp = s3Client.getObject(GetObjectRequest.builder()
        .key("path/to/object/fileName")
        .bucket("bucketName")
        .build());
```

추가 세부 정보 및 소스는 GitHub의 trusted-identity-propagation-java를 참조하세요.

JavaScript

다음 명령을 실행하여 AWS SDK for JavaScript 프로젝트에 TIP 인증 플러그인 패키지를 설치합니다.

```
$  npm i @aws-sdk-extension/trusted-identity-propagation
```

최종 에는 다음과 유사한 종속성이 포함되어야 `package.json` 합니다.

```
  "dependencies": {
"@aws-sdk-extension/trusted-identity-propagation": "^2.0.0"
  },
```

소스 코드에서 필요한 `TrustedIdentityPropagationExtension` 종속성을 가져옵니다.

다음 예제에서는 신뢰할 수 있는 자격 증명 전파 플러그인의 인스턴스를 생성하고 이를 서비스 클라이언트에 추가하는 두 가지 방법을 보여줍니다. 두 예제 모두 Amazon S3를 서비스로 사용하고 Amazon S3 Access Grants를 사용하여 사용자별 권한을 관리하지만 신뢰할 수 AWS 서비스 있는 자격 증명 전파(TIP)를 지원하는 모든에 적용할 수 있습니다.

> **Note**
>
> 이 예제에서는 Amazon S3 Access Grants에서 사용자별 권한을 설정해야 합니다. 자세한 내용은 Amazon S3 Access Grants 설명서를 참조하세요.

옵션 1: OIDC 및 STS 클라이언트 빌드 및 전달

```
import { S3Client, GetObjectCommand } from "@aws-sdk/client-s3";
import { S3ControlClient, GetDataAccessCommand } from "@aws-sdk/client-s3-control";
import { TrustedIdentityPropagationExtension } from "@aws-sdk-extension/trusted-
identity-propagation";

const s3ControlClient = new S3ControlClient({
```

```
    region: "us-east-1",
    extensions: [
        TrustedIdentityPropagationExtension.create({
            webTokenProvider: async () => {
                return 'ID_TOKEN_FROM_YOUR_IDENTITY_PROVIDER';
            },
            ssoOidcClient: customOidcClient,
            stsClient: customStsClient,
            accessRoleArn: accessRoleArn,
            applicationArn: applicationArn,
        }),
    ],
});

const getDataAccessParams = {
  Target: "S3_URI_PATH",
  Permission: "READ",
  AccountId: ACCOUNT_ID,
  InstanceArn: S3_ACCESS_GRANTS_ARN,
  TargetType: "Object",
};

try {
  const command = new GetDataAccessCommand(getDataAccessParams);
  const response = await s3ControlClient.send(command);

  const credentials = response.Credentials;

  // Create a new S3 client with the temporary credentials
  const temporaryS3Client = new S3Client({
    region: "us-east-1",
    credentials: {
      accessKeyId: credentials.AccessKeyId,
      secretAccessKey: credentials.SecretAccessKey,
      sessionToken: credentials.SessionToken,
    },
  });

  // Use the temporary S3 client to perform the operation
  const s3Params = {
    Bucket: "BUCKET_NAME",
    Key: "S3_OBJECT_KEY",
  };
  const getObjectCommand = new GetObjectCommand(s3Params);
```

```
  const s3Object = await temporaryS3Client.send(getObjectCommand);

  const fileContent = await s3Object.Body.transformToString();

  // Process the S3 object data
  console.log("Successfully retrieved S3 object:", fileContent);
} catch (error) {
  console.error("Error accessing S3 data:", error);
}
```

옵션 2: applicationRoleArn 전달 및 플러그인으로 클라이언트 생성 연기

```
import { S3Client, GetObjectCommand } from "@aws-sdk/client-s3";
import { S3ControlClient, GetDataAccessCommand } from "@aws-sdk/client-s3-control";
import { TrustedIdentityPropagationExtension } from "@aws-sdk-extension/trusted-
identity-propagation";

const s3ControlClient = new S3ControlClient({
    region: "us-east-1",
    extensions: [
        TrustedIdentityPropagationExtension.create({
            webTokenProvider: async () => {
                return 'ID_TOKEN_FROM_YOUR_IDENTITY_PROVIDER';
            },
            accessRoleArn: accessRoleArn,
            applicationRoleArn: applicationRoleArn,
            applicationArn: applicationArn,
        }),
    ],
});

// Same S3 AccessGrants workflow as Option 1
const getDataAccessParams = {
  Target: "S3_URI_PATH",
  Permission: "READ",
  AccountId: ACCOUNT_ID,
  InstanceArn: S3_ACCESS_GRANTS_ARN,
  TargetType: "Object",
};

try {
  const command = new GetDataAccessCommand(getDataAccessParams);
  const response = await s3ControlClient.send(command);
```

```
  const credentials = response.Credentials;

  const temporaryS3Client = new S3Client({
    region: "us-east-1",
    credentials: {
      accessKeyId: credentials.AccessKeyId,
      secretAccessKey: credentials.SecretAccessKey,
      sessionToken: credentials.SessionToken,
    },
  });

  const s3Params = {
    Bucket: "BUCKET_NAME",
    Key: "S3_OBJECT_KEY",
  };
  const getObjectCommand = new GetObjectCommand(s3Params);
  const s3Object = await temporaryS3Client.send(getObjectCommand);

  const fileContent = await s3Object.Body.transformToString();

  console.log("Successfully retrieved S3 object:", fileContent);
} catch (error) {
  console.error("Error accessing S3 data:", error);
}
```

추가 세부 정보 및 소스는 GitHub의 trusted-identity-propagation-js를 참조하세요.

AWS SDKs 및 도구 설정 참조

SDKs 언어별 APIs 제공합니다 AWS 서비스. 이 SDK는 인증, 재시도 동작 등을 포함하여 API 직접 호출을 성공적으로 수행하는 데 필요한 일부 번거로운 작업을 처리합니다. 이를 위해 SDK에는 요청에 사용할 보안 인증을 얻고, 각 서비스에 사용할 설정을 유지 관리하고, 글로벌 설정에 사용할 값을 가져오는 유연한 전략이 있습니다.

구성 설정에 대한 자세한 내용은 다음 단원에서 확인할 수 있습니다.

- AWS SDKs 및 도구 표준화 자격 증명 공급자 - 여러 SDK에 표준화된 공통 보안 인증 공급자.
- AWS SDKs 및 도구 표준화된 기능 - 여러 SDK에 표준화된 공통 기능.

서비스 클라이언트 생성

프로그래밍 방식으로에 액세스하기 위해 AWS 서비스 SDKs 각각에 대해 클라이언트 클래스/객체를 사용합니다 AWS 서비스. 예를 들어, 애플리케이션이 Amazon EC2에 액세스해야 하는 경우, 애플리케이션은 Amazon EC2 클라이언트 객체를 생성하여 해당 서비스와 인터페이스 합니다. 그런 다음 서비스 클라이언트를 사용하여 요청을 AWS 서비스에 보내면 됩니다. 대부분의 SDKs에서 서비스 클라이언트 객체는 변경할 수 없으므로 요청을 하는 각 서비스와 다른 구성을 사용하여 동일한 서비스에 요청하는 각 서비스에 대해 새 클라이언트를 생성해야 합니다.

설정의 우선 순위

글로벌 설정은 대부분의 SDK가 지원하는 기능, 보안 인증 공급자 및 기타 기능을 구성하며 AWS 서비스전반에 광범위하게 영향을 미칩니다. 모든 SDK에는 글로벌 설정 값을 찾기 위해 확인하는 일련의 위치 (또는 소스)가 있습니다. 조회 우선 순위 설정은 다음과 같습니다.

1. 코드나 서비스 클라이언트 자체에 설정된 모든 명시적 설정은 다른 모든 설정보다 우선합니다.
   - 일부 설정은 작업별로 설정할 수 있으며 간접 호출하는 각 작업에 대해 필요에 따라 변경할 수 있습니다. AWS CLI 또는의 경우 명령줄에 입력하는 작업당 파라미터의 형태를 AWS Tools for PowerShell취합니다. SDK의 경우 명시적 할당은 AWS 서비스 클라이언트 또는 구성 객체를 인스턴스화할 때 또는 경우에 따라 개별 API를 호출할 때 설정한 파라미터의 형태를 취할 수 있습니다.
2. Java/Kotlin만 해당: 설정에 대한 JVM 시스템 속성이 확인됩니다. 설정된 경우 해당 값은 클라이언트를 구성하는 데 사용됩니다.
3. 환경 변수를 확인합니다. 설정된 경우 해당 값은 클라이언트를 구성하는 데 사용됩니다.

4. SDK는 공유 `credentials` 파일에서 설정을 확인합니다. 설정된 경우 클라이언트가 이를 사용합니다.
5. 설정에 대한 공유 `config` 파일입니다. 설정이 있으면 SDK는 해당 설정을 사용합니다.
   - `AWS_PROFILE` 환경 변수 또는 `aws.profile` JVM 시스템 속성을 사용하여 SDK가 로드하는 프로필을 지정할 수 있습니다.
6. SDK 소스 코드 자체에서 제공하는 모든 기본값은 마지막으로 사용됩니다.

> **Note**
>
> 일부 SDK 및 도구는 순서가 다를 수 있습니다. 또한 일부 SDK 및 도구는 다른 파라미터 저장과 검색 방법을 지원합니다. 예를 들어는 SDK 스토어라는 추가 소스를 AWS SDK for .NET 지원합니다. 특정 SDK 또는 도구 제공자에 대한 자세한 내용은 사용 중인 특정 SDK 또는 도구의 설명서를 참조합니다.

순서에 따라 어떤 방법이 다른 메서드보다 우선 적용되는지 결정됩니다. 예를 들어 공유 `config` 파일에 프로파일을 설정하면 SDK 또는 도구가 먼저 다른 위치를 확인한 후에 해당 프로파일을 찾아 사용합니다. 즉, `credentials` 파일에 설정을 입력하면 `config` 파일에 있는 설정이 아닌 해당 설정이 사용됩니다. 설정과 값으로 환경 변수를 구성하면 `credentials` 및 `config` 파일 모두의 해당 설정을 재정의 합니다. 마지막으로 개별 작업 (AWS CLI 명령 라인 매개 변수 또는 API 매개 변수)또는 코드의 설정이 해당 명령에 대한 다른 모든 값보다 우선합니다.

이 가이드의 설정 페이지 이해

이 가이드의 설정 참조 섹션에 있는 페이지에는 다양한 메커니즘을 통해 설정할 수 있는 사용 가능한 설정이 자세히 설명되어 있습니다. 다음 표에는 구성 및 자격 증명 파일 설정, 환경 변수, (Java 및 Kotlin SDKs 경우) 코드 외부에서 기능을 구성하는 데 사용할 수 있는 JVM 설정이 나와 있습니다. 각 목록의 연결된 각 주제는 해당 설정 페이지로 이동합니다.

- `Config` 파일 설정 목록
- `Credentials` 파일 설정 목록
- 환경 변수 목록
- JVM 시스템 속성 목록

각 자격 증명 공급자 또는 기능에는 해당 기능을 구성하는 데 사용되는 설정이 나열되는 페이지가 있습니다. 각 설정에 대해 구성 파일에 설정을 추가하거나 환경 변수를 설정하거나(Java 및 Kotlin만 해당) JVM 시스템 속성을 설정하여 값을 설정할 수 있습니다. 각 설정은 설명의 세부 정보 위에 있는 블록에서 값을 설정하는 지원되는 모든 방법을 나열합니다. 우선 순위는 다르지만 설정 방법에 관계없이 결과 기능은 동일합니다.

설명에는 아무 작업도 하지 않을 경우 적용되는 기본값이 포함됩니다. 또한 해당 설정에 유효한 값이 무엇인지 정의합니다.

예를 들어 기능 요청 압축 페이지에서 설정을 살펴보겠습니다.

`disable_request_compression` 예제 설정의 정보는 다음을 문서화합니다.

- 코드베이스 외부에서 요청 압축을 제어하는 방법에는 세 가지가 있습니다. 다음 작업 중 하나를 수행할 수 있습니다.
  - 를 사용하여 구성 파일에서 설정 `disable_request_compression`
  - 를 사용하여 환경 변수로 설정 `AWS_DISABLE_REQUEST_COMPRESSION`
  - 또는 Java 또는 Kotlin SDK를 사용하는 경우를 사용하여 JVM 시스템 속성으로 설정합니다. `aws.disableRequestCompression`

  > **Note**
  >
  > 코드에서 직접 동일한 기능을 구성하는 방법도 있을 수 있지만,이 참조는 각 SDK에 고유하므로 이를 다루지 않습니다. 코드 자체에서 구성을 설정하려면 특정 SDK 가이드 또는 API 참조를 참조하세요.

- 아무 작업도 수행하지 않으면 값은 기본적으로 로 설정됩니다`false`.
- 이 부울 설정에 유효한 유일한 값은 `true` 및 입니다`false`.

각 기능 페이지 하단에는 Support AWS SDKs.

이 표는 SDK가 페이지에 나열된 설정을 지원하는지 여부를 보여줍니다. `Supported` 열은 다음 값을 사용하여 지원 수준을 나타냅니다.

- `Yes` - 설정은 작성된 대로 SDK에서 완전히 지원됩니다.
- `Partial` - 일부 설정이 지원되거나 동작이 설명과 다릅니다. 의 경우 추가 참고 `Partial`사항은 편차를 나타냅니다.

- No - 어떤 설정도 지원되지 않습니다. 이는 코드에서 동일한 기능을 달성할 수 있는지 여부에 대한 클레임을 하지 않으며 나열된 외부 구성 설정이 지원되지 않음을 나타냅니다.

Config 파일 설정 목록

다음 표에 나열된 설정은 공유 AWS config 파일에서 할당할 수 있습니다. 이는 전 세계에 적용되며 모든 AWS 서비스에 영향을 미칩니다. SDKs 및 도구는 고유한 설정 및 환경 변수를 지원할 수도 있습니다. 개별 SDK 또는 도구에서만 지원하는 설정 및 환경 변수를 보려면 해당 특정 SDK 또는 도구 안내서를 참조하세요.

설정 이름	세부 정보	
account_id_endpoint_mode	계정 기반 엔드포인트	
api_versions	일반 구성 설정	
auth_scheme_preference	인증 체계	
aws_access_key_id	AWS 액세스 키	
aws_account_id	계정 기반 엔드포인트	
aws_secret_access_key	AWS 액세스 키	
aws_session_token	AWS 액세스 키	
ca_bundle	일반 구성 설정	
credential_process	프로세스 보안 인증 제공자	

설정 이름	세부 정보
credential_source	역할 보안 인증 제공자 수임
defaults_mode	스마트 구성 기본값
disable_host_prefix_injection	호스트 접두사 삽입
disable_request_compression	요청 압축
duration_seconds	역할 보안 인증 제공자 수임
ec2_metadata_service_endpoint	IMDS 보안 인증 제공자
ec2_metadata_service_endpoint_mode	IMDS 보안 인증 제공자
ec2_metadata_v1_disabled	IMDS 보안 인증 제공자
endpoint_discovery_enabled	엔드포인트 검색
endpoint_url	서비스별 엔드포인트
external_id	역할 보안 인증 제공자 수임

설정 이름	세부 정보	
ignore_configured_endpoint_urls	서비스별 엔드포인트	
max_attempts	재시도 동작	
metadata_service_num_attempts	Amazon EC2 인스턴스 메타데이터	
metadata_service_timeout	Amazon EC2 인스턴스 메타데이터	
mfa_serial	역할 보안 인증 제공자 수임	
output	일반 구성 설정	
parameter_validation	일반 구성 설정	
region	AWS 리전	
request_checksum_calculation	Amazon S3에 대한 데이터 무결성 보호	
request_min_compression_size_bytes	요청 압축	
response_checksum_validation	Amazon S3에 대한 데이터 무결성 보호	
retry_mode	재시도 동작	

설정 이름	세부 정보	
role_arn	역할 보안 인증 제공자 수임	
role_session_name	역할 보안 인증 제공자 수임	
s3_disable_express_session_auth	S3 Express One Zone 세션 인증	
s3_disable_multiregion_access_points	Amazon S3 다중 리전 액세스 포인트	
s3_use_arn_region	Amazon S3 액세스 포인트	
sdk_ua_app_id	애플리케이션 ID	
sigv4a_signing_region_set	인증 체계	
source_profile	역할 보안 인증 제공자 수임	
sso_account_id	IAM 아이덴티티 센터 보안 인증 공급자	
sso_region	IAM 아이덴티티 센터 보안 인증 공급자	
sso_registration_scopes	IAM 아이덴티티 센터 보안 인증 공급자	
sso_role_name	IAM 아이덴티티 센터 보안 인증 공급자	
sso_start_url	IAM 아이덴티티 센터 보안 인증 공급자	
sts_regional_endpoints	AWS STS 리전 엔드포인트	

설정 이름	세부 정보	
use_dualstack_endpoint	이중 스택 엔드포인트 및 FIPS 엔드포인트	
use_fips_endpoint	이중 스택 엔드포인트 및 FIPS 엔드포인트	
web_identity_token_file	역할 보안 인증 제공자 수임	

Credentials 파일 설정 목록

다음 표에 나열된 설정은 공유 AWS credentials 파일에서 할당할 수 있습니다. 이는 전 세계에 적용되며 모든 AWS 서비스에 영향을 미칩니다. SDKs 및 도구는 고유한 설정 및 환경 변수를 지원할 수도 있습니다. 개별 SDK 또는 도구에서만 지원하는 설정 및 환경 변수를 보려면 해당 특정 SDK 또는 도구 안내서를 참조하세요.

설정 이름	세부 정보	
aws_access_key_id	AWS 액세스 키	
aws_secret_access_key	AWS 액세스 키	
aws_session_token	AWS 액세스 키	

환경 변수 목록

대부분의 SDK에서 지원되는 환경 변수가 아래에 나열되어 있습니다. 이는 전 세계에 적용되며 모든 AWS 서비스에 영향을 미칩니다. SDKs 및 도구는 고유한 설정 및 환경 변수를 지원할 수도 있습니다. 개별 SDK 또는 도구에서만 지원하는 설정 및 환경 변수를 보려면 해당 특정 SDK 또는 도구 안내서를 참조하세요.

설정 이름	세부 정보
AWS_ACCESS_KEY_ID	AWS 액세스 키
AWS_ACCOUNT_ID	계정 기반 엔드포인트
AWS_ACCOUNT_ID_ENDPOINT_MODE	계정 기반 엔드포인트
AWS_AUTH_SCHEME_PREFERENCE	인증 체계
AWS_CA_BUNDLE	일반 구성 설정
AWS_CONFIG_FILE	AWS SDKs 및 도구의 공유 config 및 credentials 파일 위치 찾기 및 변경
AWS_CONTAINER_AUTHORIZATION_TOKEN	컨테이너 보안 인증 제공업체
AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE	컨테이너 보안 인증 제공업체
AWS_CONTAINER_CREDENTIALS_FULL_URI	컨테이너 보안 인증 제공업체
AWS_CONTAINER_CRED	컨테이너 보안 인증 제공업체

설정 이름	세부 정보	
ENTIALS_RELATIVE_URI		
AWS_DEFAULTS_MODE	스마트 구성 기본값	
AWS_DISABLE_HOST_PREFIX_INJECTION	호스트 접두사 삽입	
AWS_DISABLE_REQUEST_COMPRESSION	요청 압축	
AWS_EC2_METADATA_DISABLED	IMDS 보안 인증 제공자	
AWS_EC2_METADATA_SERVICE_ENDPOINT	IMDS 보안 인증 제공자	
AWS_EC2_METADATA_SERVICE_ENDPOINT_MODE	IMDS 보안 인증 제공자	
AWS_EC2_METADATA_V1_DISABLED	IMDS 보안 인증 제공자	
AWS_ENABLE_ENDPOINT_DISCOVERY	엔드포인트 검색	

설정 이름	세부 정보
AWS_ENDPOINT_URL	서비스별 엔드포인트
AWS_ENDPOINT_URL_<SERVICE>	서비스별 엔드포인트
AWS_IGNORE_CONFIGURED_ENDPOINT_URLS	서비스별 엔드포인트
AWS_MAX_ATTEMPTS	재시도 동작
AWS_METADATA_SERVICE_NUM_ATTEMPTS	Amazon EC2 인스턴스 메타데이터
AWS_METADATA_SERVICE_TIMEOUT	Amazon EC2 인스턴스 메타데이터
AWS_PROFILE	공유 config 및 credentials 파일을 사용하여 AWS SDKs 및 도구 전역 구성
AWS_REGION	AWS 리전
AWS_REQUEST_CHECKSUM_CALCULATION	Amazon S3에 대한 데이터 무결성 보호
AWS_REQUEST_MIN_COMPRESSION_SIZE_BYTES	요청 압축

설정 이름	세부 정보	
AWS_RESPONSE_CHECKSUM_VALIDATION	Amazon S3에 대한 데이터 무결성 보호	
AWS_RETRY_MODE	재시도 동작	
AWS_ROLE_ARN	역할 보안 인증 제공자 수임	
AWS_ROLE_SESSION_NAME	역할 보안 인증 제공자 수임	
AWS_S3_DISABLE_EXPRESS_SESSION_AUTH	S3 Express One Zone 세션 인증	
AWS_S3_DISABLE_MULTIREGION_ACCESS_POINTS	Amazon S3 다중 리전 액세스 포인트	
AWS_S3_USE_ARN_REGION	Amazon S3 액세스 포인트	
AWS_SDK_UA_APP_ID	애플리케이션 ID	
AWS_SECRET_ACCESS_KEY	AWS 액세스 키	
AWS_SESSION_TOKEN	AWS 액세스 키	
AWS_SHARED_CREDENTIALS_FILE	AWS SDKs 및 도구의 공유 config 및 credentials 파일 위치 찾기 및 변경	

설정 이름	세부 정보	
AWS_SIGV4A_SIGNING_REGION_SET	인증 체계	
AWS_STS_REGIONAL_ENDPOINTS	AWS STS 리전 엔드포인트	
AWS_USE_DUALSTACK_ENDPOINT	이중 스택 엔드포인트 및 FIPS 엔드포인트	
AWS_USE_FIPS_ENDPOINT	이중 스택 엔드포인트 및 FIPS 엔드포인트	
AWS_WEB_IDENTITY_TOKEN_FILE	역할 보안 인증 제공자 수임	

JVM 시스템 속성 목록

AWS SDK for Java 및 AWS SDK for Kotlin (JVM 대상 지정)에 대해 다음 JVM 시스템 속성을 사용할 수 있습니다. JVM 시스템 속성을 설정하는 방법에 대한 지침은 the section called “JVM 시스템 속성을 설정하는 방법” 섹션을 참조하세요.

설정 이름	세부 정보	
aws.accessKeyId	AWS 액세스 키	
aws.accountId	계정 기반 엔드포인트	
aws.accountIdEndpointMode	계정 기반 엔드포인트	

설정 이름	세부 정보	
aws.authSchemePreference	인증 체계	
aws.configFile	AWS SDKs 및 도구의 공유 config 및 credentials 파일 위치 찾기 및 변경	
aws.defaultsMode	스마트 구성 기본값	
aws.disableEc2MetadataV1	IMDS 보안 인증 제공자	
aws.disableHostPrefixInjection	호스트 접두사 삽입	
aws.disableRequestCompression	요청 압축	
aws.disableS3ExpressAuth	S3 Express One Zone 세션 인증	
aws.ec2MetadataServiceEndpoint	IMDS 보안 인증 제공자	
aws.ec2MetadataServiceEndpointMode	IMDS 보안 인증 제공자	

설정 이름	세부 정보
aws.endpointDiscoveryEnabled	엔드포인트 검색
aws.endpointUrl	서비스별 엔드포인트
aws.endpointUrl<ServiceName>	서비스별 엔드포인트
aws.ignoreConfiguredEndpointUrls	서비스별 엔드포인트
aws.maxAttempts	재시도 동작
aws.profile	공유 config 및 credentials 파일을 사용하여 AWS SDKs 및 도구 전역 구성
aws.region	AWS 리전
aws.requestChecksumCalculation	Amazon S3에 대한 데이터 무결성 보호
aws.requestMinCompressionSizeBytes	요청 압축
aws.responseChecksumValidation	Amazon S3에 대한 데이터 무결성 보호
aws.retryMode	재시도 동작
aws.roleArn	역할 보안 인증 제공자 수임

설정 이름	세부 정보
aws.roleSessionName	역할 보안 인증 제공자 수임
aws.s3DisableMultiRegionAccessPoints	Amazon S3 다중 리전 액세스 포인트
aws.s3UseArnRegion	Amazon S3 액세스 포인트
aws.secretAccessKey	AWS 액세스 키
aws.sessionToken	AWS 액세스 키
aws.sharedCredentialsFile	AWS SDKs 및 도구의 공유 config 및 credentials 파일 위치 찾기 및 변경
aws.useDualstackEndpoint	이중 스택 엔드포인트 및 FIPS 엔드포인트
aws.useFipsEndpoint	이중 스택 엔드포인트 및 FIPS 엔드포인트
aws.webIdentityTokenFile	역할 보안 인증 제공자 수임
sdk.ua.appId	애플리케이션 ID

AWS SDKs 및 도구 표준화 자격 증명 공급자

많은 보안 인증 공급자가 일관된 기본값을 유지하고 여러 SDK에서 동일한 방식으로 작동하도록 표준화되었습니다. 이러한 일관성은 여러 SDK에서 코딩할 때 생산성과 명확성을 높입니다. 코딩으로 모든 설정을 재정의할 수 있습니다. 자세한 내용은 특정 SDK API를 참조하십시오.

> **Important**
>
> 모든 SDK가 모든 제공자를 지원하는 것은 아니며 공급자 내의 모든 측면을 지원하는 것은 아닙니다.

주제

자격 증명 공급자 체인 이해

모든 SDK에는 AWS 서비스에 요청을 하는 데 사용할 유효한 보안 인증을 찾기 위해 확인하는 일련의 위치 (또는 소스)가 있습니다. 유효한 보안 인증 정보를 찾은 후에는 검색이 중지됩니다. 이러한 체계적인 검색을 자격 증명 공급자 체인이라고 합니다.

표준화된 자격 증명 공급자 중 하나를 사용하는 경우 AWS SDKs 만료 시 항상 자격 증명을 자동으로 갱신하려고 시도합니다. 기본 제공 자격 증명 공급자 체인은 체인에서 사용 중인 공급자에 관계없이 자격 증명을 새로 고칠 수 있는 기능을 애플리케이션에 제공합니다. 이를 위해 SDK에 추가 코드가 필요하지 않습니다.

각 SDK에서 사용하는 고유한 체인은 다르지만 대부분 다음과 같은 소스를 포함합니다.

보안 인증 제공업체	설명
AWS 액세스 키	AWS IAM 사용자의 액세스 키(예: `AWS_ACCESS_KEY_ID` , 및 `AWS_SECRET_ACCESS_KEY`).
웹 자격 증명 또는 OpenID Connect와 페더레이션- 역할 보안 인증 제공자 수임	Login with Amazon, Facebook, Google 또는 다른 OpenID Connect(OIDC)호환 IdP와 같은 널리 알려진 외부 보안 인증 공급자(IdP)를 사용해 로그인합니다. AWS Security Token Service ()의 JSON 웹 토큰(JWT)을 사용하여 IAM 역할의 권한을 수임합니다AWS STS.
IAM 아이덴티티 센터 보안 인증 공급자	에서 자격 증명을 가져옵니다 AWS IAM Identity Center.
역할 보안 인증 제공자 수임	IAM 역할의 권한을 수임하여 다른 리소스에 액세스할 수 있습니다. (역할에 대한 임시 보안 인증을 검색하여 사용).
컨테이너 보안 인증 제공업체	Amazon Elastic Container Service(Amazon ECS) 및 Amazon Elastic Kubernetes Service(Amazon EKS) 보안 인증. 컨테이너 자격 증명 공급자는 고객의 컨테이너화된 애플리케이션에 대한 자격 증명을 가져옵니다.
프로세스 보안 인증 제공자	사용자 정의 보안 인증 공급자. IAM Roles Anywhere를 비롯한 외부 소스 또는 프로세스에서 보안 인증을 가져옵니다.
IMDS 보안 인증 제공자	Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스 프로파일 보안 인증. IAM 역할을 사용자의 각 EC2 인스턴스에 연결합니다. 그러면 인스턴스에서 실행되는 코드에 대해 해당 역할의 임시 보안 인증을 사용할 수 있게 됩니다. 보안 인증은 Amazon EC2 메타데이터 서비스를 통해 전달됩니다.

체인의 각 단계마다 설정 값을 할당하는 여러 방법이 있습니다. 코드에 지정된 설정값이 항상 우선합니다. 그러나 환경 변수과 공유 `config` 및 `credentials` 파일을 사용하여 AWS SDKs 및 도구 전역 구성같은 경우도 있습니다. 자세한 내용은 설정의 우선 순위 단원을 참조하십시오.

SDK별 및 도구별 자격 증명 공급자 체인

SDK 또는 도구의 특정 자격 증명 공급자 체인 세부 정보로 바로 이동하려면 다음에서 SDK 또는 도구를 선택합니다.

- AWS CLI
- SDK for C++
- SDK for Go
- SDK for Java
- SDK for JavaScript
- SDK for Kotlin
- SDK for .NET
- SDK for PHP
- SDK for Python (Boto3)
- SDK for Ruby
- SDK for Rust
- SDK for Swift
- PowerShell용 도구

AWS 액세스 키

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

AWS IAM 사용자의 액세스 키를 자격 AWS 증명으로 사용할 수 있습니다. AWS SDK는 이러한 AWS 자격 증명을 자동으로 사용하여 API 요청에 서명 AWS하므로 워크로드가 AWS 리소스와 데이터에 안전하고 편리하게 액세스할 수 있습니다. 보안 인증이 일시적 유효하지 않거나 만료 후에는 더 이상 유효하지 않도록 항상 `aws_session_token`을 사용하는 것이 좋습니다. 장기 자격 증명 사용은 권장되지 않습니다.

> **Note**
>
> AWS 가 이러한 임시 자격 증명을 새로 고칠 AWS 수 없는 경우 워크로드가 영향을 받지 않도록 자격 증명의 유효성을 확장할 수 있습니다.

공유 AWS `credentials` 파일은 애플리케이션 소스 디렉터리 외부에 안전하게 있고 공유 `config` 파일의 SDK별 설정과 분리되어 있으므로 자격 증명 정보를 저장하는 데 권장되는 위치입니다.

자격 AWS 증명 및 액세스 키 사용에 대한 자세한 내용은 IAM 사용 설명서의 AWS 보안 자격 증명 및 IAM 사용자의 액세스 키 관리를 참조하세요. https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_access-keys.html

다음을 사용하여 이 기능을 구성하십시오.

`aws_access_key_id` - 공유 AWS **`config`** 파일 설정, **`aws_access_key_id`** - 공유 AWS **`credentials`** 파일 설정(권장 방법), **`AWS_ACCESS_KEY_ID`** - 환경 변수, **`aws.accessKeyId`** - JVM 시스템 속성: Java/Kotlin만 해당

사용자를 인증하기 위한 자격 증명의 일부로 사용되는 AWS 액세스 키를 지정합니다.

`aws_secret_access_key` - 공유 AWS **`config`** 파일 설정, **`aws_secret_access_key`** - 공유 AWS **`credentials`** 파일 설정(권장 방법), **`AWS_SECRET_ACCESS_KEY`** - 환경 변수, **`aws.secretAccessKey`** - JVM 시스템 속성: Java/Kotlin만 해당

사용자를 인증하기 위한 자격 증명의 일부로 사용되는 AWS 보안 키를 지정합니다.

`aws_session_token` - 공유 AWS **`config`** 파일 설정, **`aws_session_token`** - 공유 AWS **`credentials`** 파일 설정(권장 방법), **`AWS_SESSION_TOKEN`** - 환경 변수, **`aws.sessionToken`** - JVM 시스템 속성: Java/Kotlin만 해당

사용자를 인증하기 위한 자격 증명의 일부로 사용되는 AWS 세션 토큰을 지정합니다. 역할 수임 요청이 성공하면 반환되는 임시 보안 인증 정보의 일부로 이 값을 받습니다. 세션 토큰은 수동으로 임시 보안 보안 인증을 지정하는 경우에만 필요합니다. 하지만 장기 보안 인증 정보를 사용하는 대신 항상 임시 보안 보안 인증을 사용하는 것이 좋습니다. 보안 권장 사항은 IAM의 보안 모범 사례를 참조하십시오.

이러한 값을 구하는 방법에 대한 자세한 내용은 단기 자격 증명을 사용하여 AWS SDKs 및 도구 인증 단원을 참조하십시오.

`config`또는 `credentials` 파일에 이러한 필수 값을 설정하는 예:

```
[default]
aws_access_key_id = AKIAIOSFODNN7EXAMPLE
aws_secret_access_key = wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
aws_session_token = AQoEXAMPLEH4aoAH0gNCAPy...truncated...zrkuWJOgQs8IZZaIv2BXIa2R4Olgk
```

명령 행을 통한 환경 변수 설정의 Linux/macOS 예제:

```
export AWS_ACCESS_KEY_ID=AKIAIOSFODNN7EXAMPLE
export AWS_SECRET_ACCESS_KEY=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
export
 AWS_SESSION_TOKEN=AQoEXAMPLEH4aoAH0gNCAPy...truncated...zrkuWJOgQs8IZZaIv2BXIa2R4Olgk
```

명령줄을 통해 환경 변수를 설정하는 Windows 예제:

```
setx AWS_ACCESS_KEY_ID AKIAIOSFODNN7EXAMPLE
setx AWS_SECRET_ACCESS_KEY wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
setx
 AWS_SESSION_TOKEN AQoEXAMPLEH4aoAH0gNCAPy...truncated...zrkuWJOgQs8IZZaIv2BXIa2R4Olgk
```

AWS SDKs 도구 지원

다음 SDK는 이 주제에서 설명하는 기능 및 설정을 지원합니다. 모든 일부 예외가 기록됩니다. 모든 JVM 시스템 속성 설정은 AWS SDK for Java 및 AWS SDK for Kotlin 에서만 지원됩니다.

SDK	지원	참고 또는 추가 정보
AWS CLI v2	예	
SDK for C++	예	공유 `config` 파일은 지원되지 않습니다.
SDK for Go V2 (1.x)	예	
SDK for Go 1.x (V1)	예	공유 `config` 파일 설정을 사용하려면 구성 파일에서 로드를 켜야 합니다. 세션을 참조하십시오.
SDK for Java 2.x	예	
SDK for Java 1.x	예	

SDK	지원	참고 또는 추가 정보
SDK for JavaScript 3.x	예	
SDK for JavaScript 2.x	예	
SDK for Kotlin	예	
.NET 4.x용 SDK	예	
SDK for .NET 3.x	예	
SDK for PHP 3.x	예	
SDK for Python (Boto3)	예	
SDK for Ruby 3.x	예	
SDK for Rust	예	
SDK for Swift	예	
PowerShell V5용 도구	예	
PowerShell V4용 도구	예	환경 변수는 지원되지 않습니다.

역할 보안 인증 제공자 수임

> **Note**
>
> 설정 페이지의 레이아웃을 이해하거나 다음에 나오는 Support AWS SDKs이 가이드의 설정 페이지 이해.

역할 수임에는 액세스 권한이 없을 수 있는 AWS 리소스에 액세스하기 위해 일련의 임시 보안 보안 인증을 사용하는 것이 포함됩니다. 이러한 임시 보안 인증은 액세스 키 ID, 보안 액세스 키 및 보안 토큰으로 구성됩니다.

역할을 수임하도록 SDK 또는 도구를 설정하려면 먼저 수임할 특정 역할을 만들거나 식별해야 합니다. IAM 역할은 Amazon 리소스 이름(ARN)역할로 고유하게 식별됩니다. 역할은 다른 엔티티와 신뢰 관계를 구축합니다. 역할을 사용하는 신뢰할 수 있는 엔터티는 AWS 서비스, 다른 AWS 계정, 웹 자격 증명 공급자 또는 OIDC 또는 SAML 페더레이션일 수 있습니다.

IAM 역할을 식별한 후 해당 역할을 신뢰할 수 있는 경우 해당 역할에서 부여한 권한을 사용하도록 SDK 또는 도구를 구성할 수 있습니다. 이렇게 하려면 다음 설정을 사용하십시오.

이러한 설정 사용을 시작하는 방법에 대한 지침은 이 안내서의 자격 AWS 증명이 있는 역할을 수임하여 AWS SDKs 및 도구 인증를 참조하세요.

역할 보안 인증 제공자 수임 설정

다음을 사용하여 이 기능을 구성하십시오.

`credential_source` - 공유 AWS **`config`** 파일 설정

SDK나 도구가 `role_arn` 파라미터로 지정된 역할을 수임하기 위한 권한을 가진 보안 인증을 인증을 찾을 수 있는 위치를 지정하기 위해 Amazon EC2 인스턴스 또는 Amazon Elastic Container Service 컨테이너 내에서 사용됩니다.

기본값: 없음

유효값:

- 환경 – SDK나 도구가 `AWS_ACCESS_KEY_ID` 및 `AWS_SECRET_ACCESS_KEY` 환경 변수에서 소스 보안 인증을 검색하도록 지정합니다.
- Ec2InstanceMetadata – SDK 또는 도구가 EC2 인스턴스 프로파일에 연결된 IAM 역할을 사용하여 소스 보안 인증을 가져오도록 지정합니다.
- EcsContainer - SDK 또는 도구가 Amazon ECS 컨테이너에 연결된 IAM 역할 또는 Amazon EKS 컨테이너에 연결된 IAM 역할을 사용하여 소스 자격 증명을 가져오도록 지정합니다.

`credential_source`과 `source_profile` 모두를 동일한 프로파일에서 지정할 수 없습니다.

보안 인증을 Amazon EC2에서 소싱해야 함을 나타내도록 `config` 파일에 설정하는 예:

```
credential_source = Ec2InstanceMetadata
role_arn = arn:aws:iam::123456789012:role/my-role-name
```

`duration_seconds` - 공유 AWS **`config`** 파일 설정

역할 세션의 최대 기간(초)을 지정합니다.

이 설정은 프로파일에서 역할 수임을 지정한 경우에만 적용됩니다.

기본값: 3600초(1시간)

유효한 값: 이 값의 범위는 900초(15분)부터 해당 역할에 대한 구성된 최대 세션 기간 설정(최대값: 43200초 또는 12시간)까지 가능합니다. 자세한 내용은 IAM 사용 설명서의 역할의 최대 세션 기간 설정 보기를 참조하십시오.

config 파일에서 이를 설정하는 예:

```
duration_seconds = 43200
```

external_id - 공유 AWS **config** 파일 설정

타사에서 고객 계정의 역할을 수임하는 데 사용하는 고유한 식별자를 지정합니다.

이 설정은 프로파일에서 역할을 수임하도록 지정하고 역할에 대한 신뢰 정책에서 ExternalId에 대한 값을 필요로 하는 경우에만 적용됩니다. 값은 프로파일이 역할을 지정할 때 AssumeRole 작업에 전달되는 ExternalId 파라미터에 매핑됩니다.

기본값: 없음.

유효한 값: IAM 사용 설명서의 AWS 리소스에 대한 액세스 권한을 타사에 부여할 때 외부 ID를 사용하는 방법을 참조하세요.

config 파일에서 이를 설정하는 예:

```
external_id = unique_value_assigned_by_3rd_party
```

mfa_serial - 공유 AWS **config** 파일 설정

사용자가 역할 수임 시 사용해야 하는 다중 인증(MFA)장치의 ID 또는 일련 번호를 지정합니다.

해당 역할에 대한 신뢰 정책에 MFA 인증을 필요로 하는 조건이 포함된 역할을 수임하는 경우 필요합니다. MFA에 대한 자세한 내용은 AWS IAM 사용 설명서의 IAM의 멀티 팩터 인증을 참조하세요.

기본값: 없음.

유효한 값: 이 값은 하드웨어 디바이스용 일련 번호(예: GAHT12345678)또는 가상 MFA 디바이스용 Amazon 리소스 이름(ARN)(예:)일 수 있습니다. ARN의 형식은 다음과 같습니다. arn:aws:iam::*account-id*:mfa/*mfa-device-name*

config 파일에서 이를 설정하는 예:

이 예제에서는 계정에 대해 생성MyMFADevice되고 사용자에 대해 활성화된 라는 가상 MFA 디바이스를 가정합니다.

```
mfa_serial = arn:aws:iam::123456789012:mfa/MyMFADevice
```

role_arn - 공유 AWS **config** 파일 설정, **AWS_ROLE_ARN** - 환경 변수, **aws.roleArn** - JVM 시스템 속성: Java/Kotlin만 해당

이 프로파일을 사용하여 요청된 작업을 수행하는 데 사용할 IAM 역할의 Amazon 리소스 이름(ARN)을 지정합니다.

기본값: 없음.

유효한 값: 이 값은 다음과 같은 형식의 IAM 역할의 ARN이어야 합니다. `arn:aws:iam::account-id:role/role-name`

또한 다음 설정 중 하나를 지정해야 합니다.

- `source_profile` — 이 프로파일에서 역할을 수임할 권한이 있는 보안 인증을 찾는 데 사용할 다른 프로파일을 식별합니다.
- `credential_source` — 현재 환경 변수로 식별되는 보안 인증 또는 Amazon EC2 인스턴스 프로파일 또는 Amazon ECS 컨테이너 인스턴스에 첨부된 보안 인증을 사용합니다.
- `web_identity_token_file` — 모바일 또는 웹 애플리케이션에서 인증된 사용자에 대해 퍼블릭 ID 공급자 또는 OpenID Connect(OIDC)호환 보안 인증 공급자를 사용합니다.

role_session_name - 공유 AWS **config** 파일 설정, **AWS_ROLE_SESSION_NAME** - 환경 변수, **aws.roleSessionName** - JVM 시스템 속성: Java/Kotlin만 해당

역할 세션에 연결할 이름을 지정합니다. 이 이름은 이 세션과 연결된 항목에 대한 AWS CloudTrail 로그에 나타나며, 감사할 때 유용할 수 있습니다. 자세한 내용은 AWS CloudTrail 사용 설명서의 CloudTrail userIdentity 요소를 참조하세요.

기본값: 선택적 파라미터입니다. 이 값을 제공하지 않은 경우 프로파일이 역할을 수임할 때 세션 이름이 자동으로 생성됩니다.

유효한 값: AWS CLI 또는 AWS API가 사용자를 대신하여 `AssumeRole` 작업(또는 작업과 같은 `AssumeRoleWithWebIdentity` 작업)을 호출할 때 `RoleSessionName` 파라미터에 제공됩니다. 값은 쿼리할 수 있는 수임된 역할 사용자 Amazon 리소스 이름(ARN)의 일부가 되며, 이 프로파일에서 간접 호출한 작업에 대한 CloudTrail 로그 항목의 일부로 표시됩니다.

arn:aws:sts::*123456789012*:assumed-role/*my-role-name*/***my-role_session_name***.

config 파일에서 이를 설정하는 예:

```
role_session_name = my-role-session-name
```

source_profile - 공유 AWS **config** 파일 설정

보안 인증이 원래 프로파일의 role_arn 설정에 지정된 역할을 수임하는 데 사용되는 다른 프로파일을 지정합니다. 공유 AWS config 및 credentials 파일에서 프로필이 사용되는 방법을 이해하려면 섹션을 참조하세요공유 config 및 credentials 파일.

역할 수임 프로파일이기도 한 프로파일을 지정하는 경우 보안 인증을 완전히 확인하기 위해 각 역할이 순차적으로 수임됩니다. SDK가 보안 인증이 있는 프로파일을 발견하면 이 체인이 중지됩니다. 역할 체인은 AWS CLI 또는 AWS API 역할 세션을 최대 1시간으로 제한하며 늘릴 수 없습니다. 자세한 내용은 IAM 사용 설명서의 역할 용어 및 개념을 참조하십시오.

기본값: 없음.

유효한 값: config 및 credentials 파일에 정의된 프로파일 이름으로 구성된 텍스트 문자열입니다. 또한 현재 프로파일에서 role_arn의 값도 지정해야 합니다.

credential_source과 source_profile 모두를 동일한 프로파일에서 지정할 수 없습니다.

구성 파일에서 이를 설정하는 예:

```
[profile A]
source_profile = B
role_arn =  arn:aws:iam::123456789012:role/RoleA
role_session_name = ProfileARoleSession

[profile B]
credential_process = ./aws_signing_helper credential-process --certificate /
path/to/certificate --private-key /path/to/private-key --trust-anchor-
arn arn:aws:rolesanywhere:region:account:trust-anchor/TA_ID --profile-
arn arn:aws:rolesanywhere:region:account:profile/PROFILE_ID --role-arn
 arn:aws:iam::account:role/ROLE_ID
```

이전 예제에서 A 프로필은 연결된 B 프로필의 자격 증명을 자동으로 조회하도록 SDK 또는 도구에 지시합니다. 이 경우 B 프로필은에서 제공하는 자격 증명 도우미 도구를 사용하여 SDK의 AWS 자격 증명을 IAM Roles Anywhere를 사용하여 AWS SDKs 및 도구 인증 가져옵니다. 이러한 임시 보

안 인증은 코드에서 AWS 리소스에 액세스하기 위해 사용됩니다. 지정된 역할에는 명령 AWS 서비스또는 API 메서드와 같이 요청된 코드를 실행할 수 있도록 허용하는 IAM 권한 정책이 연결되어 있어야 합니다. 프로필에서 수행하는 모든 작업A에는 CloudTrail 로그에 역할 세션 이름이 포함됩니다.

역할 체인의 두 번째 예에서는 Amazon Elastic Compute Cloud 인스턴스에 애플리케이션이 있고 해당 애플리케이션이 다른 역할을 수임하도록 하려는 경우 다음 구성을 사용할 수 있습니다.

```
[profile A]
source_profile = B
role_arn =  arn:aws:iam::123456789012:role/RoleA
role_session_name = ProfileARoleSession

[profile B]
credential_source=Ec2InstanceMetadata
```

프로필A은 Amazon EC2 인스턴스의 자격 증명을 사용하여 지정된 역할을 수임하고 자격 증명을 자동으로 갱신합니다.

`web_identity_token_file` - 공유 AWS **`config`** 파일 설정, **`AWS_WEB_IDENTITY_TOKEN_FILE`** - 환경 변수, **`aws.webIdentityTokenFile`** - JVM 시스템 속성: Java/Kotlin만 해당

지원되는 OAuth 2.0 공급자 또는 OpenID Connect ID 공급자의 액세스 토큰을 포함하는 파일의 경로를 지정합니다.

이 설정을 사용하면 Google, Facebook 및 Amazon 등과 같은 웹 ID 페더레이션 공급자를 사용하여 인증할 수 있습니다. SDK 또는 개발자 도구는 이 파일의 내용을 로드하고 사용자를 대신하여 `AssumeRoleWithWebIdentity` 작업을 호출할 때 `WebIdentityToken` 인수로서 전달합니다.

기본값: 없음.

유효한 값: 이 값은 경로 및 파일 이름이어야 합니다. 파일에는 ID 공급자가 제공한 OAuth 2.0 액세스 토큰 또는 OpenID Connect ID 토큰을 포함해야 합니다. 상대 경로는 프로세스의 작업 디렉터리를 기준으로 처리됩니다.

AWS SDKs 도구 지원

다음 SDK는 이 주제에서 설명하는 기능 및 설정을 지원합니다. 모든 일부 예외가 기록됩니다. 모든 JVM 시스템 속성 설정은 AWS SDK for Java 및 AWS SDK for Kotlin 에서만 지원됩니다.

SDK	지원	참고 또는 추가 정보
AWS CLI v2	예	
SDK for C++	부분적	`credential_source` 이 지원되지 않음. `duration_seconds` 이 지원되지 않음. `mfa_serial` 이 지원되지 않음.
SDK for Go V2 (1.x)	예	
SDK for Go 1.x (V1)	예	공유 `config` 파일 설정을 사용하려면 구성 파일에서 로드를 켜야 합니다. 세션을 참조하십시오.
SDK for Java 2.x	부분적	`mfa_serial` 지원되지 않습니다. 지원되지 `duration_seconds` 않습니다.
SDK for Java 1.x	부분적	`credential_source` 지원되지 않음. 지원되지 `mfa_serial` 않음. JVM 시스템 속성은 지원되지 않습니다.
SDK for JavaScript 3.x	예	
SDK for JavaScript 2.x	부분적	`credential_source` 이 지원되지 않음.
SDK for Kotlin	예	
.NET 4.x용 SDK	예	
SDK for .NET 3.x	예	
SDK for PHP 3.x	예	
SDK for Python (Boto3)	예	
SDK for Ruby 3.x	예	

SDK	지원	참고 또는 추가 정보
SDK for Rust	예	
SDK for Swift	예	
PowerShell V5용 도구	예	
PowerShell V4용 도구	예	

컨테이너 보안 인증 제공업체

> **Note**
>
> 설정 페이지의 레이아웃을 이해하거나 다음에 나오는 Support AWS SDKs이 가이드의 설정 페이지 이해.

컨테이너 자격 증명 공급자는 고객의 컨테이너화된 애플리케이션에 대한 자격 증명을 가져옵니다. 이 보안 인증 공급자는 Amazon Elastic Container Service(Amazon ECS) 및 Amazon Elastic Kubernetes Service(Amazon EKS) 고객에게 유용합니다. SDK는 GET 요청을 통해 지정된 HTTP 엔드포인트에서 보안 인증을 로드하려고 시도합니다.

Amazon ECS를 사용하는 경우 보안 인증 격리, 권한 부여 및 감사 가능성을 개선하기 위해 작업 IAM 역할을 사용하는 것이 좋습니다. 구성된 경우 Amazon ECS는 SDK 및 도구가 보안 인증을 얻기 위해 사용하는 `AWS_CONTAINER_CREDENTIALS_RELATIVE_URI` 환경 변수를 설정합니다. 이 기능을 사용하도록 Amazon ECS를 구성하려면 Amazon Elastic Container Service 개발자 안내서의 태스크 IAM 역할을 참조하세요.

Amazon EKS를 사용하는 경우 보안 인증 격리, 최소 권한, 감사 가능성, 독립적인 운영, 재사용성 및 확장성을 개선하기 위해 Amazon EKS Pod Identity를 사용하는 것이 좋습니다. 포드와 IAM 역할은 모두 Kubernetes 서비스 계정과 연결되어 애플리케이션에 대한 보안 인증을 관리합니다. Amazon EKS 포드 ID에 대해 자세히 알아보려면 Amazon EKS 사용 설명서의 Amazon EKS Pod Identities를 참조하세요. 구성된 경우 Amazon EKS는 SDK 및 도구가 보안 인증을 얻기 위해 사용하는 `AWS_CONTAINER_CREDENTIALS_FULL_URI` 및 `AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE` 환경 변수를 설정합니다. 설정 정보는 Amazon EKS 사용 설명서의 Amazon EKS Pod Identity Agent

설정을 참조하거나 AWS 블로그 웹 사이트의 Amazon EKS Pod Identity simplifies IAM permissions for applications on Amazon EKS clusters를 참조하세요.

다음을 사용하여 이 기능을 구성하십시오.

`AWS_CONTAINER_CREDENTIALS_FULL_URI` - 환경 변수

보안 인증을 요청할 때 SDK가 사용할 전체 HTTP URL 엔드포인트를 지정합니다. 여기에는 스키마와 호스트가 모두 포함됩니다.

기본값: 없음.

유효한 값: 유효한 URI.

참고: 이 설정은 *`AWS_CONTAINER_CREDENTIALS_RELATIVE_URI`*설정의 대안이며 *`AWS_CONTAINER_CREDENTIALS_RELATIVE_URI`*이설정되지 않은 경우에만 사용됩니다.

명령 행을 통한 환경 변수 설정의 Linux/macOS 예제:

```
export AWS_CONTAINER_CREDENTIALS_FULL_URI=http://localhost/get-credentials
```

or

```
export AWS_CONTAINER_CREDENTIALS_FULL_URI=http://localhost:8080/get-credentials
```

`AWS_CONTAINER_CREDENTIALS_RELATIVE_URI` - 환경 변수

보안 인증을 요청할 때 SDK가 사용할 상대 HTTP URL 엔드포인트를 지정합니다. 값은 기본 Amazon ECS 호스트 이름인 `169.254.170.2`에 추가됩니다.

기본값: 없음.

유효한 값: 유효한 상대 URI.

명령 행을 통한 환경 변수 설정의 Linux/macOS 예제:

```
export AWS_CONTAINER_CREDENTIALS_RELATIVE_URI=/get-credentials?a=1
```

`AWS_CONTAINER_AUTHORIZATION_TOKEN` - 환경 변수

인증 토큰을 일반 텍스트로 지정합니다. 이 변수를 설정하면 SDK는 환경 변수 값을 사용하여 HTTP 요청의 권한 헤더를 설정합니다.

기본값: 없음.

유효한 값: 문자열.

참고: 이 설정은 *AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE*설정의 대안이며 *AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE*이설정되지 않은 경우에만 사용됩니다.

명령 행을 통한 환경 변수 설정의 Linux/macOS 예제:

```
export AWS_CONTAINER_CREDENTIALS_FULL_URI=http://localhost/get-credential
export AWS_CONTAINER_AUTHORIZATION_TOKEN=Basic abcd
```

AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE - 환경 변수

일반 텍스트로 된 인증 토큰을 포함하는 파일의 절대 파일 경로를 지정합니다.

기본값: 없음.

유효한 값: 문자열.

명령 행을 통한 환경 변수 설정의 Linux/macOS 예제:

```
export AWS_CONTAINER_CREDENTIALS_FULL_URI=http://localhost/get-credential
export AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE=/path/to/token
```

AWS SDKs 도구 지원

다음 SDK는 이 주제에서 설명하는 기능 및 설정을 지원합니다. 모든 일부 예외가 기록됩니다. 모든 JVM 시스템 속성 설정은 AWS SDK for Java 및 AWS SDK for Kotlin 에서만 지원됩니다.

SDK	지원	참고 또는 추가 정보
AWS CLI v2	예	
SDK for C++	예	
SDK for Go V2 (1.x)	예	
SDK for Go 1.x (V1)	예	

SDK	지원	참고 또는 추가 정보
SDK for Java 2.x	예	Lambda SnapStart가 활성화AWS_CONTAINER_CREDENTIALS_FULL_URI AWS_CONTAINER_AUTHORIZATION_TOKEN 되고 인증에 자동으로 사용됩니다.
SDK for Java 1.x	예	Lambda SnapStart가 활성화AWS_CONTAINER_CREDENTIALS_FULL_URI AWS_CONTAINER_AUTHORIZATION_TOKEN 되고 인증에 자동으로 사용됩니다.
SDK for JavaScript 3.x	예	
SDK for JavaScript 2.x	예	
SDK for Kotlin	예	
.NET 4.x용 SDK	예	Lambda SnapStart가 활성화AWS_CONTAINER_CREDENTIALS_FULL_URI AWS_CONTAINER_AUTHORIZATION_TOKEN 되고 인증에 자동으로 사용됩니다.
SDK for .NET 3.x	예	Lambda SnapStart가 활성화AWS_CONTAINER_CREDENTIALS_FULL_URI AWS_CONTAINER_AUTHORIZATION_TOKEN 되고 인증에 자동으로 사용됩니다.
SDK for PHP 3.x	예	
SDK for Python (Boto3)	예	Lambda SnapStart가 활성화AWS_CONTAINER_CREDENTIALS_FULL_URI AWS_CONTAINER_AUTHORIZATION_TOKEN 되고 인증에 자동으로 사용됩니다.
SDK for Ruby 3.x	예	
SDK for Rust	예	
SDK for Swift	예	
PowerShell V5용 도구	예	
PowerShell V4용 도구	예	

IAM 아이덴티티 센터 보안 인증 공급자

> **Note**
>
> 설정 페이지의 레이아웃을 이해하거나 다음에 나오는 Support AWS SDKs이 가이드의 설정 페이지 이해.

이 인증 메커니즘은 AWS IAM Identity Center 를 사용하여 코드 AWS 서비스 에 대한 Single Sign-On(SSO) 액세스 권한을 얻습니다.

> **Note**
>
> AWS SDK API 설명서에서 IAM Identity Center 자격 증명 공급자를 SSO 자격 증명 공급자라고 합니다.

IAM Identity Center를 활성화한 후 공유 AWS `config` 파일에서 설정에 대한 프로파일을 정의합니다. 이 프로파일은 IAM ID 센터 액세스 포털에 연결하는 데 사용됩니다. 사용자가 IAM Identity Center에서 성공적으로 인증하면 포털은 해당 사용자와 관련된 IAM 역할에 대한 단기 보안 인증을 반환합니다. SDK가 구성에서 임시 자격 증명을 가져와 AWS 서비스 요청에 사용하는 방법을 알아보려면 섹션을 참조하세요AWS SDKs 및 도구에 대한 IAM Identity Center 인증 확인 방법.

`config` 파일을 통해 IAM ID 센터를 구성하는 두 가지 방법이 있습니다.

- (권장) SSO 토큰 공급자 구성 - 세션 기간 연장. 사용자 지정 세션 기간에 대한 지원이 포함됩니다.
- 새로 고칠 수 없는 레거시 구성 - 고정된 8시간 세션을 사용합니다.

두 구성 모두 세션이 만료되면 다시 로그인해야 합니다.

다음 두 안내서에는 IAM Identity Center에 대한 추가 정보가 포함되어 있습니다.

- AWS IAM Identity Center 사용 설명서
- AWS IAM Identity Center 포털 API 참조

SDK 및 도구가 이 구성을 사용하여 보안 인증을 사용하고 새로 고치는 방법에 대한 자세한 내용은 AWS SDKs 및 도구에 대한 IAM Identity Center 인증 확인 방법 섹션을 참조하십시오.

사전 조건

먼저 IAM Identity Center를 활성화해야 합니다. IAM Identity Center 인증 활성화에 대한 자세한 내용은 AWS IAM Identity Center 사용 설명서의 활성화 AWS IAM Identity Center를 참조하세요.

> **Note**
>
> 또는이 페이지에 자세히 설명된 전체 사전 조건 및 필요한 공유 `config` 파일 구성은 설정에 대한 안내 지침을 참조하세요IAM Identity Center를 사용하여 AWS SDK 및 도구 인증.

SSO 토큰 공급자 구성

SSO 토큰 공급자 구성을 사용하면 AWS SDK 또는 도구가 연장된 세션 기간까지 세션을 자동으로 새로 고칩니다. 세션 기간 및 최대 기간에 대한 자세한 내용은 AWS IAM Identity Center 사용 설명서의 AWS 액세스 포털 및 IAM Identity Center 통합 애플리케이션의 세션 기간 구성을 참조하세요.

`config` 파일의 `sso-session` 섹션은 SSO 액세스 토큰을 획득하기 위한 구성 변수를 그룹화하는 데 사용되며, 그런 다음 자격 AWS 증명을 획득하는 데 사용할 수 있습니다. `config` 파일 내이 섹션에 대한 자세한 내용은 섹션을 참조하세요구성 파일 형식.

다음 공유 `config` 파일 예제에서는 `dev` 프로필을 사용하여 SDK 또는 도구를 구성하여 IAM Identity Center 자격 증명을 요청합니다.

```
[profile dev]
sso_session = my-sso
sso_account_id = 111122223333
sso_role_name = SampleRole

[sso-session my-sso]
sso_region = us-east-1
sso_start_url = https://my-sso-portal.awsapps.com/start
sso_registration_scopes = sso:account:access
```

이전 예제에서는 `sso-session` 섹션을 정의하고 프로필에 연결했음을 보여줍니다. 일반적으로 SDK가 AWS 자격 증명을 요청할 수 있도록 `profile` 섹션에서 `sso_account_id` 및를 `sso_role_name` 설정해야 합니다. `sso-session` 섹션 내에서 `sso_start_url`, 및 `sso_region`를 설정해야 `sso_registration_scopes` 합니다.

`sso_account_id` 및 `sso_role_name`은 SSO 토큰 구성의 모든 시나리오에 필수적이지는 않습니다. 애플리케이션에서 보유자 인증을 지원하는 AWS 서비스 만 사용하는 경우 기존 AWS 자격 증명이

필요하지 않습니다. 보유자 인증은 보유자 토큰이라는 보안 토큰을 사용하는 HTTP 인증 체계입니다. 이 시나리오에서는 sso_account_id 및 sso_role_name은 필수가 아닙니다. 서비스가 보유자 토큰 권한 부여를 지원하는지 확인하려면 개별 AWS 서비스 가이드를 참조하세요.

등록 범위는 sso-session의 일부로 구성됩니다. 범위는 애플리케이션의 사용자 계정 액세스를 제한하는 OAuth 2.0의 메커니즘입니다. 이전 예제에서는 계정 및 역할을 나열sso_registration_scopes하는 데 필요한 액세스를 제공하도록를 설정합니다.

다음 예제에서는 여러 프로파일에서 동일한 sso-session 구성을 재사용하는 방법을 보여줍니다.

```
[profile dev]
sso_session = my-sso
sso_account_id = 111122223333
sso_role_name = SampleRole

[profile prod]
sso_session = my-sso
sso_account_id = 111122223333
sso_role_name = SampleRole2

[sso-session my-sso]
sso_region = us-east-1
sso_start_url = https://my-sso-portal.awsapps.com/start
sso_registration_scopes = sso:account:access
```

인증 토큰은 세션 이름을 기반으로 하는 파일 이름을 사용하여 ~/.aws/sso/cache 디렉터리 아래의 디스크에 캐시됩니다.

새로 고칠 수 없는 레거시 구성

새로 고칠 수 없는 기존 구성을 사용하는 자동 토큰 새로 고침은 지원되지 않습니다. 대신 SSO 토큰 공급자 구성을(를)사용하는 것이 좋습니다.

새로 고칠 수 없는 기존 구성을 사용하려면 프로파일에서 다음 설정을 지정해야 합니다.

- sso_start_url
- sso_region
- sso_account_id
- sso_role_name

`sso_start_url` 및 `sso_region` 설정을 사용하여 프로파일의 사용자 포털을 지정합니다. `sso_account_id` 및 `sso_role_name` 설정으로 권한을 지정합니다.

다음 예제에서는 `config` 파일에 필요한 네 가지 값을 설정합니다.

```
[profile my-sso-profile]
sso_start_url = https://my-sso-portal.awsapps.com/start
sso_region = us-west-2
sso_account_id = 111122223333
sso_role_name = SSOReadOnlyRole
```

인증 토큰은 `sso_start_url`에 기반한 파일 이름을 가진 `~/.aws/sso/cache` 디렉터리 아래의 디스크에 캐시됩니다.

IAM Identity Center 보안 인증 공급자 설정

다음을 사용하여 이 기능을 구성하십시오.

`sso_start_url` - 공유 AWS **`config`** 파일 설정

조직의 IAM Identity Center 발급자 URL 또는 액세스 포털 URL을 가리키는 URL입니다. 자세한 내용은 AWS IAM Identity Center 사용 설명서의 AWS 액세스 포털 사용을 참조하세요.

이 값을 찾으려면 IAM Identity Center 콘솔을 열고 대시보드를 보고 AWS 액세스 포털 URL을 찾습니다.

- 또는 버전 2.22.0부터 AWS 발급자 URL에 값을 사용할 AWS CLI수 있습니다.

`sso_region` - 공유 AWS **`config`** 파일 설정

IAM Identity Center 포털 호스트, 즉 IAM Identity Center를 활성화하기 전에 선택한 리전이 AWS 리전 포함된 . 이는 기본 AWS 리전과 독립적이며 다를 수 있습니다.

AWS 리전 및 해당 코드의 전체 목록은의 리전 엔드포인트를 참조하세요Amazon Web Services 일반 참조. 이 값을 찾으려면 IAM Identity Center 콘솔을 열고 대시보드를 확인한 다음 리전을 찾으십시오.

`sso_account_id` - 공유 AWS **`config`** 파일 설정

인증을 위해 AWS Organizations 서비스를 통해 추가된의 숫자 ID AWS 계정 입니다.

사용 가능한 계정 목록을 보려면 IAM Identity Center 콘솔로 이동하여 AWS 계정 페이지를 여십시오. AWS IAM Identity Center 포털 API 참조의 ListAccounts API 메서드를 사용하여 사용 가능한 계정 목록을 볼 수도 있습니다. 예를 들어 AWS CLI , 메서드 list-accounts를 호출할 수 있습니다.

`sso_role_name` - 공유 AWS **`config`** 파일 설정

사용자의 최종 권한을 정의하는 IAM 역할로 프로비저닝된 권한 집합의 이름입니다. 역할은에서 AWS 계정 지정한에 있어야 합니다`sso_account_id`. Amazon 리소스 이름(ARN)역할을 사용하지 말고 역할 이름을 사용하십시오.

권한 세트에는 IAM 정책 및 사용자 지정 권한 정책이 첨부되어 있으며 할당된 AWS 계정에 대한 사용자의 액세스 수준을 정의합니다.

당 사용 가능한 권한 세트 목록을 보려면 IAM Identity Center 콘솔로 AWS 계정이동하여 AWS 계정 페이지를 엽니다. AWS 계정 테이블에 나열된 올바른 권한 세트 이름을 선택합니다. AWS IAM Identity Center 포털 API 참조의 ListAccountsRoles API 메서드를 사용하여 사용 가능한 권한 집합 목록을 볼 수도 있습니다. 예를 들어 AWS CLI , 메서드 list-account-roles를 호출할 수 있습니다.

`sso_registration_scopes` - 공유 AWS **`config`** 파일 설정

`sso-session`에 대해 인증될 쉼표로 구분된 유효한 범위 문자열 목록입니다. 애플리케이션은 하나 이상의 범위를 요청할 수 있으며 애플리케이션에 발급되는 액세스 토큰은 부여된 범위로 제한됩니다. IAM Identity Center 서비스에서 새로 고침 토큰을 다시 받으려면 `sso:account:access`의 최소 범위를 부여해야 합니다. 사용 가능한 액세스 범위 옵션 목록은 AWS IAM Identity Center 사용 설명서의 액세스 범위를 참조하세요.

이러한 범위는 등록된 OIDC 클라이언트에 대해 인증받기 위해 요청된 권한과 클라이언트가 검색한 액세스 토큰을 정의합니다. 범위는 IAM Identity Center 보유자 토큰 인증 엔드포인트에 대한 액세스를 승인합니다.

이 설정은 새로 고칠 수 없는 레거시 구성에는 적용되지 않습니다. 레거시 구성을 사용하여 발급된 토큰은 암시적으로 범위 `sso:account:access`(으)로 제한됩니다.

AWS SDKs 도구 지원

다음 SDK는 이 주제에서 설명하는 기능 및 설정을 지원합니다. 모든 일부 예외가 기록됩니다. 모든 JVM 시스템 속성 설정은 AWS SDK for Java 및 AWS SDK for Kotlin 에서만 지원됩니다.

SDK	지원	참고 또는 추가 정보
AWS CLI v2	예	
SDK for C++	예	

SDK	지원	참고 또는 추가 정보
SDK for Go V2 (1.x)	예	
SDK for Go 1.x (V1)	예	공유 config 파일 설정을 사용하려면 구성 파일에서 로드를 켜야 합니다. 세션을 참조하십시오.
SDK for Java 2.x	예	구성 값은 credentials 파일에서도 지원됩니다.
SDK for Java 1.x	아니요	
SDK for JavaScript 3.x	예	
SDK for JavaScript 2.x	예	
SDK for Kotlin	예	
.NET 4.x용 SDK	예	
SDK for .NET 3.x	예	
SDK for PHP 3.x	예	
SDK for Python (Boto3)	예	
SDK for Ruby 3.x	예	
SDK for Rust	부분	새로 고칠 수 없는 레거시 구성에만 해당.
SDK for Swift	예	
PowerShell V5용 도구	예	
PowerShell V4용 도구	예	

IMDS 보안 인증 제공자

> **Note**
>
> 설정 페이지의 레이아웃을 이해하거나 다음에 나오는 Support AWS SDKs이 가이드의 설정 페이지 이해.

인스턴스 메타데이터 서비스(IMDS)는 실행 중인 인스턴스를 구성 또는 관리하는 데 사용할 수 있는 인스턴스 관련 데이터를 제공합니다. 사용 가능한 데이터에 대한 자세한 내용은 Amazon EC2 사용 설명서의 인스턴스 메타데이터 작업을 참조하세요. Amazon EC2는 인스턴스에 대한 다양한 정보를 제공할 수 있는 로컬 엔드포인트를 제공합니다. 인스턴스에 역할이 연결된 경우 해당 역할에 유효한 보안 인증 세트를 제공할 수 있습니다. SDK는 해당 엔드포인트를 사용하여 기본 보안 인증 공급자 체인의 일부로 보안 인증을 확인할 수 있습니다. 세션 토큰을 사용하는 보다 안전한 IMDS 버전인 인스턴스 메타데이터 서비스 버전 2(IMDSv2)가 기본으로 사용됩니다. 재시도할 수 없는 조건(HTTP 오류 코드 403, 404, 405)으로 인해 이 기본이 실패할 경우 IMDSv1이 대체 수단으로 사용됩니다.

다음을 사용하여 이 기능을 구성하십시오.

`AWS_EC2_METADATA_DISABLED` - 환경 변수

보안 인증 획득에 Amazon EC2 인스턴스 메타데이터 서비스 (IMDS)사용 시도 여부

기본값: `false`.

유효값:

- **`true`** - 보안 인증을 얻는 데 IMDS를 사용하지 않습니다.
- **`false`** - 보안 인증을 얻는 데 IMDS를 사용합니다.

`ec2_metadata_v1_disabled` - 공유 AWS **`config`** 파일 설정, **`AWS_EC2_METADATA_V1_DISABLED`** - 환경 변수, **`aws.disableEc2MetadataV1`** - JVM 시스템 속성: Java/Kotlin만 해당

IMDSv2가 실패할 경우 IMDSv1(Instance Metadata Service Version 1)을 폴백으로 사용할지 여부.

> **Note**
>
> 새 SDK는 IMDSv1을 지원하지 않으므로 이 설정을 지원하지 않습니다. 자세한 내용은 테이블 AWS SDKs 도구 지원를 참조하세요.

기본값: `false`

유효값:

- **`true`** - IMDSv1을 폴백으로 사용하지 않습니다.
- **`false`** - IMDSv1을 폴백으로 사용합니다.

`ec2_metadata_service_endpoint` - 공유 AWS **`config`** 파일 설정, **`AWS_EC2_METADATA_SERVICE_ENDPOINT`** - 환경 변수, **`aws.ec2MetadataServiceEndpoint`** - JVM 시스템 속성: Java/Kotlin만 해당

IMDS 엔드포인트. 이 값은 AWS SDKs. Amazon EC2

기본값: `ec2_metadata_service_endpoint_mode`와 `IPv4`이 같으면 기본 엔드포인트는 `http://169.254.169.254`입니다. `ec2_metadata_service_endpoint_mode`와 `IPv6`이 같으면 기본 엔드포인트는 `http://[fd00:ec2::254]`입니다.

유효한 값: 유효한 URI.

`ec2_metadata_service_endpoint_mode` - 공유 AWS **`config`** 파일 설정, **`AWS_EC2_METADATA_SERVICE_ENDPOINT_MODE`** - 환경 변수, **`aws.ec2MetadataServiceEndpointMode`** - JVM 시스템 속성: Java/Kotlin만 해당

IMDS의 엔드포인트 모드.

기본값:`IPv4`.

유효한 값: `IPv4`, `IPv6`.

> **Note**
>
> IMDS 보안 인증 제공자는 자격 증명 공급자 체인 이해의 일부입니다. 그러나 IMDS 자격증 제공자는 여기에 있는 일련의 제공자를 거친 후에만 확인됩니다. 따라서 프로그램에서 이 공급자의 보안 인증을 사용하려면 구성에서 다른 유효한 보안 인증 공급자를 제거하거나 다른 프로파일을 사용해야 합니다. 또는 보안 인증 제공자 체인에 의존하여 어떤 제공자가 유효한 보안 인증을 반환하는지 자동으로 검색하는 대신 IMDS 보안 인증 제공자의 사용을 코드로 지정하십시오. 서비스 클라이언트를 생성할 때 보안 인증 소스를 직접 지정할 수 있습니다.

IMDS 보안 인증 보안

기본적으로 AWS SDK가 유효한 자격 증명으로 구성되지 않은 경우 SDK는 Amazon EC2 인스턴스 메타데이터 서비스(IMDS)를 사용하여 AWS 역할의 자격 증명을 검색하려고 시도합니다. `AWS_EC2_METADATA_DISABLED`환경 변수를 `true`로 설정하여 이 동작을 비활성화할 수 있습니다. 이를 통해 Amazon EC2 인스턴스 메타데이터 서비스를 가장 신뢰할 수 없는 네트워크에서 불필요한 네트워크 활동을 방지하고 보안을 강화합니다.

> **Note**
>
> AWS 유효한 자격 증명으로 구성된 SDK 클라이언트는 이러한 설정에 관계없이 IMDS를 사용하여 자격 증명을 검색하지 않습니다.

Amazon EC2 IMDS 보안 인증 비활성화

이 환경 변수를 설정하는 방법은 사용 중인 운영 체제와 변경 내용을 지속적으로 적용할지 여부에 따라 달라집니다.

Linux 및 macOS

Linux 또는 macOS를 사용하는 고객은 다음 명령을 사용해 이 환경 변수를 설정할 수 있습니다.

```
$ export AWS_EC2_METADATA_DISABLED=true
```

여러 쉘 세션 및 시스템 재시작 시에도 이 설정을 유지하려면 위의 명령을 `.bash_profile`, `.zsh_profile`, 혹은 `.profile` 등의 쉘 프로파일 파일에 추가할 수 있습니다.

Windows

Windows를 사용하는 고객은 다음 명령을 사용해 이 환경 변수를 설정할 수 있습니다.

```
$ set AWS_EC2_METADATA_DISABLED=true
```

이 설정을 여러 쉘 세션 및 시스템 재시작에 걸쳐 지속되게 하려면 다음 명령을 대신 사용할 수 있습니다.

```
$ setx AWS_EC2_METADATA_DISABLED=true
```

> **Note**
>
> 이 setx 명령은 현재 쉘 세션에 값을 적용하지 않으므로 변경 내용을 적용하려면 쉘을 다시 로드하거나 다시 열어야 합니다.

AWS SDKs 도구 지원

다음 SDK는 이 주제에서 설명하는 기능 및 설정을 지원합니다. 모든 일부 예외가 기록됩니다. 모든 JVM 시스템 속성 설정은 AWS SDK for Java 및 AWS SDK for Kotlin 에서만 지원됩니다.

SDK	지원	참고 또는 추가 정보
AWS CLI v2	예	
SDK for C++	예	
SDK for Go V2 (1.x)	예	
SDK for Go 1.x (V1)	예	공유 config 파일 설정을 사용하려면 구성 파일에서 로드를 켜야 합니다. 세션을 참조하십시오.
SDK for Java 2.x	예	
SDK for Java 1.x	부분	JVM 시스템 속성: com.amazonaws.sdk.disableEc2MetadataV1 대신 aws.disableEc2MetadataV1 를 사용aws.ec2MetadataServiceEndpoint 하며 지원되지 aws.ec2MetadataServiceEndpointMode 않습니다.
SDK for JavaScript 3.x	예	
SDK for JavaScript 2.x	예	
SDK for Kotlin	예	IMDSv1 폴백을 사용하지 않습니다.
.NET 4.x용 SDK	예	
SDK for .NET 3.x	예	

SDK	지원	참고 또는 추가 정보
SDK for PHP 3.x	예	
SDK for Python (Boto3)	예	
SDK for Ruby 3.x	예	
SDK for Rust	예	IMDSv1 폴백을 사용하지 않습니다.
SDK for Swift	예	
PowerShell V5용 도구	예	를 사용하여 코드에서 IMDSv1 폴백을 명시적으로 비활성화할 수 있습니다`[Amazon.Util.EC2InstanceMetadata]::EC2MetadataV1Disabled = $true` .
PowerShell V4용 도구	예	를 사용하여 코드에서 IMDSv1 폴백을 명시적으로 비활성화할 수 있습니다`[Amazon.Util.EC2InstanceMetadata]::EC2MetadataV1Disabled = $true` .

프로세스 보안 인증 제공자

> **Note**
>
> 설정 페이지의 레이아웃을 이해하거나 다음에 나오는 Support AWS SDKs이 가이드의 설정 페이지 이해.

SDK는 사용자 지정 사용 사례에 맞게 보안 인증 공급자 체인을 확장할 수 있는 방법을 제공합니다. 이 공급자는 온프레미스 자격 증명 저장소에서 자격 증명을 검색하거나 온프레미스 ID 제공업체와 통합하는 등 사용자 지정 구현을 제공하는 데 사용할 수 있습니다.

예를 들어 IAM Roles Anywhere는 `credential_process`를 사용하여 애플리케이션을 대신하여 임시 자격 증명을 가져옵니다. 이 용도로 `credential_process`를 구성하려면 IAM Roles Anywhere를 사용하여 AWS SDKs 및 도구 인증을(를)참조하십시오.

> **Note**
>
> 다음은 외부 프로세스에서 자격 증명을 소싱하는 방법을 설명하며 외부에서 소프트웨어를 실행하는 경우 사용할 수 있습니다 AWS. AWS 컴퓨팅 리소스를 기반으로 빌드하는 경우 다른 자격 증명 공급자를 사용합니다. 이 옵션을 사용하는 경우 운영 체제의 보안 모범 사례를 사용하여 구성 파일이 최대한 잠겨 있는지 확인해야 합니다. SDKs 및는 이러한 정보를 캡처하고 기록할 AWS CLI 수 `StdErr`있어 권한이 없는 사용자에게 노출될 수 있으므로 사용자 지정 자격 증명 도구가에 비밀 정보를 쓰지 않는지 확인합니다.

다음을 사용하여 이 기능을 구성하십시오.

`credential_process` - 공유 AWS **`config`** 파일 설정

사용할 보안 인증을 생성하거나 검색하기 위해 SDK 또는 도구가 실행하는 외부 명령을 지정합니다. 설정은 SDK가 간접 호출할 프로그램/명령의 이름을 지정합니다. SDK는 프로세스를 간접 호출할 때 프로세스가 `stdout`에 JSON 데이터를 쓸 때까지 기다립니다. 사용자 지정 공급자는 특정 형식으로 정보를 반환해야 합니다. 이 정보에는 SDK 또는 도구가 사용자를 인증하는 데 사용할 수 있는 보안 인증이 포함됩니다.

> **Note**
>
> 프로세스 보안 인증 공급자는 자격 증명 공급자 체인 이해의 일부입니다. 그러나 프로세스 보안 인증 공급자는 이 시리즈에 속한 다른 여러 공급자를 거친 후에만 확인됩니다. 따라서 프로그램에서 이 공급자의 보안 인증을 사용하려면 구성에서 다른 유효한 보안 인증 공급자를 제거하거나 다른 프로파일을 사용해야 합니다. 또는 보안 인증 공급자 체인에 의존하여 유효한 보안 인증을 반환하는 공급자를 자동으로 검색하는 대신 코드에 프로세스 보안 인증 공급자의 사용을 지정하십시오. 서비스 클라이언트를 생성할 때 보안 인증 소스를 직접 지정할 수 있습니다.

보안 인증 프로그램 경로 지정

설정 값은 SDK 또는 개발 도구가 사용자를 대신하여 실행하는 프로그램의 경로를 포함하는 문자열입니다.

- 경로와 파일 이름은 A~Z, a~z, 0~9, 하이픈(-), 밑줄(_), 마침표(.), 슬래시(/), 백슬래시(\)및 공백 등의 문자는 다음과 같습니다.

- 경로 또는 파일 이름에 공백이 있으면 전체 경로와 파일 이름을 큰 따옴표(" ")로 묶습니다.
- 파라미터 이름이나 파라미터 값에 공백이 있으면 해당 요소를 큰 따옴표(" ")로 묶습니다. 전체 페어가 아니라 이름 또는 값만 묶으세요.
- 문자열 안에 환경 변수를 포함하지 마십시오. 예를 들어 $HOME 또는 %USERPROFILE%을 포함할 수 없습니다.
- 홈 폴더를 ~로 지정하지 마십시오. * 전체 경로 또는 기본 파일 이름을 지정해야 합니다. 기본 파일 이름이 있는 경우, 시스템은 PATH 환경 변수로 지정된 폴더 내에서 프로그램을 찾으려고 시도합니다. 경로는 운영 체제에 따라 다릅니다.

  다음 예제는 Linux/macOS의 공유 config 파일에서 credential_process를 설정하는 방법을 보여줍니다.

  ```
  credential_process = "/path/to/credentials.sh" parameterWithoutSpaces "parameter with
   spaces"
  ```

  다음 예제는 Windows의 공유 config 파일에서 credential_process를 설정하는 방법을 보여줍니다.

  ```
  credential_process = "C:\Path\To\credentials.cmd" parameterWithoutSpaces "parameter
   with spaces"
  ```

- 전용 프로필 내에서 지정할 수 있습니다.

  ```
  [profile cred_process]
  credential_process = /Users/username/process.sh
  region = us-east-1
  ```

보안 인증 프로그램에서 유효한 출력

SDK는 프로파일에 지정된 대로 명령을 실행한 다음, 표준 출력 스트림에서 데이터를 읽습니다. 지정한 명령은 스크립트나 바이너리 프로그램이 STDOUT에서 다음 구문과 일치하는 JSON 출력을 생성해야 하는지 여부를 지정합니다.

```
{
    "Version": 1,
    "AccessKeyId": "an AWS access key",
    "SecretAccessKey": "your AWS secret access key",
    "SessionToken": "the AWS session token for temporary credentials",
    "Expiration": "RFC3339 timestamp for when the credentials expire"
```

```
}
```

> **Note**
>
> 이 문서의 작성일 현재, `Version` 키는 1로 설정되어 있습니다. 구조가 발전하면서 시간에 따라 이 값이 증가할 수 있습니다.

`Expiration` 키는 RFC3339 형식의 타임스탬프입니다. `Expiration` 키가 도구의 출력에 존재하지 않으면 SDK는 보안 인증이 새로 고침이 되지 않은 장기 보안 인증이라고 가정합니다. 그렇지 않은 경우 보안 인증은 임시 보안 인증으로 간주되며, 기간이 만료되기 전에 `credential_process` 명령을 다시 실행하면 자동으로 새로 고침됩니다.

> **Note**
>
> SDK는 assume-role 보안 인증을 맡는 방법으로 외부 프로세스 보안 인증을 캐싱하지 않습니다. 캐싱이 필요한 경우에는 외부 프로세스에서 이를 실행해야 합니다.

외부 프로세스는 보안 인증을 검색하는 동안 오류가 발생했음을 나타내기 위해 0이 아닌 반환 코드를 반환할 수 있습니다.

AWS SDKs 도구 지원

다음 SDK는 이 주제에서 설명하는 기능 및 설정을 지원합니다. 모든 일부 예외가 기록됩니다. 모든 JVM 시스템 속성 설정은 AWS SDK for Java 및 AWS SDK for Kotlin 에서만 지원됩니다.

SDK	지원	참고 또는 추가 정보
AWS CLI v2	예	
SDK for C++	예	
SDK for Go V2 (1.x)	예	
SDK for Go 1.x (V1)	예	공유 `config` 파일 설정을 사용하려면 구성 파일에서 로드를 켜야 합니다. 세션을 참조하십시오.

SDK	지원	참고 또는 추가 정보
SDK for Java 2.x	예	
SDK for Java 1.x	예	
SDK for JavaScript 3.x	예	
SDK for JavaScript 2.x	예	
SDK for Kotlin	예	
.NET 4.x용 SDK	예	
SDK for .NET 3.x	예	
SDK for PHP 3.x	예	
SDK for Python (Boto3)	예	
SDK for Ruby 3.x	예	
SDK for Rust	예	
SDK for Swift	예	
PowerShell V5용 도구	예	
PowerShell V4용 도구	예	

AWS SDKs 및 도구 표준화된 기능

많은 기능이 일관된 기본값을 유지하고 여러 SDK에서 동일한 방식으로 작동하도록 표준화되었습니다. 이러한 일관성은 여러 SDK에서 코딩할 때 생산성과 명확성을 높입니다. 코드에서 모든 설정을 재정의할 수 있습니다. 자세한 내용은 사용자 특정 SDK API를 참조하십시오.

> **Important**
>
> 모든 SDK가 모든 기능을 지원하거나 기능 내의 모든 측면을 지원하는 것은 아닙니다.

주제

계정 기반 엔드포인트

> **Note**
>
> 설정 페이지의 레이아웃을 이해하거나 다음에 나오는 Support AWS SDKs이 가이드의 설정 페이지 이해.

계정 기반 엔드포인트는 AWS 계정 ID를 사용하여이 기능을 지원하는 서비스에 대한 요청을 라우팅하여 고성능과 확장성을 보장하는 데 도움이 됩니다. 계정 기반 엔드포인트를 지원하는 AWS SDK 및 서비스를 사용하는 경우 SDK 클라이언트는 리전 엔드포인트가 아닌 계정 기반 엔드포인트를 구성하고 사용합니다. SDK 클라이언트에 계정 ID가 표시되지 않으면 클라이언트는 리전 엔드포인트를 사용합니다. 계정 기반 엔드포인트는 형식입니다. `https://<account-id>.ddb.<region>.amazonaws.com`여기서 *`<account-id>`* 및 *`<region>`*는 AWS 계정 ID 및 입니다 AWS 리전.

다음을 사용하여 이 기능을 구성하십시오.

`aws_account_id` - 공유 AWS **`config`** 파일 설정, **`AWS_ACCOUNT_ID`** - 환경 변수, **`aws.accountId`** - JVM 시스템 속성: Java/Kotlin만 해당

AWS 계정 ID입니다. 계정 기반 엔드포인트 라우팅에 사용됩니다. AWS 계정 ID의 형식은 111122223333입니다.

계정 기반 엔드포인트 라우팅은 일부 서비스에 더 나은 요청 성능을 제공합니다.

`account_id_endpoint_mode` - 공유 AWS **`config`** 파일 설정, **`AWS_ACCOUNT_ID_ENDPOINT_MODE`** - 환경 변수, **`aws.accountIdEndpointMode`** - JVM 시스템 속성: Java/Kotlin만 해당

이 설정은 필요한 경우 계정 기반 엔드포인트 라우팅을 끄고 계정 기반 규칙을 우회하는 데 사용됩니다.

기본값: `preferred`

유효한 값:

- **`preferred`** - 가능한 경우 엔드포인트에 계정 ID가 포함되어야 합니다.
- **`disabled`** - 확인된 엔드포인트에는 계정 ID가 포함되지 않습니다.
- **`required`** - 엔드포인트에는 계정 ID가 포함되어야 합니다. 계정 ID를 사용할 수 없는 경우 SDK에서 오류가 발생합니다.

AWS SDKs 도구 지원

다음 SDK는 이 주제에서 설명하는 기능 및 설정을 지원합니다. 모든 일부 예외가 기록됩니다. 모든 JVM 시스템 속성 설정은 AWS SDK for Java 및 AWS SDK for Kotlin 에서만 지원됩니다.

SDK	지원	SDK 버전에서 릴리스됨	참고 또는 추가 정보
AWS CLI v2	예	2.25.0	
AWS CLI v1	예	1.38.0	
SDK for C++	아니요		
SDK for Go V2 (1.x)	예	v1.35.0	
SDK for Go 1.x (V1)	아니요		
SDK for Java 2.x	예	v2.28.4	
SDK for Java 1.x	예	v1.12.771	
SDK for JavaScript 3.x	예	v3.656.0	
SDK for JavaScript 2.x	아니요		
SDK for Kotlin	예	v1.3.37	
.NET 4.x용 SDK	예	4.0.0	
SDK for .NET 3.x	아니요		
SDK for PHP 3.x	예	v3.318.0	

SDK	지원	SDK 버전에서 릴리스됨	참고 또는 추가 정보
SDK for Python (Boto3)	예	1.37.0	
SDK for Ruby 3.x	예	v1.123.0	
SDK for Rust	아니요		
SDK for Swift	예	1.2.0	
PowerShell V5용 도구	아니요		
PowerShell V4용 도구	아니요		

애플리케이션 ID

> **Note**
>
> 설정 페이지의 레이아웃을 이해하거나 다음에 나오는 Support AWS SDKs이 가이드의 설정 페이지 이해.

여러 고객 애플리케이션에서 단일를 사용하여를 호출할 AWS 계정 수 있습니다 AWS 서비스. 애플리케이션 ID는 고객이를 사용하여 호출 세트를 수행한 소스 애플리케이션을 식별할 수 있는 방법을 제공합니다 AWS 계정. AWS SDKs 및 서비스는이 값을 고객 커뮤니케이션에 다시 표시하는 것 외에는이 값을 사용하거나 해석하지 않습니다. 예를 들어이 값을 운영 이메일 또는에 포함하여 알림과 연결된 애플리케이션을 AWS Health Dashboard 고유하게 식별할 수 있습니다.

다음을 사용하여 이 기능을 구성하십시오.

sdk_ua_app_id - 공유 AWS **config** 파일 설정, **AWS_SDK_UA_APP_ID** - 환경 변수, **sdk.ua.appId** - JVM 시스템 속성: Java/Kotlin만 해당

이 설정은 특정 내의 어떤 애플리케이션이를 AWS 계정 호출하는지 식별하기 위해 애플리케이션에 할당하는 고유한 문자열입니다 AWS.

기본값: None

유효한 값: 최대 길이가 50인 문자열입니다. 문자, 숫자 및 !,#,$,%,&*+-,.,^_`,|, 등의 특수 문자가 허용됩니다~.

config 파일에서 이 값을 설정하는 예:

```
[default]
sdk_ua_app_id=ABCDEF
```

명령 행을 통한 환경 변수 설정의 Linux/macOS 예제:

```
export AWS_SDK_UA_APP_ID=ABCDEF
export AWS_SDK_UA_APP_ID="ABC DEF"
```

명령줄을 통해 환경 변수를 설정하는 Windows 예제:

```
setx AWS_SDK_UA_APP_ID ABCDEF
setx AWS_SDK_UA_APP_ID="ABC DEF"
```

사용 중인 쉘에 특별한 의미가 있는 기호를 포함하는 경우 값을 적절하게 이스케이프합니다.

AWS SDKs 도구 지원

다음 SDK는 이 주제에서 설명하는 기능 및 설정을 지원합니다. 모든 일부 예외가 기록됩니다. 모든 JVM 시스템 속성 설정은 AWS SDK for Java 및 AWS SDK for Kotlin 에서만 지원됩니다.

SDK	지원	참고 또는 추가 정보
AWS CLI v2	예	
SDK for C++	예	공유 config 파일은 지원되지 않습니다.

SDK	지원	참고 또는 추가 정보
SDK for Go V2 (1.x)	예	
SDK for Go 1.x (V1)	아니요	
SDK for Java 2.x	부분	공유 `config` 파일 설정은 지원되지 않으며 환경 변수는 지원되지 않습니다.
SDK for Java 1.x	아니요	
SDK for JavaScript 3.x	예	
SDK for JavaScript 2.x	아니요	
SDK for Kotlin	예	JVM 시스템 속성은 입니다`aws.userAgentAppId` .
.NET 4.x용 SDK	예	
SDK for .NET 3.x	예	
SDK for PHP 3.x	예	
SDK for Python (Boto3)	예	
SDK for Ruby 3.x	예	
SDK for Rust	예	
SDK for Swift	아니요	

SDK	지원	참고 또는 추가 정보
PowerShell V5용 도구	아니요	
PowerShell V4용 도구	아니요	

Amazon EC2 인스턴스 메타데이터

> **Note**
>
> 설정 페이지의 레이아웃을 이해하거나 다음에 나오는 Support AWS SDKs이 가이드의 설정 페이지 이해.

Amazon EC2는 인스턴스 메타데이터 서비스(IMDS)라는 인스턴스 서비스를 제공합니다. 이 서비스에 대한 자세한 내용은 Amazon EC2 사용 설명서의 인스턴스 메타데이터 작업을 참조하세요. IAM 역할을 사용하여 구성한 Amazon EC2 인스턴스에서 보안 인증을 가져오려면 인스턴스 메타데이터 서비스에 대한 연결을 조정할 수 있습니다.

다음을 사용하여 이 기능을 구성하십시오.

`metadata_service_num_attempts` - 공유 AWS **`config`** 파일 설정,
`AWS_METADATA_SERVICE_NUM_ATTEMPTS` - 환경 변수

이 설정은 인스턴스 메타데이터 서비스에서 데이터 검색을 시도할 때 검색 포기하기 전까지의 총 시도 횟수를 지정합니다.

기본값: 1

유효한 값: 1 보다 크거나 같음.

metadata_service_timeout - 공유 AWS **config** 파일 설정,
AWS_METADATA_SERVICE_TIMEOUT - 환경 변수

인스턴스 메타데이터 서비스에서 데이터 검색을 시도할 때 제한 시간 도달까지 걸리는 시간(초)을 지정합니다.

기본값: 1

유효한 값: 1 보다 크거나 같음.

config파일에서 이러한 값을 설정하는 예:

```
[default]
metadata_service_num_attempts=10
metadata_service_timeout=10
```

명령 행을 통한 환경 변수 설정의 Linux/macOS 예제:

```
export AWS_METADATA_SERVICE_NUM_ATTEMPTS=10
export AWS_METADATA_SERVICE_TIMEOUT=10
```

명령줄을 통해 환경 변수를 설정하는 Windows 예제:

```
setx AWS_METADATA_SERVICE_NUM_ATTEMPTS 10
setx AWS_METADATA_SERVICE_TIMEOUT 10
```

AWS SDKs 도구 지원

다음 SDK는 이 주제에서 설명하는 기능 및 설정을 지원합니다. 모든 일부 예외가 기록됩니다. 모든 JVM 시스템 속성 설정은 AWS SDK for Java 및 AWS SDK for Kotlin 에서만 지원됩니다.

SDK	지원	참고 또는 추가 정보
AWS CLI v2	예	
SDK for C++	아니요	

SDK	지원	참고 또는 추가 정보
SDK for Go V2 (1.x)	아니요	
SDK for Go 1.x (V1)	아니요	
SDK for Java 2.x	부분	`AWS_METADATA_SERVICE_TIMEOUT` 만 지원됩니다.
SDK for Java 1.x	부분	`AWS_METADATA_SERVICE_TIMEOUT` 만 지원됩니다.
SDK for JavaScript 3.x	아니요	
SDK for JavaScript 2.x	아니요	
SDK for Kotlin	아니요	
.NET 4.x용 SDK	아니요	
SDK for .NET 3.x	아니요	
SDK for PHP 3.x	예	

SDK	지원	참고 또는 추가 정보
SDK for Python (Boto3)	예	
SDK for Ruby 3.x	아니요	
SDK for Rust	아니요	
SDK for Swift	아니요	
PowerShell V5용 도구	아니요	
PowerShell V4용 도구	아니요	

Amazon S3 액세스 포인트

> **Note**
>
> 설정 페이지의 레이아웃을 이해하거나 다음에 나오는 Support AWS SDKs이 가이드의 설정 페이지 이해.

Amazon S3 서비스는 Amazon S3 버킷과의 상호 작용 대체 방법으로 액세스 포인트를 제공합니다. 액세스 포인트는 버킷에 직접 적용되지 않고 S3 버킷에 고유한 정책 및 구성을 적용할 수 있습니다. AWS SDKs 사용하면 버킷 이름을 명시적으로 지정하는 대신 API 작업에 버킷 필드의 액세스 포인트 Amazon 리소스 이름(ARNs)을 사용할 수 있습니다. 액세스 포인트 ARN과 `GetObject` 을 사용하여

버킷에서 객체를 가져오거나, 액세스 포인트 ARN과 `PutObject`을 사용하여 버킷에 객체를 추가하는 등의 특정 작업에 이 방법을 사용합니다.

Amazon S3 액세스 포인트 및 ARN에 대한 자세한 내용은 Amazon S3 사용 설명서의 액세스 포인트 사용을 참조하십시오.

다음을 사용하여 이 기능을 구성하십시오.

`s3_use_arn_region` - 공유 AWS **`config`** 파일 설정, **`AWS_S3_USE_ARN_REGION`** - 환경 변수, **`aws.s3UseArnRegion`** - JVM 시스템 속성: Java/Kotlin만 해당, 코드에서 값을 직접 구성하려면 특정 SDK를 직접 참조하십시오.

이 설정은 SDK가 액세스 포인트 ARN을 사용하여 요청에 대한 리전 엔드포인트를 AWS 리전 구성하는지 여부를 제어합니다. SDK는 ARN AWS 리전 이 가장 실패할 가능성이 높은 교차 AWS 파티션 호출을 방지 AWS 리전 하도록 구성된 클라이언트의와 동일한 파티션에서 제공되는지 확인합니다. 다중 정의의 경우 코드로 구성된 설정이 우선 적용되고 환경 변수 설정이 그 뒤를 따릅니다.

기본값: `false`

유효한 값:

- **`true`** - SDK는 클라이언트가 구성한 대신 엔드포인트를 구성할 AWS 리전 때 ARN을 사용합니다 AWS 리전. 예외: 클라이언트의 구성 AWS 리전 이 FIPS인 AWS 리전경우 ARN의와 일치해야 합니다 AWS 리전. 이렇게 하지 않으면 오류가 발생합니다.
- **`false`**— SDK는 엔드포인트를 구성할 클라이너트가 구성한 AWS 리전 을 사용합니다.

AWS SDKs 도구 지원

다음 SDK는 이 주제에서 설명하는 기능 및 설정을 지원합니다. 모든 일부 예외가 기록됩니다. 모든 JVM 시스템 속성 설정은 AWS SDK for Java 및 AWS SDK for Kotlin 에서만 지원됩니다.

SDK	지원	참고 또는 추가 정보
AWS CLI v2	예	
SDK for C++	예	
SDK for Go V2 (1.x)	예	

SDK	지원	참고 또는 추가 정보
SDK for Go 1.x (V1)	예	공유 config 파일 설정을 사용하려면 구성 파일에서 로드를 켜야 합니다. 세션을 참조하십시오.
SDK for Java 2.x	예	
SDK for Java 1.x	예	JVM 시스템 속성은 지원되지 않습니다.
SDK for JavaScript 3.x	예	
SDK for JavaScript 2.x	예	
SDK for Kotlin	예	
.NET 4.x용 SDK	예	
SDK for .NET 3.x	예	표준 우선 순위를 따르지 않습니다. 공유 config 파일 값이 환경 변수보다 우선합니다.
SDK for PHP 3.x	예	
SDK for Python (Boto3)	예	
SDK for Ruby 3.x	예	
SDK for Rust	아니요	
SDK for Swift	아니요	
PowerShell V5용 도구	예	표준 우선 순위를 따르지 않습니다. 공유 config 파일 값이 환경 변수보다 우선합니다.
PowerShell V4용 도구	예	표준 우선 순위를 따르지 않습니다. 공유 config 파일 값이 환경 변수보다 우선합니다.

Amazon S3 다중 리전 액세스 포인트

> **Note**
>
> 설정 페이지의 레이아웃을 이해하거나 다음에 나오는 Support AWS SDKs이 가이드의 설정 페이지 이해.

Amazon S3 다중 리전 액세스 포인트는 애플리케이션이 여러 AWS 리전리전에 있는 Amazon S3 버킷의 요청을 이행하는 데 사용할 수 있는 글로벌 엔드포인트를 제공합니다. 다중 리전 액세스 포인트를 사용하여 단일 리전에서 사용되는 것과 동일한 아키텍처로 다중 리전 애플리케이션을 구축하면 전 세계 어디에서나 해당 애플리케이션을 실행할 수 있습니다.

다중 리전 액세스 포인트에 대한 자세한 내용을 알아보려면 Amazon S3 사용 설명서의 Amazon S3의 다중 리전 액세스 포인트를 참조하십시오.

다중 리전 액세스 포인트 Amazon 리소스 이름(ARN)에 대한 자세한 내용을 알아보려면 Amazon S3 사용 설명서의 다중 리전 액세스 포인트를 사용하여 요청 생성하기을 참조하십시오.

다중 리전 액세스 포인트 생성에 대한 자세한 내용을 알아보려면 Amazon S3 사용 설명서의 Amazon S3의 다중 리전 액세스 포인트 관리를 참조하십시오.

SigV4A 알고리즘은 글로벌 리전 요청에 서명하는 데 사용되는 서명 구현입니다. 이 알고리즘은 AWS 공통 런타임(CRT) 라이브러리에 대한 종속성을 통해 SDK로 획득됩니다.

다음을 사용하여 이 기능을 구성하십시오.

`s3_disable_multiregion_access_points` - 공유 AWS **`config`** 파일 설정, **`AWS_S3_DISABLE_MULTIREGION_ACCESS_POINTS`** - 환경 변수, **`aws.s3DisableMultiRegionAccessPoints`** - JVM 시스템 속성: Java/Kotlin만 해당, 코드에서 값을 직접 구성하려면 특정 SDK를 직접 참조하십시오.

이 설정은 SDK가 잠재적으로 리전 간 요청을 시도할지 여부를 제어합니다. 다중 정의의 경우 코드로 구성된 설정이 우선 적용되고 환경 변수 설정이 그 뒤를 따릅니다.

기본값: `false`

유효값:

- **`true`**— 리전 간 요청 사용을 중지합니다.
- **`false`**— 다중 리전 액세스 포인트를 사용하여 리전 간 요청을 활성화합니다.

AWS SDKs 도구 지원

다음 SDK는 이 주제에서 설명하는 기능 및 설정을 지원합니다. 모든 일부 예외가 기록됩니다. 모든 JVM 시스템 속성 설정은 AWS SDK for Java 및 AWS SDK for Kotlin 에서만 지원됩니다.

SDK	지원	참고 또는 추가 정보
AWS CLI v2	예	
SDK for C++	예	
SDK for Go V2 (1.x)	예	
SDK for Go 1.x (V1)	아니요	
SDK for Java 2.x	예	
SDK for Java 1.x	아니요	
SDK for JavaScript 3.x	예	
SDK for JavaScript 2.x	아니요	
SDK for Kotlin	예	
.NET 4.x용 SDK	예	
SDK for .NET 3.x	예	
SDK for PHP 3.x	예	
SDK for Python (Boto3)	예	
SDK for Ruby 3.x	예	

SDK	지원	참고 또는 추가 정보
SDK for Rust	예	
SDK for Swift	아니요	
PowerShell V5용 도구	예	
PowerShell V4용 도구	예	

S3 Express One Zone 세션 인증

> **Note**
>
> 설정 페이지의 레이아웃을 이해하거나 다음에 나오는 Support AWS SDKs이 가이드의 설정 페이지 이해.

S3 Express One Zone은 자주 액세스하는 데이터에 대해 10밀리초 미만의 지연 시간을 제공하는 Amazon S3의 고성능 스토리지 클래스입니다. S3 Express One Zone 버킷을 사용하는 경우 AWS SDKs 및 도구는 데이터 요청의 지연 시간이 짧은 권한 부여에 최적화된 세션 기반 인증을 자동으로 사용합니다. 영역(객체 수준) 작업과 함께 세션 토큰을 사용하여 세션의 여러 요청에 대해 권한 부여와 관련된 지연 시간을 분산하여 인증 오버헤드를 줄이고 전체 요청 성능을 개선할 수 있습니다.

S3 Express One Zone 버킷은와 같은 가용 영역 ID를 포함하는 특정 이름 지정 형식을 사용합니다bucket-name--usw2-az1--x-s3. SDK는이 이름 지정 패턴을 감지하면 요청을 적절한 S3 Express One Zone 엔드포인트로 자동으로 라우팅하고 최적화된 인증 흐름을 적용합니다. 세션 인증은 버킷에 대한 지연 시간이 짧은 액세스를 제공하는 임시 버킷별 자격 증명을 생성하며 SDK에 의해 자동으로 캐시되고 새로 고쳐집니다. 자세한 내용은 Amazon S3 사용 설명서의 S3 Express One Zone을 참조하세요. Amazon S3

기본적으로 세션 인증은 S3 Express One Zone 버킷에 대해 활성화됩니다.

다음을 사용하여 이 기능을 구성하십시오.

s3_disable_express_session_auth - 공유 AWS **config** 파일 설정, **AWS_S3_DISABLE_EXPRESS_SESSION_AUTH** - 환경 변수, **aws.disableS3ExpressAuth** - JVM 시스템 속성: Java/Kotlin만 해당

S3 Express One Zone 세션 인증의 비활성화 여부를 제어합니다. 로 설정하면 SDK`true`는 세션 인증 대신 S3 Express One Zone 버킷에 표준 SigVSigV4 인증을 사용합니다.

기본값: `false`

유효한 값:

- **true** - S3 Express One Zone 세션 인증을 비활성화합니다.
- **false** - S3 Express One Zone 세션 인증을 활성화합니다.

`config` 파일에서 이 값을 설정하는 예:

```
[default]
s3_disable_express_session_auth=true
```

명령 행을 통한 환경 변수 설정의 Linux/macOS 예제:

```
export AWS_S3_DISABLE_EXPRESS_SESSION_AUTH=true
```

명령줄을 통해 환경 변수를 설정하는 Windows 예제:

```
setx AWS_S3_DISABLE_EXPRESS_SESSION_AUTH true
```

AWS SDKs 도구 지원

다음 SDK는 이 주제에서 설명하는 기능 및 설정을 지원합니다. 모든 일부 예외가 기록됩니다. 모든 JVM 시스템 속성 설정은 AWS SDK for Java 및 AWS SDK for Kotlin 에서만 지원됩니다.

SDK	지원	참고 또는 추가 정보
AWS CLI v2	예	

SDK	지원	참고 또는 추가 정보
AWS CLI v1	아니요	
SDK for C++	예	
SDK for Go V2 (1.x)	예	
SDK for Go 1.x (V1)	아니요	공유 config 파일 설정을 사용하려면 구성 파일에서 로드를 켜야 합니다. 세션을 참조하십시오.
SDK for Java 2.x	예	
SDK for Java 1.x	아니요	
SDK for JavaScript 3.x	예	
SDK for JavaScript 2.x	아니요	
SDK for Kotlin	예	JVM 시스템 속성은 입니다aws.s3DisableExpressSessionAuth .
.NET 4.x용 SDK	예	
SDK for .NET 3.x	예	
SDK for PHP 3.x	예	
SDK for Python (Boto3)	예	
SDK for Ruby 3.x	예	

SDK	지원	참고 또는 추가 정보
SDK for Rust	예	
SDK for Swift	예	
PowerShell V5용 도구	예	
PowerShell V4용 도구	예	

인증 체계

> **Note**
>
> 설정 페이지의 레이아웃을 이해하거나 다음에 나오는 Support AWS SDKs이 가이드의 설정 페이지 이해.

AWS 서비스는 AWS 서명 버전 4(SigV4) 및 AWS 서명 버전 4a(SigV4a)와 같은 여러 인증 체계를 지원합니다. 기본적으로 SDKs 서비스 모델 정의를 기반으로 인증 체계를 선택하고 최상의 호환성을 제공하는 체계의 우선순위를 지정합니다. 그러나 특정 요구 사항에 맞게 최적화하도록 선호하는 인증 체계를 구성할 수 있습니다.

SigV4와 달리 SigV4a로 서명된 요청은 여러에서 유효합니다 AWS 리전. SigV4a는 리전 간 요청 서명을 통해 향상된 가용성을 제공하므로 리전 중단 시 백업 리전으로 자동 장애 조치가 가능합니다. 이는 AWS Identity and Access Management 또는 Amazon CloudFront와 같은 글로벌 서비스에 특히 유용합니다.

이 두 인증 체계에 대한 자세한 내용은 IAM 사용 설명서의 AWS API 요청에 대한 서명 버전 4를 참조하세요.

다음을 사용하여 이 기능을 구성하십시오.

auth_scheme_preference - 공유 AWS **config** 파일 설정, **AWS_AUTH_SCHEME_PREFERENCE** - 환경 변수, **aws.authSchemePreference** - JVM 시스템 속성: Java/Kotlin만 해당

우선 순위에 따라 쉼표로 구분된 기본 인증 체계 목록을 지정합니다. 서비스가 여러 인증 체계를 지원하는 경우 SDK는 지정된 순서로이 목록의 체계를 사용하려고 시도하며, 원하는 체계를 사용할 수 없는 경우 기본 동작으로 돌아갑니다.

기본값: 없음.

유효한 값: 다음 중 하나 이상의 쉼표로 구분된 목록입니다.

- **sigv4** - 서명 버전 4(가장 빠른 성능, 단일 리전)
- **sigv4a** - 서명 버전 4a(향상된 가용성, 리전 간 지원, SigV4보다 느린 서명 성능)
- **httpBearerAuth** - HTTP 베어러 토큰 인증

스키마 이름 사이의 공백 및 탭 문자는 무시됩니다.

SigV4a를 선호하도록 config 파일에서이 값을 설정하는 예:

```
[default]
auth_scheme_preference=sigv4a,sigv4
```

sigv4a_signing_region_set - 공유 AWS **config** 파일 설정, **AWS_SIGV4A_SIGNING_REGION_SET** - 환경 변수

SigV4a 다중 리전 서명을 AWS 리전 위한 쉼표로 구분된 목록을 지정합니다. SigV4a가 선택한 인증 체계인 경우 요청에 대해 설정된 기본 리전으로 사용됩니다.

기본값: 요청에 의해 결정됩니다.

유효한 값: 쉼표로 구분된 목록입니다 AWS 리전. 리전 간 공백 및 탭 문자는 무시됩니다.

AWS SDKs 도구 지원

다음 SDK는 이 주제에서 설명하는 기능 및 설정을 지원합니다. 모든 일부 예외가 기록됩니다. 모든 JVM 시스템 속성 설정은 AWS SDK for Java 및 AWS SDK for Kotlin 에서만 지원됩니다.

SDK	지원	참고 또는 추가 정보
AWS CLI v2	예	

SDK	지원	참고 또는 추가 정보
SDK for C++	아니요	
SDK for Go V2 (1.x)	예	
SDK for Go 1.x (V1)	아니요	
SDK for Java 2.x	예	
SDK for Java 1.x	아니요	
SDK for JavaScript 3.x	예	
SDK for JavaScript 2.x	아니요	
SDK for Kotlin	예	
.NET 4.x용 SDK	아니요	
SDK for .NET 3.x	아니요	
SDK for PHP 3.x	예	
SDK for Python (Boto3)	예	
SDK for Ruby 3.x	예	

SDK	지원	참고 또는 추가 정보
SDK for Rust	예	
SDK for Swift	예	
PowerShell V5용 도구	아니요	
PowerShell V4용 도구	아니요	

AWS 리전

> **Note**
>
> 설정 페이지의 레이아웃을 이해하거나 다음에 나오는 Support AWS SDKs이 가이드의 설정 페이지 이해.

AWS 리전 는 작업 시 이해해야 할 중요한 개념입니다 AWS 서비스.

를 사용하면 특정 지리적 영역에 물리적으로 상주 AWS 서비스 하는에 액세스할 AWS 리전수 있습니다. 이는 데이터와 애플리케이션이 귀사 및 귀사의 사용자가 액세스하기 가까운 곳에서 계속 실행되도록 하는 데 도움이 될 수 있습니다. 리전에서는 내결함성, 안정성 및 복원성을 지원하고 지연 시간을 줄일 수도 있습니다. 리전을 통해 사용자는 가용 상태를 유지하며 리전 중단의 영향을 받지 않는 중복 리소스를 생성할 수 있습니다.

대부분의 AWS 서비스 요청은 특정 지리적 리전과 연결됩니다. 한 리전에서 생성한 리소스는 AWS 서비스서비스에서 제공하는 복제 기능을 명시적으로 사용하지 않는 한 다른 리전에 존재하지 않습니다. 예를 들어, Amazon S3와 Amazon EC2 크로스 리전 복제를 지원합니다. IAM과 같은 일부 서비스의 경우 리전 리소스가 없습니다.

AWS 일반 참조에는 다음 정보가 포함됩니다.

- 리전과 엔드포인트 간의 관계를 이해하고 기존 리전 엔드포인트 목록을 보려면 AWS 서비스 엔드포인트를 참조하십시오.
- 각각의 AWS 서비스에 대해 지원되는 모든 리전 및 엔드포인트의 현재 목록을 보려면 서비스 엔드포인트 및 할당량을 참조하십시오.

서비스 클라이언트 생성

프로그래밍 방식으로에 액세스하기 위해 AWS 서비스 SDKs 각에 대해 클라이언트 클래스/객체를 사용합니다 AWS 서비스. 예를 들어 애플리케이션에서 Amazon EC2에 액세스해야 하는 경우 애플리케이션은 Amazon EC2 클라이언트 객체를 생성하여 해당 서비스와 인터페이스합니다.

코드 자체에서 클라이언트에 대해 명시적으로 지정된 리전이 없는 경우 클라이언트는 기본적으로 다음 `region` 설정을 통해 설정된 리전을 사용합니다. 하지만 개별 클라이언트 객체에 대해 클라이언트의 활성 리전을 명시적으로 설정할 수 있습니다. 이러한 방식으로 리전을 설정하면 특별한 서비스 클라이언트에 대한 전역 설정에 우선합니다. 대체 리전은 해당 클라이언트를 인스턴스화하는 동안 SDK에 따라 지정됩니다(특정 SDK 가이드 또는 SDK의 코드 베이스 확인).

다음을 사용하여 이 기능을 구성하십시오.

`region` - 공유 AWS **`config`** 파일 설정, **`AWS_REGION`** - 환경 변수, **`aws.region`** - JVM 시스템 속성: Java/Kotlin만 해당

AWS 요청에 사용할 기본 AWS 리전 값을 지정합니다. 이 리전은 사용할 특정 지역과 함께 제공되지 않은 SDK 서비스 요청에 사용됩니다.

기본값: 없음. 이 값을 명시적으로 지정해야 합니다.

유효값:

- AWS 일반 참조의 AWS 서비스 엔드포인트에 나열된 대로 선택한 서비스에서 사용할 수 있는 모든 리전 코드. 예를 들어, `us-east-1` 값은 엔드포인트를 AWS 리전 미국 동부(버지니아 북부)로 설정합니다.
- `aws-global`는 AWS Security Token Service (AWS STS) 및 Amazon Simple Storage Service(Amazon S3)와 같은 리전 엔드포인트 외에도 별도의 글로벌 엔드포인트를 지원하는 서비스에 대한 글로벌 엔드포인트를 지정합니다.

`config` 파일에서 이 값을 설정하는 예:

```
[default]
```

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

명령 행을 통한 환경 변수 설정의 Linux/macOS 예제:

```
export AWS_REGION=us-west-2
```

명령줄을 통해 환경 변수를 설정하는 Windows 예제:

```
setx AWS_REGION us-west-2
```

대부분의 SDK에는 애플리케이션 코드 내에서 기본 리전을 설정하는 데 사용할 수 있는 "구성" 객체가 있습니다. 자세한 내용은 특정 AWS SDK 개발자 안내서를 참조하세요.

AWS SDKs 도구 지원

다음 SDK는 이 주제에서 설명하는 기능 및 설정을 지원합니다. 모든 일부 예외가 기록됩니다. 모든 JVM 시스템 속성 설정은 AWS SDK for Java 및 AWS SDK for Kotlin 에서만 지원됩니다.

SDK	지원	참고 또는 추가 정보
AWS CLI v2	예	AWS CLI v2는의 값을의 값AWS_REGION 보다 먼저 사용합니다AWS_DEFAULT_REGION (두 변수 모두 확인됨).
AWS CLI v1	예	AWS CLI v1은이 용도로 AWS_DEFAULT_REGION 라는 환경 변수를 사용합니다.
SDK for C++	예	
SDK for Go V2 (1.x)	예	
SDK for Go 1.x (V1)	예	공유 config 파일 설정을 사용하려면 구성 파일에서 로드를 켜야 합니다. 세션을 참조하십시오.
SDK for Java 2.x	예	
SDK for Java 1.x	예	
SDK for JavaScript 3.x	예	

SDK	지원	참고 또는 추가 정보
SDK for JavaScript 2.x	예	
SDK for Kotlin	예	
.NET 4.x용 SDK	예	
SDK for .NET 3.x	예	
SDK for PHP 3.x	예	
SDK for Python (Boto3)	예	이 SDK는 이러한 용도로 명명된 `AWS_DEFAULT_REGION` 환경 변수를 사용합니다.
SDK for Ruby 3.x	예	
SDK for Rust	예	
SDK for Swift	예	
PowerShell V5용 도구	예	
PowerShell V4용 도구	예	

AWS STS 리전 엔드포인트

> **Note**
>
> 설정 페이지의 레이아웃을 이해하거나 다음에 나오는 Support AWS SDKs이 가이드의 설정 페이지 이해.

AWS Security Token Service (AWS STS)는 글로벌 및 리전 서비스 모두로 사용할 수 있습니다. 일부 AWS SDKs 및 CLIs 기본적으로 글로벌 서비스 엔드포인트(`https://sts.amazonaws.com`)를 사용하는 반면, 일부는 리전 서비스 엔드포인트()를 사용합니다`https://sts.{region_identifier}.{partition_domain}`. 기본적으로 활성화된 리전에서 AWS STS 글로벌 엔드포인트에 대한 요청은 요청이 시작된 리전과 동일한 리전에서 자동으로 처리됩니다. 옵트

인 리전에서 AWS STS 글로벌 엔드포인트에 대한 요청은 단일 AWS 리전미국 동부(버지니아 북부)에서 처리됩니다. AWS STS 엔드포인트에 대한 자세한 내용은 AWS Security Token Service API 참조의 엔드포인트 또는 AWS Identity and Access Management 사용 설명서의 AWS STS 의 관리를 AWS 리전 참조하세요.

가능하면 리전 엔드포인트를 사용하고를 구성하는 것이 AWS 가장 좋습니다AWS 리전. 상용이 아닌 파티션의 고객은 리전 엔드포인트를 사용해야 합니다. 모든 SDKs 및 도구가이 설정을 지원하는 것은 아니지만 글로벌 및 리전 엔드포인트에 대해 정의된 동작이 있습니다. 자세한 내용은 다음 섹션을 참조하세요.

> **Note**
>
> AWS 는 복원력과 성능을 향상시키기 위해 기본적으로 활성화된 리전의 AWS Security Token Service (AWS STS) 글로벌 엔드포인트(`https://sts.amazonaws.com`)를 변경했습니다. 글로벌 엔드포인트에 대한 AWS STS 요청은 워크로드 AWS 리전 와 동일한에서 자동으로 제공됩니다. 이러한 변경 사항은 옵트인 리전에 배포되지 않습니다. 적절한 AWS STS 리전 엔드포인트를 사용하는 것이 좋습니다. 자세한 내용은 AWS Identity and Access Management 사용 설명서의 AWS STS 글로벌 엔드포인트 변경 사항을 참조하세요.

이 설정을 지원하는 SDKs 및 도구의 경우 고객은 다음을 사용하여 기능을 구성할 수 있습니다.

`sts_regional_endpoints` - 공유 AWS **`config`** 파일 설정, **`AWS_STS_REGIONAL_ENDPOINTS`** - 환경 변수

이 설정은 SDK 또는 도구가 AWS Security Token Service ()와 통신하는 데 사용하는 엔드포인트를 결정하는 AWS 서비스 방법을 지정합니다AWS STS.

기본값: `regional`, 다음 표의 예외를 참조하세요.

> **Note**
>
> 2022년 7월 이후에 출시되는 모든 새 SDK 메이저 버전은 `regional`으로 기본값이 설정됩니다. 새 SDK 메이저 버전에서는 `regional` 동작을 사용하여 이 설정을 없앨 수 있습니다. 이 변경으로 인한 향후 영향을 줄이려면 가능하면 사용자 애플리케이션에서`regional`을 사용하여 시작하는 것이 좋습니다.

유효한 값:, (권장 값: *`regional`*)

- **`legacy`** - 글로벌 AWS STS 엔드포인트를 사용합니다`sts.amazonaws.com`.
- **`regional`** - SDK 또는 도구는 항상 현재 구성된 리전의 AWS STS 엔드포인트를 사용합니다. 예를 들어 클라이언트가를 사용하도록 구성된 경우 `us-west-2`에 대한 모든 호출 AWS STS 은 글로벌 엔드포인트 `sts.us-west-2.amazonaws.com`대신 리전 `sts.amazonaws.com` 엔드포인트에 대해 이루어집니다. 이 설정이 활성화된 상태에서 글로벌 엔드포인트에 요청을 보내려면 리전을 `aws-global`로 설정하면 됩니다.

`config` 파일에서 이러한 값을 설정하는 예:

```
[default]
sts_regional_endpoints = regional
```

명령 행을 통한 환경 변수 설정의 Linux/macOS 예제:

```
export AWS_STS_REGIONAL_ENDPOINTS=regional
```

명령줄을 통해 환경 변수를 설정하는 Windows 예제:

```
setx AWS_STS_REGIONAL_ENDPOINTS regional
```

AWS SDKs 도구 지원

> **Note**
>
> 가능하면 리전 엔드포인트를 사용하고를 구성하는 것이 AWS 가장 좋습니다AWS 리전.

다음 표에는 SDK 또는 도구에 대한 요약이 나와 있습니다.

- 지원 설정: STS 리전 엔드포인트에 대한 공유 `config` 파일 변수 및 환경 변수가 지원되는지 여부입니다.
- 기본 설정 값: 지원되는 경우 설정의 기본값입니다.
- 기본 서비스 클라이언트 대상 STS 엔드포인트: 설정을 변경할 수 없는 경우에도 클라이언트가 사용하는 기본 엔드포인트입니다.
- 서비스 클라이언트 폴백 동작: 리전 엔드포인트를 사용해야 하지만 리전이 구성되지 않은 경우 SDK가 수행하는 작업입니다. 이는 기본값으로 인해 리전 엔드포인트를 사용하는지 또는 설정에 의해 `regional`가 선택되었기 때문에 리전 엔드포인트를 사용하는지에 관계없이 동작입니다.

테이블은 다음 값도 사용합니다.

- 글로벌 엔드포인트: `https://sts.amazonaws.com`.
- 리전 엔드포인트: 애플리케이션에서 AWS 리전 구성된를 기반으로 합니다.
- **`us-east-1`** (리전): `us-east-1` 리전 엔드포인트를 사용하지만 일반적인 글로벌 요청보다 세션 토큰이 더 깁니다.

SDK		기본 설정 값	기본 서비스 클라이언트 대상 STS 엔드포인트	서비스 클라이언트 폴백 동작	참고 또는 추가 정보
AWS CLI v2	이	N/A	리전 엔드포인트	글로벌 엔드포인트	
AWS CLI v1	아	`legacy`	글로벌 엔드포인트	글로벌 엔드포인트	
SDK for C++	이	N/A	리전 엔드포인트	`us-east-1` (리전)	
SDK for Go V2 (1.x)	이	N/A	리전 엔드포인트	요청 실패	
SDK for Go 1.x (V1)	아	`legacy`	글로벌 엔드포인트	글로벌 엔드포인트	공유 `config` 파일 설정을 사용하려면 구성 파일에서 로드를 켜야 합니다. 세션을 참조하십시오.
SDK for Java 2.x	이	N/A	리전 엔드포인트	요청 실패	리전이 구성되지 않은 경우 `AssumeRole` 및 `AssumeRoleWithWebIdentity` 는 글로벌 STS 엔드포인트를 사용합니다.

SDK		기본 설정 값	기본 서비스 클라이언트 대상 STS 엔드포인트	서비스 클라이언트 폴백 동작	참고 또는 추가 정보
SDK for Java 1.x	아	`legacy`	글로벌 엔드포인트	글로벌 엔드포인트	
SDK for JavaScript 3.x	이 니 오	N/A	리전 엔드포인트	요청 실패	
SDK for JavaScript 2.x	아	`legacy`	글로벌 엔드포인트	글로벌 엔드포인트	
SDK for Kotlin	이 니 오	N/A	리전 엔드포인트	글로벌 엔드포인트	
.NET 4.x용 SDK	이 니 오	N/A	리전 엔드포인트	`us-east-1` (리전)	
SDK for .NET 3.x	아	`regional`	글로벌 엔드포인트	글로벌 엔드포인트	
SDK for PHP 3.x	아	`regional`	글로벌 엔드포인트	요청 실패	
SDK for Python (Boto3)	아	`regional`	글로벌 엔드포인트	글로벌 엔드포인트	
SDK for Ruby 3.x	아	`regional`	리전 엔드포인트	요청 실패	

SDK		기본 설정 값	기본 서비스 클라이언트 대상 STS 엔드포인트	서비스 클라이언트 폴백 동작	참고 또는 추가 정보
SDK for Rust	이	N/A	리전 엔드포인트	요청 실패	
SDK for Swift	이	N/A	리전 엔드포인트	요청 실패	
PowerShell V5용 도구	아	`regional`	글로벌 엔드포인트	글로벌 엔드포인트	
PowerShell V4용 도구	아	`regional`	글로벌 엔드포인트	글로벌 엔드포인트	

Amazon S3에 대한 데이터 무결성 보호

> **Note**
>
> 설정 페이지의 레이아웃을 이해하거나 다음에 나오는 Support AWS SDKs이 가이드의 설정 페이지 이해.

일정 기간 동안 AWS SDKs는 Amazon Simple Storage Service에 데이터를 업로드하거나 Amazon Simple Storage Service에서 데이터를 다운로드할 때 데이터 무결성 검사를 지원했습니다. 이전에는 이러한 검사가 옵트인되었습니다. 이제 CRC32 또는 CRC64NVME와 같은 CRC 기반 알고리즘을 사용하여 이러한 검사를 기본적으로 활성화했습니다. 각 SDK 또는 도구에는 기본 알고리즘이 있지만 다른 알고리즘을 선택할 수 있습니다. 원하는 경우 업로드에 대해 미리 계산된 체크섬을 계속 수동으로 제공할 수도 있습니다. 업로드, 멀티파트 업로드, 다운로드 및 암호화 모드 전반에서 일관된 동작은 클라이언트 측 무결성 검사를 간소화합니다.

최신 버전의 AWS SDKs 및는 각 업로드에 대해 순환 중복 검사(CRC) 기반 체크섬을 AWS CLI 자동으로 계산하여 Amazon S3로 전송합니다. Amazon S3는 객체와 체크섬을 객체의 메타데이터에 내구성

있게 저장하기 전에 서버 측 체크섬을 독립적으로 계산하고 제공된 값과 비교하여 검증합니다. 객체와 함께 메타데이터에 체크섬을 저장하면 객체를 다운로드할 때 동일한 체크섬을 자동으로 반환하고 다운로드를 검증하는 데 사용할 수 있습니다. 객체의 메타데이터에 저장된 체크섬은 언제든지 확인할 수도 있습니다.

체크섬 작업, 멀티파트 업로드 또는 지원되는 체크섬 알고리즘 목록에 대한 자세한 내용은 Amazon Simple Storage Service 사용 설명서의 Amazon S3에서 객체 무결성 확인을 참조하세요.

멀티파트 업로드:

또한 Amazon S3는 개발자에게 단일 파트 및 멀티파트 업로드에서 일관된 전체 객체 체크섬을 제공합니다.

여러 파트로 파일을 업로드할 때 SDKs 각 파트의 체크섬을 계산합니다. Amazon S3는 이러한 체크섬을 사용하여 `UploadPart` API를 통해 각 부분의 무결성을 확인합니다. 또한 Amazon S3는 `CompleteMultipartUpload` API를 호출할 때 전체 파일의 크기와 체크섬을 검증합니다.

SDK에 멀티파트 업로드를 지원하는 Amazon S3 Transfer Manager가 있는 경우 AWS SDKs 도구 지원 표에 있는 SDK별 기본 알고리즘을 사용하여 부분에 대한 체크섬이 검증됩니다. 설정을 로 설정`checksum_type``FULL_OBJECT`하거나 CRC64NVME 알고리즘을 사용하도록 선택하여 전체 객체 체크섬에 옵트인할 수 있습니다.

이전 버전의 SDK를 사용하는 경우 AWS CLI:

애플리케이션이 SDK 또는 도구의 2024년 12월 이전 버전을 사용하는 경우 Amazon S3는 여전히 새 객체에 대한 CRC64NVME 체크섬을 계산하고 나중에 참조할 수 있도록 객체 메타데이터에 저장합니다. 나중에 저장된 CRC를 사용자 측에서 계산된 CRC와 비교하고 네트워크 전송이 올바른지 확인할 수 있습니다. 또한 이전 버전에서 이를 해결하기 위한 표준 기술인 `PutObject` 또는 `UploadPart` 요청과 함께 사전 계산된 자체 체크섬을 제공하여 무결성 보호를 수동으로 확장할 수 있습니다.

다음을 사용하여 이 기능을 구성하십시오.

`request_checksum_calculation` - 공유 AWS **`config`** 파일 설정, **`AWS_REQUEST_CHECKSUM_CALCULATION`** - 환경 변수, **`aws.requestChecksumCalculation`** - JVM 시스템 속성: Java/Kotlin만 해당

> 기본적으로 사용자는 요청을 보낼 때 요청 체크섬을 계산하도록 옵트인됩니다. 사용자는 요청 빌드의 일부로 사용 가능한 체크섬 알고리즘 중 하나를 선택할 수 있습니다. 그렇지 않으면 SDK별 기본 알고리즘이 사용됩니다. 각 SDK 또는 도구의 기본 알고리즘은 AWS SDKs 도구 지원표를 참조하세요.

기본값: `WHEN_SUPPORTED`

유효한 값:

- **`WHEN_SUPPORTED`** - 체크섬 검증은 Amazon S3로의 데이터 전송과 같은 API 작업에서 지원되는 경우 모든 요청 페이로드에 대해 수행됩니다.
- **`WHEN_REQUIRED`** - 체크섬 검증은 API 작업에 필요한 경우에만 수행됩니다.

`response_checksum_validation` - 공유 AWS **`config`** 파일 설정, **`AWS_RESPONSE_CHECKSUM_VALIDATION`** - 환경 변수, **`aws.responseChecksumValidation`** - JVM 시스템 속성: Java/Kotlin만 해당

기본적으로 사용자는 요청을 보낼 때 응답 체크섬 검증에 옵트인됩니다. 체크섬은 응답 페이로드에 대해 계산되고 체크섬 응답 헤더와 비교됩니다. 체크섬 검증에 실패하면 페이로드를 읽을 때 사용자에게 오류가 발생합니다.

체크섬 응답 헤더는 체크섬의 알고리즘도 나타냅니다. Amazon S3 클라이언트는 체크섬을 지원하는 모든 Amazon S3 API 작업에 대한 응답 체크섬을 검증하려고 시도합니다. 그러나 SDK가 지정된 체크섬 알고리즘을 구현하지 않은 경우이 검증은 건너뜁니다.

기본값: `WHEN_SUPPORTED`

유효한 값:

- **`WHEN_SUPPORTED`** - Amazon S3로의 데이터 전송과 같은 API 작업에서 지원하는 경우 모든 응답 페이로드에 대해 체크섬 검증이 수행됩니다.
- **`WHEN_REQUIRED`** - 체크섬 검증은 API 작업에서 지원되고 호출자가 작업에 체크섬을 명시적으로 활성화한 경우에만 수행됩니다. 예를 들어 Amazon S3 `GetObject` API가 호출되고 `ChecksumMode` 파라미터가 활성화됨으로 설정된 경우입니다.

AWS SDKs 도구 지원

다음 SDK는 이 주제에서 설명하는 기능 및 설정을 지원합니다. 모든 일부 예외가 기록됩니다. 모든 JVM 시스템 속성 설정은 AWS SDK for Java 및 AWS SDK for Kotlin 에서만 지원됩니다.

> **Note**
>
> 다음 표에서 'CRT'는를 참조AWS 공통 런타임(CRT) 라이브러리하며 프로젝트에 추가 종속성을 추가해야 할 수 있습니다.

SDK	지원	기본 체크섬 알고리즘	지원되는 체크섬 알고리즘	참고 또는 추가 정보
AWS CLI v2	예	CRC64NVME	CRC64NVME, CRC32, CRC32C, SHA1, SHA256	AWS CLI v1의 경우 기본 알고리즘과 지원되는 알고리즘은 Python(Boto3)과 동일합니다.
SDK for C++	예	CRC64NVME	CRC64NVME, CRC32, CRC32C, SHA1, SHA256	
SDK for Go V2 (1.x)	예	CRC32	CRC64NVME, CRC32, CRC32C, SHA1, SHA256	
SDK for Go 1.x (V1)	아니요			
SDK for Java 2.x	예	CRC32	CRC64NVME(CRT 만 경유), CRC32, CRC32C, SHA1, SHA256	
SDK for Java 1.x	아니요			
SDK for JavaScript 3.x	예	CRC32	CRC32, CRC32C, SHA1, SHA256	
SDK for JavaScript 2.x	아니요			
SDK for Kotlin	예	CRC32	CRC32, CRC32C, SHA1, SHA256	

SDK	지원	기본 체크섬 알고리즘	지원되는 체크섬 알고리즘	참고 또는 추가 정보
.NET 4.x용 SDK	예	CRC32	CRC32, CRC32C, SHA1, SHA256	
SDK for .NET 3.x	예	CRC32	CRC32, CRC32C, SHA1, SHA256	
SDK for PHP 3.x	예	CRC32	CRC32, CRC32C(CRT를 통해서만), SHA1, SHA256	`awscrt` CRC32C를 사용하려면 확장이 필요합니다.
SDK for Python (Boto3)	예	CRC32	CRC64NVME(CRT를 통해서만), CRC32, CRC32C(CRT를 통해서만), SHA1, SHA256	
SDK for Ruby 3.x	예	CRC32	CRC64NVME(CRT를 통해서만), CRC32, CRC32C(CRT를 통해서만), SHA1, SHA256	
SDK for Rust	예	CRC32	CRC64NVME, CRC32, CRC32C, SHA1, SHA256	
SDK for Swift	예	CRC32	CRC64NVME, CRC32, CRC32C, SHA1, SHA256	모든 알고리즘에 필요한 CRT 종속성입니다.
PowerShell V5용 도구	예	CRC32	CRC32, CRC32C, SHA1, SHA256	
PowerShell V4용 도구	예	CRC32	CRC32, CRC32C, SHA1, SHA256	

이중 스택 엔드포인트 및 FIPS 엔드포인트

> **Note**
>
> 설정 페이지의 레이아웃을 이해하거나 다음에 나오는 Support AWS SDKs이 가이드의 설정 페이지 이해.

다음을 사용하여 이 기능을 구성하십시오.

use_dualstack_endpoint - 공유 AWS **config** 파일 설정, **AWS_USE_DUALSTACK_ENDPOINT** - 환경 변수, **aws.useDualstackEndpoint** - JVM 시스템 속성: Java/Kotlin만 해당

SDK에서 듀얼 스택 엔드포인트에 요청을 보낼지 여부를 설정합니다. IPv4 및 IPv6 트래픽을 모두 지원하는 이중 스택 엔드포인트에 대한 자세한 내용은 Amazon 심플 스토리지 서비스 사용 설명서의 Amazon S3 이중 스택 엔드포인트 사용을 참조하십시오. 이중 스택 엔드포인트는 일부 리전에 사용할 수 있는 서비스입니다.

기본값: `false`

유효한 값:

- **true**— SDK 또는 도구는 이중 스택 엔드포인트를 사용하여 네트워크 요청을 시도합니다. 서비스 및/또는 AWS 리전에 대한 이중 스택 엔드포인트가 없는 경우 요청 오류가 생깁니다.
- **false**— SDK 또는 도구는 듀얼 스택 엔드포인트를 사용하여 네트워크 요청을 하지 않습니다.

use_fips_endpoint - 공유 AWS **config** 파일 설정, **AWS_USE_FIPS_ENDPOINT** - 환경 변수, **aws.useFipsEndpoint** - JVM 시스템 속성: Java/Kotlin만 해당

SDK 또는 도구로 FIPS 호환 엔드포인트로 요청을 보낼지 여부 활성 혹은 비활 성화 합니다. 연방 정보 처리 표준 (FIPS)은 데이터 및 암호화에 대한 미국 정부 보안 요구 사항의 집합입니다. 정부 기관, 파트너 및 연방 정부와 거래하려는 기관은 FIPS 지침을 준수해야 합니다. 표준 AWS 엔드포인트와 달리 FIPS 엔드포인트는 FIPS 140-2를 준수하는 TLS 소프트웨어 라이브러리를 사용합니다. 이 설정이 활성화되어 있고의 서비스에 대한 FIPS 엔드포인트가 없는 경우 AWS 호출 AWS 리전이 실패할 수 있습니다. 서비스별 엔드포인트 및 `--endpoint-url` 옵션은이 설정을 AWS Command Line Interface 재정의합니다.

FIPS 엔드포인트를 지정하는 다른 방법에 대한 자세한 내용은 서비스별 FIPS 엔드포인트를 AWS 리전참조하세요. https://aws.amazon.com/compliance/fips/ Amazon Elastic Compute Cloud 서비스 엔드포인트에 대한 자세한 내용은 Amazon EC2 API 참조의 이중 스택(IPv4 및 IPv6) 엔드포인트를 참조하십시오.

기본값: `false`

유효값:

- **`true`**— SDK 또는 도구는 FIPS 준수 엔드포인트에 요청을 보냅니다.
- **`false`**— SDK 또는 도구는 FIPS 호환 엔드포인트에 요청을 보내지 않습니다.

AWS SDKs 도구 지원

다음 SDK는 이 주제에서 설명하는 기능 및 설정을 지원합니다. 모든 일부 예외가 기록됩니다. 모든 JVM 시스템 속성 설정은 AWS SDK for Java 및 AWS SDK for Kotlin 에서만 지원됩니다.

SDK	지원	참고 또는 추가 정보
AWS CLI v2	예	
SDK for C++	예	
SDK for Go V2 (1.x)	예	
SDK for Go 1.x (V1)	예	공유 `config` 파일 설정을 사용하려면 구성 파일에서 로드를 켜야 합니다. 세션을 참조하십시오.
SDK for Java 2.x	예	
SDK for Java 1.x	아니요	
SDK for JavaScript 3.x	예	
SDK for JavaScript 2.x	예	
SDK for Kotlin	예	
.NET 4.x용 SDK	예	
SDK for .NET 3.x	예	
SDK for PHP 3.x	예	

SDK	지원	참고 또는 추가 정보
SDK for Python (Boto3)	예	
SDK for Ruby 3.x	예	
SDK for Rust	예	
SDK for Swift	예	
PowerShell V5용 도구	예	
PowerShell V4용 도구	예	

엔드포인트 검색

> **Note**
>
> 설정 페이지의 레이아웃을 이해하거나 다음에 나오는 Support AWS SDKs이 가이드의 설정 페이지 이해.

SDKs 엔드포인트 검색을 사용하여 서비스 엔드포인트(URLs을 사용하여 다양한 리소스에 액세스)에 액세스하는 동시에 필요에 따라가 URLs AWS 을 변경할 수 있는 유연성을 유지합니다. 이렇게 하면 코드가 새 엔드포인트를 자동으로 탐지할 수 있습니다. 일부 서비스에는 고정된 엔드포인트가 없습니다. 대신 런타임 중에 먼저 엔드포인트를 가져오기를 요청하여 사용 가능한 엔드포인트를 확보할 수 있습니다. 사용 가능한 엔드포인트를 검색한 후 코드는 해당 엔드포인트를 사용하여 다른 작업에 액세스합니다. 예를 들어 Amazon Timestream의 경우 SDK는 가용 엔드포인트를 검색하도록 `DescribeEndpoints` 요청하고 해당 엔드포인트를 사용하여 `CreateDatabase` 또는 `CreateTable` 같은 특정 작업을 완료합니다.

다음을 사용하여 이 기능을 구성하십시오.

`endpoint_discovery_enabled` - 공유 AWS **`config`** 파일 설정, **`AWS_ENABLE_ENDPOINT_DISCOVERY`** - 환경 변수, **`aws.endpointDiscoveryEnabled`** - JVM 시스템 속성: Java/Kotlin만 해당, 코드에서 값을 직접 구성하려면 특정 SDK를 직접 참조하십시오.

DynamoDB에 대한 엔드포인트 검색을 켜거나 끕니다.

엔드포인트 검색은 Timestream에서 필수이고 Amazon DynamoDB에서는 선택 사항입니다. 이 설정은 기본적으로 서비스에 엔드포인트 검색이 필요한지 여부에 `false` 따라 `true` 또는 로 설정됩니다. Timestream 요청은 기본적으로 `true`이고 Amazon DynamoDB 요청은 기본적으로 입니다`false`.

유효한 값:

- **`true`**— SDK는 엔드포인트 검색이 선택사항인 서비스의 엔드포인트 검색을 자동으로 시도해야 합니다.
- **`false`**— SDK는 엔드포인트 검색이 선택 사항인 서비스의 엔드포인트 검색을 자동으로 시도해서는 안 됩니다.

AWS SDKs 도구 지원

다음 SDK는 이 주제에서 설명하는 기능 및 설정을 지원합니다. 모든 일부 예외가 기록됩니다. 모든 JVM 시스템 속성 설정은 AWS SDK for Java 및 AWS SDK for Kotlin 에서만 지원됩니다.

SDK	지원	참고 또는 추가 정보
AWS CLI v2	예	
SDK for C++	예	
SDK for Go V2 (1.x)	예	
SDK for Go 1.x (V1)	예	공유 `config` 파일 설정을 사용하려면 구성 파일에서 로드를 켜야 합니다. 세션을 참조하십시오.
SDK for Java 2.x	예	SDK for Java 2.x는 환경 변수 이름`AWS_ENDPOINT_DISCOVERY_ENABLED` 에를 사용합니다.
SDK for Java 1.x	부분	JVM 시스템 속성은 지원되지 않습니다.
SDK for JavaScript 3.x	예	
SDK for JavaScript 2.x	예	
SDK for Kotlin	예	

SDK	지원	참고 또는 추가 정보
.NET 4.x용 SDK	예	
SDK for .NET 3.x	예	
SDK for PHP 3.x	예	
SDK for Python (Boto3)	예	
SDK for Ruby 3.x	예	
SDK for Rust	부분	타임스트림에만 지원됩니다.
SDK for Swift	아니요	
PowerShell V5용 도구	예	
PowerShell V4용 도구	예	

일반 구성 설정

> **Note**
>
> 설정 페이지의 레이아웃을 이해하거나 다음에 나오는 Support AWS SDKs이 가이드의 설정 페이지 이해.

SDK는 전체 SDK 동작을 구성하는 몇 가지 일반 설정을 지원합니다.

다음을 사용하여 이 기능을 구성하십시오.

`api_versions` - 공유 AWS **`config`** 파일 설정

일부 AWS 서비스는 이전 버전과의 호환성을 지원하기 위해 여러 API 버전을 유지 관리합니다. 기본적으로 SDK 및 AWS CLI 작업은 최신 API 버전을 사용합니다. 요청에 사용할 특정 API 버전을 요구하려면 프로파일에 `api_versions` 설정을 포함하십시오.

기본값: 없음. (SDK에는 최신 API 버전이 사용합니다.)

유효한 값: 중첩된 설정으로, 그 뒤에 사용할 AWS 서비스와 API 버전을 각각 식별하는 하나 이상의 들여쓰기된 줄이 옵니다. 사용 가능한 API 버전을 알아보려면 AWS 서비스 설명서를 참조하세요.

이 예제에서는 `config` 파일의 두 AWS 서비스에 대해 특정 API 버전을 설정합니다. 이러한 API 버전은 이러한 설정이 포함된 프로파일 하에서 실행되는 명령에서만 사용됩니다. 다른 서비스의 명령은 해당 서비스 API의 최신 버전을 사용합니다.

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

`ca_bundle` - 공유 AWS **`config`** 파일 설정, **`AWS_CA_BUNDLE`** - 환경 변수

SSL/TLS 연결을 설정할 때 사용할 사용자 지정 인증서 번들(`.pem` 확장명이 있는 파일)의 경로를 지정합니다.

기본값: 없음

유효한 값: 전체 경로 또는 기본 파일 이름을 지정합니다. 기본 파일 이름이 있는 경우, 시스템은 `PATH` 환경 변수로 지정된 폴더 내에서 프로그램을 찾으려고 시도합니다.

`config` 파일에서 이 값을 설정하는 예:

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

운영 체제가 경로를 처리하고 경로 문자를 이스케이프하는 방법이 다르기 때문에 Windows의 `config` 파일에서이 값을 설정하는 예는 다음과 같습니다.

```
[default]
ca_bundle = C:\\Users\\username\\.aws\\aws-custom-bundle.pem
```

명령 행을 통한 환경 변수 설정의 Linux/macOS 예제:

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

명령줄을 통해 환경 변수를 설정하는 Windows 예제:

```
setx AWS_CA_BUNDLE C:\dev\apps\ca-certs\cabundle-2019mar05.pem
```

output - 공유 AWS **config** 파일 설정

AWS CLI 및 기타 AWS SDKs 및 도구에서 결과의 형식을 지정하는 방법을 지정합니다.

기본값: `json`

유효값:

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

parameter_validation - 공유 AWS **config** 파일 설정

SDK 또는 도구가 AWS 서비스 엔드포인트에 보내기 전에 명령줄 파라미터를 검증할지 여부를 지정합니다.

기본값: `true`

유효값:

- **`true`** – 기본값입니다. SDK 또는 도구는 명령줄 파라미터를 클라이언트측에서 검증합니다. 이렇게 하면 SDK 또는 도구가 파라미터가 유효한지 확인하고 일부 오류를 포착하는 데 도움이 됩니다. SDK 또는 도구는 요청을 AWS 서비스 엔드포인트로 보내기 전에 유효하지 않은 요청을 거부할 수 있습니다.
- **`false`** - SDK 또는 도구는 AWS 서비스 엔드포인트로 전송하기 전에 명령줄 파라미터를 검증하지 않습니다. AWS 서비스 엔드포인트는 모든 요청을 검증하고 유효하지 않은 요청을 거부할 책임이 있습니다.

AWS SDKs 도구 지원

다음 SDK는 이 주제에서 설명하는 기능 및 설정을 지원합니다. 모든 일부 예외가 기록됩니다. 모든 JVM 시스템 속성 설정은 AWS SDK for Java 및 AWS SDK for Kotlin 에서만 지원됩니다.

SDK	지원	참고 또는 추가 정보
AWS CLI v2	부분	`api_versions` 이 지원되지 않음.
SDK for C++	예	
SDK for Go V2 (1.x)	부분	`api_versions` 및 `parameter_validation` 이 지원되지 않음.
SDK for Go 1.x (V1)	부분	`api_versions` 및 `parameter_validation` 이 지원되지 않음. 공유 `config` 파일 설정을 사용하려면 구성 파일에서 로드를 켜야 합니다. 세션을 참조하십시오.
SDK for Java 2.x	아니요	
SDK for Java 1.x	아니요	
SDK for JavaScript 3.x	예	
SDK for JavaScript 2.x	예	
SDK for Kotlin	아니요	
.NET 4.x용 SDK	아니요	

SDK	지원	참고 또는 추가 정보
SDK for .NET 3.x	아니요	
SDK for PHP 3.x	예	
SDK for Python (Boto3)	예	
SDK for Ruby 3.x	예	
SDK for Rust	아니요	
SDK for Swift	아니요	
PowerShell V5용 도구	아니요	
PowerShell V4용 도구	아니요	

호스트 접두사 삽입

> **Note**
>
> 설정 페이지의 레이아웃을 이해하거나 다음에 나오는 Support AWS SDKs이 가이드의 설정 페이지 이해.

호스트 접두사 삽입은 AWS SDKs가 특정 API 작업에 대한 서비스 엔드포인트의 호스트 이름 앞에 접두사를 자동으로 추가하는 기능입니다. 이 접두사는 정적 문자열이거나 요청 파라미터의 데이터를 포함하는 동적 값일 수 있습니다.

예를 들어 Amazon Simple Storage Service를 사용하여 Amazon S3 객체 또는 버킷에 대한 작업을 수행할 때 SDK는 최종 API 엔드포인트의 버킷 이름과 AWS 계정 ID를 대체합니다.

이 동작은 일반 AWS 서비스 엔드포인트에 필요하지만 VPC 엔드포인트 또는 로컬 테스트 도구와 같은 사용자 지정 엔드포인트를 사용할 때 문제가 발생할 수 있습니다. 이러한 경우 호스트 접두사 삽입을 비활성화해야 할 수 있습니다.

다음을 사용하여 이 기능을 구성하십시오.

`disable_host_prefix_injection` - 공유 AWS **`config`** 파일 설정, **`AWS_DISABLE_HOST_PREFIX_INJECTION`** - 환경 변수, **`aws.disableHostPrefixInjection`** - JVM 시스템 속성: Java/Kotlin만 해당

이 설정은 SDK 또는 도구가 SDK의 클라이언트 객체 또는 변수에 정의된 호스트 접두사 앞에 추가하여 엔드포인트 호스트 이름을 수정할지 여부를 제어합니다.

기본값: `false`

유효한 값:

- **`true`** - 호스트 접두사 삽입을 비활성화합니다. SDK는 엔드포인트 호스트 이름을 수정하지 않습니다.
- **`false`** - 호스트 접두사 삽입을 활성화합니다. SDK는 호스트 접두사를 엔드포인트 호스트 이름 앞에 추가합니다.

`config` 파일에서 이 값을 설정하는 예:

```
[default]
disable_host_prefix_injection = true
```

명령 행을 통한 환경 변수 설정의 Linux/macOS 예제:

```
export AWS_DISABLE_HOST_PREFIX_INJECTION=true
```

명령줄을 통해 환경 변수를 설정하는 Windows 예제:

```
setx AWS_DISABLE_HOST_PREFIX_INJECTION true
```

호스트 접두사 삽입의 예

다음 예제 표는 호스트 접두사 삽입이 활성화 및 비활성화될 때 SDKs가 최종 엔드포인트를 수정하는 방법을 보여줍니다.

- 호스트 접두사: SDK의 클라이언트 객체 또는 코드의 변수에 설정된 호스트 접두사 속성 문자열의 템플릿입니다.
- 입력: SDK의 클라이언트 객체 또는 코드의 변수에 설정된 추가 입력입니다.
- 클라이언트 엔드포인트: 클라이언트의 파생 엔드포인트입니다.
- 설정 값: 이전 설정의 해결된 값입니다.
- 결과 엔드포인트: SDK 클라이언트가 API 호출에 사용하는 결과 엔드포인트입니다.

호스트 접두사	입력	클라이언트 엔드포인트	값 설정	결과 엔드포인트
"data."	{}	"https://service.us-west-2.amazonaws.com"	false	"https://data.service.us-west-2.amazonaws.com"
"{Bucket}-{AccountId}."	Bucket: "amzn-s3-demo-bucket1", AccountId :"123456789012"	"https://service.us-west-2.amazonaws.com"	false	"https://amzn-s3-demo-bucket1-123456789012.service.us-west-2.amazonaws.com"
"data."	{}	"https://override.us-west-2.amazonaws.com" (as an override endpoint)	true	"https://override.us-west-2.amazonaws.com"

AWS SDKs 도구 지원

다음 SDK는 이 주제에서 설명하는 기능 및 설정을 지원합니다. 모든 일부 예외가 기록됩니다. 모든 JVM 시스템 속성 설정은 AWS SDK for Java 및 AWS SDK for Kotlin 에서만 지원됩니다.

SDK	지원	참고 또는 추가 정보
AWS CLI v2	예	
SDK for C++	아니요	설정은 지원되지 않지만를 사용하여 클라이언트의 코드에서 구성할 수 있습니다`enableHostPrefixInjection`.
SDK for Go V2 (1.x)	아니요	미들웨어를 사용하여 비활성화할 수 있습니다.
SDK for Go 1.x (V1)	아니요	
SDK for Java 2.x	아니요	설정은 지원되지 않지만를 사용하여 클라이언트의 코드에서 구성할 수 있습니다`SdkAdvancedClientOption.DISABLE_HOST_PREFIX_INJECTION`.
SDK for Java 1.x	아니요	설정은 지원되지 않지만를 사용하여 클라이언트의 코드에서 구성할 수 있습니다`withDisableHostPrefixInjection`.
SDK for JavaScript 3.x	아니요	설정은 지원되지 않지만를 사용하여 클라이언트의 코드에서 구성할 수 있습니다`disableHostPrefix`.
SDK for JavaScript 2.x	아니요	설정은 지원되지 않지만를 사용하여 클라이언트의 코드에서 구성할 수 있습니다`hostPrefixEnabled`.

SDK	지원	참고 또는 추가 정보
SDK for Kotlin	아니요	
.NET 4.x용 SDK	아니요	설정은 지원되지 않지만를 사용하여 클라이언트의 코드에서 구성할 수 있습니다`DisableHostPrefixInjection`.
SDK for .NET 3.x	아니요	설정은 지원되지 않지만를 사용하여 클라이언트의 코드에서 구성할 수 있습니다`DisableHostPrefixInjection`.
SDK for PHP 3.x	아니요	설정은 지원되지 않지만를 사용하여 클라이언트의 코드에서 구성할 수 있습니다`disable_host_prefix_injection`.
SDK for Python (Boto3)	예	를 사용하여 클라이언트의 코드로 구성할 수 있습니다`inject_host_prefix`.
SDK for Ruby 3.x	아니요	설정은 지원되지 않지만를 사용하여 클라이언트의 코드에서 구성할 수 있습니다`disable_host_prefix_injection`.
SDK for Rust	아니요	
SDK for Swift	아니요	
PowerShell V5용 도구	아니요	설정은 지원되지 않지만 파라미터를 사용하여 특정 cmdlet에 포함할 수 있습니다`-ClientConfig @{DisableHostPrefixInjection = $true}`.

SDK	지원	참고 또는 추가 정보
PowerShell V4용 도구	아니요	설정은 지원되지 않지만 파라미터를 사용하여 특정 cmdlet에 포함할 수 있습니다`-ClientConfig @{DisableHostPrefixInjection = $true}` .

IMDS 클라이언트

> **Note**
>
> 설정 페이지의 레이아웃을 이해하거나 다음에 나오는 Support AWS SDKs이 가이드의 설정 페이지 이해.

SDK는 세션 지향 요청을 사용하여 인스턴스 메타데이터 서비스 버전 2 (IMDSv2)클라이언트를 구현합니다. IMDSv2에 대한 자세한 내용은 Amazon EC2 사용 설명서의 IMDSv2 사용을 참조하세요. Amazon EC2 IMDS 클라이언트는 SDK 코드 베이스에서 사용할 수 있는 클라이언트 구성 객체를 통해 구성할 수 있습니다.

다음을 사용하여 이 기능을 구성하십시오.

`retries` - 클라이언트 구성 객체 멤버

실패한 요청에 대한 추가 재시도 횟수입니다.

기본값: 3

유효한 값: 0보다 큰 숫자.

`port` - 클라이언트 구성 객체 멤버

엔드포인트 포트.

기본값: 80

유효한 값: 숫자.

`token_ttl` - 클라이언트 구성 객체 멤버

토큰의 TTL.

기본값: 21,600초(6시간, 최대 할당 시간).

유효한 값: 숫자.

endpoint - 클라이언트 구성 객체 멤버

IMDS 엔드포인트.

기본값: endpoint_mode와 IPv4이 같으면 기본 엔드포인트는 http://169.254.169.254입니다. endpoint_mode와 IPv6이 같으면 기본 엔드포인트는 http://[fd00:ec2::254]입니다.

유효한 값: 유효한 URI.

대부분의 SDK에 의해 지원되는 옵션은 다음과 같습니다. 자세한 내용은 특정 SDK 코드베이스를 참조하십시오.

endpoint_mode- 클라이언트 구성 객체 멤버

IMDS의 엔드포인트 모드.

기본값: IPv4

유효한 값: IPv4, IPv6

http_open_timeout- 클라이언트 구성 객체 멤버 (이름은 다를 수 있음)

연결이 열릴 때까지 기다리는 시간 (초).

기본값: 1초.

유효한 값: 0보다 큰 숫자.

http_read_timeout- 클라이언트 구성 객체 멤버 (이름은 다를 수 있음)

데이터 청크 하나를 읽는 데 걸리는 시간 (초).

기본값: 1초.

유효한 값: 0보다 큰 숫자.

http_debug_output- 클라이언트 구성 객체 멤버 (이름은 다를 수 있음)

디버깅을 위한 출력 스트림을 설정합니다.

기본값: 없음.

유효한 값: STDOUT과 같은 유효한 I/O 스트림.

backoff- 클라이언트 구성 객체 멤버 (이름은 다를 수 있음)

재시도 또는 고객이 백오프 기능을 제공하여 전화를 걸 때까지 기다려야 하는 시간 (초). 이는 기본 지수 백오프 전략을 재정의 합니다.

기본값: SDK에 따라 다릅니다.

유효한 값: SDK에 따라 다릅니다. 숫자 값이거나 사용자 지정 함수 호출이 될 수 있습니다.

AWS SDKs 도구 지원

다음 SDK는 이 주제에서 설명하는 기능 및 설정을 지원합니다. 모든 일부 예외가 기록됩니다. 모든 JVM 시스템 속성 설정은 AWS SDK for Java 및 AWS SDK for Kotlin 에서만 지원됩니다.

SDK	지원	참고 또는 추가 정보
AWS CLI v2	예	
SDK for C++	아니요	
SDK for Go V2 (1.x)	예	
SDK for Go 1.x (V1)	예	
SDK for Java 2.x	예	
SDK for Java 1.x	예	
SDK for JavaScript 3.x	예	
SDK for JavaScript 2.x	예	
SDK for Kotlin	아니요	
.NET 4.x용 SDK	예	

SDK	지원	참고 또는 추가 정보
SDK for .NET 3.x	예	
SDK for PHP 3.x	예	
SDK for Python (Boto3)	예	
SDK for Ruby 3.x	예	
SDK for Rust	예	
SDK for Swift	예	
PowerShell V5용 도구	예	
PowerShell V4용 도구	예	

재시도 동작

> **Note**
>
> 설정 페이지의 레이아웃을 이해하거나 다음에 나오는 Support AWS SDKs이 가이드의 설정 페이지 이해.

재시도 동작에는 SDK가 AWS 서비스에 보내진 요청으로 인한 장애 복구를 시도하는 방법에 대한 설정이 포함됩니다.

다음을 사용하여 이 기능을 구성하십시오.

`retry_mode` - 공유 AWS **`config`** 파일 설정, **`AWS_RETRY_MODE`** - 환경 변수, **`aws.retryMode`** - JVM 시스템 속성: Java/Kotlin만 해당

SDK 또는 개발자 도구의 재시도 방법을 지정합니다.

기본값:이 값은 SDK에 따라 다릅니다. 특정 SDK 가이드 또는 SDK의 코드 베이스에서 기본를 확인합니다`retry_mode`.

유효한 값:

- `standard` – (권장) AWS SDKs. 이 모드에는 재시도되는 표준 오류 세트가 포함되며 가용성과 안정성을 극대화하기 위해 재시도 횟수를 자동으로 조정합니다. 이 모드는 다중 테넌트 애플리케이션에서 사용하기에 안전합니다. 명시적으로 `max_attempts`으로 구성되어 있지 않으면 이 모드의 기본 최대 시도 횟수는 3회입니다.
- `adaptive` - 표준 모드의 기능과 자동 클라이언트 측 속도 제한을 포함하는 특수 사용 사례에만 적합한 재시도 모드입니다. 이 재시도 모드는 애플리케이션 테넌트를 격리하지 않는 한 다중 테넌트 애플리케이션에는 권장되지 않습니다. 자세한 정보는 `standard` 및 `adaptive` 재시도 모드 선택을 참조하세요. 이 모드는 실험적이며 향후 동작이 변경될 수 있습니다.
- `legacy` – (권장되지 않음) SDK에 고유합니다(특정 SDK 가이드 또는 SDK의 코드 기반 확인).

`max_attempts` - 공유 AWS **`config`** 파일 설정, **`AWS_MAX_ATTEMPTS`** - 환경 변수, **`aws.maxAttempts`** - JVM 시스템 속성: Java/Kotlin만 해당

요청에 대한 최대 시도 횟수를 지정합니다.

기본값: 이 값이 지정되지 않은 경우 기본값은 `retry_mode` 설정 값에 따라 달라집니다.

- `retry_mode`이 `legacy` 인 경우— 사용자 SDK 고유의 기본값을 사용합니다(사용자 고유 SDK 설명서 또는 `max_attempts` 기본 SDK의 코드베이스를 확인하십시오).
- `retry_mode`이 `standard`인 경우 — 세 번 시도합니다.
- `retry_mode`이 `adaptive`인 경우 — 세 번 시도합니다.

유효한 값: 0보다 큰 숫자.

`standard` 및 `adaptive` 재시도 모드 선택

사용량이에 더 적합하다고 확신하지 않는 한 `standard` 재시도 모드를 사용하는 것이 좋습니다`adaptive`.

> **Note**
>
> 이 `adaptive` 모드에서는 백엔드 서비스가 요청을 제한할 수 있는 범위에 따라 클라이언트를 풀링한다고 가정합니다. 이렇게 하지 않으면 두 리소스에 동일한 클라이언트를 사용하는 경우 한 리소스의 제한으로 인해 관련 없는 리소스에 대한 요청이 지연될 수 있습니다.

표준	적응형
애플리케이션 사용 사례: 모두.	애플리케이션 사용 사례: 1. 지연 시간에 민감하지 않습니다. 2. 클라이언트는 단일 리소스에만 액세스하거나 액세스 중인 서비스 리소스별로 클라이언트를 별도로 풀링하는 로직을 제공합니다.
중단 시 SDK가 재시도되지 않도록 회로 차단을 지원합니다.	중단 시 SDK가 재시도되지 않도록 회로 차단을 지원합니다.
실패 시 지터링된 지수 백오프를 사용합니다.	동적 백오프 기간을 사용하여 지연 시간 증가 가능성에 대한 대가로 실패한 요청 수를 최소화하려고 시도합니다.
첫 번째 요청 시도를 지연하지 않고 재시도만 합니다.	초기 요청 시도를 제한하거나 지연시킬 수 있습니다.

`adaptive` 모드를 사용하기로 선택한 경우 애플리케이션은 제한될 수 있는 각 리소스를 중심으로 설계된 클라이언트를 구성해야 합니다. 이 경우 리소스는 각 리소스를 생각하는 것보다 더 세밀하게 조정됩니다 AWS 서비스. 에는 요청을 제한하는 데 사용하는 추가 차원이 있을 AWS 서비스 수 있습니다. Amazon DynamoDB 서비스를 예로 들어 보겠습니다. DynamoDB는 AWS 리전 및 액세스 중인 테이블을 사용하여 요청을 제한합니다. 즉, 코드가 액세스하는 테이블 하나가 다른 테이블보다 더 많이 제한될 수 있습니다. 코드가 동일한 클라이언트를 사용하여 모든 테이블에 액세스하고 해당 테이블 중 하나에 대한 요청이 제한되는 경우 적응형 재시도 모드는 모든 테이블에 대한 요청 속도를 줄입니다. 코드는 Region-and-table 페어당 하나의 클라이언트를 갖도록 설계되어야 합니다. `adaptive` 모드를 사용할 때 예기치 않은 지연 시간이 발생하는 경우 사용 중인 서비스에 대한 특정 AWS 설명서 가이드를 참조하세요.

재시도 모드 구현 세부 정보

AWS SDKs는 토큰 버킷을 사용하여 요청을 재시도할지 여부와 (`adaptive`재시도 모드의 경우) 요청을 얼마나 빨리 보내야 하는지 결정합니다. SDK는 재시도 토큰 버킷과 요청 속도 토큰 버킷이라는 두 개의 토큰 버킷을 사용합니다.

- 재시도 토큰 버킷은 중단 시 업스트림 및 다운스트림 서비스를 보호하기 위해 SDK가 재시도를 일시적으로 비활성화해야 하는지 여부를 결정하는 데 사용됩니다. 재시도를 시도하기 전에 버킷에서 토

큰을 획득하고 요청이 성공하면 토큰이 버킷으로 반환됩니다. 재시도를 시도할 때 버킷이 비어 있는 경우 SDK는 요청을 재시도하지 않습니다.

- 요청 속도 토큰 버킷은 요청을 보낼 속도를 결정하는 데 `adaptive` 재시도 모드에서만 사용됩니다. 토큰은 요청이 전송되기 전에 버킷에서 획득되며 서비스에서 반환되는 제한 응답에 따라 동적으로 결정된 속도로 버킷에 반환됩니다.

다음은 `standard` 모드와 `adaptive` 모드 모두에 대한 고급 의사코드입니다.

```
MakeSDKRequest() {
  attempts = 0
  loop {
    GetSendToken()
    response = SendHTTPRequest()
    RequestBookkeeping(response)
    if not Retryable(response)
      return response
    attempts += 1
    if attempts >= MAX_ATTEMPTS:
      return response
    if not HasRetryQuota(response)
      return response
    delay = ExponentialBackoff(attempts)
    sleep(delay)
  }
}
```

다음은 의사코드에 사용되는 구성 요소에 대한 자세한 내용입니다.

`GetSendToken`:

이 단계는 `adaptive` 재시도 모드에서만 사용됩니다. 이 단계에서는 요청 속도 토큰 버킷에서 토큰을 가져옵니다. 토큰을 사용할 수 없는 경우 토큰을 사용할 수 있을 때까지 기다립니다. SDK에 대기하는 대신 요청에 실패할 수 있는 구성 옵션이 있을 수 있습니다. 버킷의 토큰은 클라이언트가 수신한 제한 응답 수에 따라 동적으로 결정되는 속도로 다시 채워집니다.

`SendHTTPRequest`:

이 단계에서는 요청을 로 보냅니다 AWS. AWS SDKs HTTP 요청을 할 때 연결 풀을 사용하여 기존 연결을 재사용하는 HTTP 라이브러리를 사용합니다. 일반적으로 제한 오류로 인해 요청이 실패한 경우 연결이 재사용되지만 일시적인 오류로 인해 요청이 실패한 경우에는 재사용되지 않습니다.

RequestBookkeeping:

요청이 성공하면 토큰이 토큰 버킷에 추가됩니다. adaptive 재시도 모드의 경우에만 수신된 응답 유형에 따라 요청 속도 토큰 버킷의 채우기 속도가 업데이트됩니다.

Retryable:

이 단계에서는 다음을 기반으로 응답을 재시도할 수 있는지 여부를 결정합니다.

- HTTP 상태 코드 .
- 서비스에서 반환된 오류 코드입니다.
- 연결 오류로, 서비스로부터 HTTP 응답을 받지 못한 SDK에서 수신한 모든 오류로 정의됩니다.

일시적 오류(HTTP 상태 코드 400, 408, 500, 502, 503, 504)와 조절 오류(HTTP 상태 코드 400, 403, 429, 502, 503, 509)는 모두 재시도될 수 있습니다. SDK 재시도 동작은 서비스의 오류 코드 또는 서비스의 기타 데이터와 조합하여 결정됩니다.

MAX_ATTEMPTS:

retry_mode 설정에 의해 재정의되지 않는 한 기본 최대 시도 횟수는 max_attempts 설정에 의해 설정됩니다.

HasRetryQuota

이 단계에서는 재시도 토큰 버킷에서 토큰을 가져옵니다. 재시도 토큰 버킷이 비어 있으면 요청이 재시도되지 않습니다.

ExponentialBackoff

재시도할 수 있는 오류의 경우 잘린 지수 백오프를 사용하여 재시도 지연을 계산합니다. SDK는 지터가 있는 잘린 이진 지수 백오프를 사용합니다. 다음 알고리즘은 요청에 대한 응답의 절전 시간(초)이 어떻게 정의되는지 보여줍니다. i

```
seconds_to_sleep_i = min(b*r^i, MAX_BACKOFF)
```

위 알고리즘에는 다음 값이 적용됩니다.

b = random number within the range of: 0 <= b <= 1

r = 2

대부분의 SDK용 `MAX_BACKOFF = 20 seconds`. 사용자 특정 SDK 가이드 또는 소스 코드를 참조하여 확인합니다.

AWS SDKs 도구 지원

다음 SDK는 이 주제에서 설명하는 기능 및 설정을 지원합니다. 모든 일부 예외가 기록됩니다. 모든 JVM 시스템 속성 설정은 AWS SDK for Java 및 AWS SDK for Kotlin 에서만 지원됩니다.

SDK	지원	참고 또는 추가 정보
AWS CLI v2	예	
SDK for C++	예	
SDK for Go V2 (1.x)	예	
SDK for Go 1.x (V1)	아니요	
SDK for Java 2.x	예	
SDK for Java 1.x	예	JVM 시스템 속성: `com.amazonaws.sdk.maxAttempts` 대신를 사용하고 `com.amazonaws.sdk.retryMode` 대신를 `aws.maxAttempts` 사용합니다`aws.retryMode` .
SDK for JavaScript 3.x	예	
SDK for JavaScript 2.x	아니요	최대 재시도 횟수, 지터가 있는 지수 백오프, 재시도 백오프를 위한 사용자 지정 방법 옵션을 지원합니다.
SDK for Kotlin	예	
.NET 4.x용 SDK	예	
SDK for .NET 3.x	예	
SDK for PHP 3.x	예	

SDK	지원	참고 또는 추가 정보
SDK for Python (Boto3)	예	
SDK for Ruby 3.x	예	
SDK for Rust	예	
SDK for Swift	예	
PowerShell V5용 도구	예	
PowerShell V4용 도구	예	

요청 압축

> **Note**
>
> 설정 페이지의 레이아웃을 이해하거나 다음에 나오는 Support AWS SDKs이 가이드의 설정 페이지 이해.

AWS SDKs페이로드 수신을 지원하는 AWS 서비스 에 요청을 보낼 때 페이로드를 자동으로 압축할 수 있습니다. 페이로드를 서비스로 보내기 전에 클라이언트에서 압축하면 데이터를 서비스로 보내는 데 필요한 전체 요청 수와 대역폭을 줄일 수 있을 뿐만 아니라 페이로드 크기에 대한 서비스 제한으로 인해 실패한 요청도 줄일 수 있습니다. 압축을 위해 SDK 또는 도구는 서비스와 SDK에서 모두 지원하는 인코딩 알고리즘을 선택합니다. 그러나 현재 가능한 인코딩 목록은 gzip으로만 구성되어 있지만 향후 확장될 수 있습니다.

요청 압축은 애플리케이션에서 Amazon CloudWatch를 사용하는 경우 특히 유용할 수 있습니다. CloudWatch는 모니터링 및 운영 데이터를 로그, 지표 및 이벤트 형태로 수집하는 모니터링 및 관찰성 서비스입니다. 압축을 지원하는 서비스 작업의 한 예로 CloudWatch의 PutMetricDataAPI 방법을 들 수 있습니다.

다음을 사용하여 이 기능을 구성하십시오.

disable_request_compression - 공유 AWS **config** 파일 설정, **AWS_DISABLE_REQUEST_COMPRESSION** - 환경 변수, **aws.disableRequestCompression** - JVM 시스템 속성: Java/Kotlin만 해당

요청을 보내기 전에 SDK 또는 도구가 페이로드를 압축할지 여부를 설정하거나 해제합니다.

기본값: `false`

유효값:

- **true** - 요청 압축을 해제합니다.
- **false** - 가능하면 요청 압축을 사용합니다.

request_min_compression_size_bytes - 공유 AWS **config** 파일 설정, **AWS_REQUEST_MIN_COMPRESSION_SIZE_BYTES** - 환경 변수, **aws.requestMinCompressionSizeBytes** - JVM 시스템 속성: Java/Kotlin만 해당

SDK 또는 도구가 압축해야 하는 요청 본문의 최소 크기(바이트)를 설정합니다. 작은 페이로드는 압축 시 더 길어질 수 있으므로 압축을 수행하는 데 적합한 하한선이 있습니다. 이 값은 포함되며, 이 값보다 크거나 같은 요청 크기는 압축됩니다.

기본값: 10240바이트

유효값: 0~10485760바이트 사이의 정수 값.

AWS SDKs 도구 지원

다음 SDK는 이 주제에서 설명하는 기능 및 설정을 지원합니다. 모든 일부 예외가 기록됩니다. 모든 JVM 시스템 속성 설정은 AWS SDK for Java 및 AWS SDK for Kotlin 에서만 지원됩니다.

SDK	지원	참고 또는 추가 정보
AWS CLI v2	예	
SDK for C++	예	
SDK for Go V2 (1.x)	예	

SDK	지원	참고 또는 추가 정보
SDK for Go 1.x (V1)	아니요	
SDK for Java 2.x	예	
SDK for Java 1.x	아니요	
SDK for JavaScript 3.x	예	
SDK for JavaScript 2.x	아니요	
SDK for Kotlin	예	
.NET 4.x용 SDK	예	
SDK for .NET 3.x	예	
SDK for PHP 3.x	예	
SDK for Python (Boto3)	예	
SDK for Ruby 3.x	예	
SDK for Rust	예	
SDK for Swift	아니요	
PowerShell V5용 도구	예	
PowerShell V4용 도구	예	

서비스별 엔드포인트

> **Note**
>
> 설정 페이지의 레이아웃을 이해하거나 다음에 나오는 Support AWS SDKs이 가이드의 설정 페이지 이해.

서비스별 엔드포인트 구성은 API 요청에 대해 사용자가 선택한 엔드포인트를 사용하고 선택을 유지할 수 있는 옵션을 제공합니다. 이러한 설정은 로컬 엔드포인트, VPC 엔드포인트 및 타사 로컬 AWS 개발 환경을 지원할 수 있는 유연성을 제공합니다. 테스트 환경과 프로덕션 환경에 서로 다른 엔드포인트를 사용할 수 있습니다. 개별 AWS 서비스서비스에 대한 엔드포인트 URL을 지정할 수 있습니다.

다음을 사용하여 이 기능을 구성하십시오.

`endpoint_url` - 공유 AWS **`config`** 파일 설정, **`AWS_ENDPOINT_URL`** - 환경 변수, **`aws.endpointUrl`** - JVM 시스템 속성: Java/Kotlin만 해당

프로파일 내에서 직접 지정하거나 환경 변수로 지정하는 경우 이 설정은 모든 서비스 요청에 사용되는 엔드포인트를 지정합니다. 이 엔드포인트는 구성된 모든 서비스별 엔드포인트에 의해 재정의됩니다.

공유 AWS `config` 파일의 `services` 섹션 내에서이 설정을 사용하여 특정 서비스에 대한 사용자 지정 엔드포인트를 설정할 수도 있습니다. `services` 섹션에서 사용할 모든 서비스 식별자 키 목록은 서비스별 엔드포인트 식별자 섹션을 참조하십시오.

기본값: none

유효한 값: 엔드포인트의 스키마와 호스트가 포함된 URL. URL에는 하나 이상의 경로 세그먼트가 포함된 경로 구성 요소가 선택적으로 포함될 수 있습니다.

`AWS_ENDPOINT_URL_<SERVICE>` - 환경 변수, **`aws.endpointUrl<ServiceName>`** - JVM 시스템 속성: Java/Kotlin만 해당

`AWS_ENDPOINT_URL_<SERVICE>`, 여기서 `<SERVICE>`는 AWS 서비스 식별자이며,는 특정 서비스에 대한 사용자 지정 엔드포인트를 설정합니다. 모든 서비스별 환경 변수 목록은 서비스별 엔드포인트 식별자을 참조하십시오.

이 서비스별 엔드포인트는 `AWS_ENDPOINT_URL`에 설정된 모든 글로벌 엔드포인트보다 우선합니다.

기본값: none

유효한 값: 엔드포인트의 스키마와 호스트가 포함된 URL. URL에는 하나 이상의 경로 세그먼트가 포함된 경로 구성 요소가 선택적으로 포함될 수 있습니다.

ignore_configured_endpoint_urls - 공유 AWS **config** 파일 설정, **AWS_IGNORE_CONFIGURED_ENDPOINT_URLS** - 환경 변수, **aws.ignoreConfiguredEndpointUrls** - JVM 시스템 속성: Java/Kotlin만 해당

이 설정은 모든 사용자 지정 엔드포인트 구성을 무시하는 데 사용됩니다.

코드에 설정되거나 서비스 클라이언트 자체에 설정된 명시적 엔드포인트는 이 설정과 상관없이 사용된다는 점에 유의하십시오. 예를 들어 `--endpoint-url` 명령줄 파라미터를 AWS CLI 명령과 함께 포함하거나 엔드포인트 URL을 클라이언트 생성자에 전달하면 항상 적용됩니다.

기본값: `false`

유효값:

- **true**— SDK 또는 도구는 공유 `config` 파일 또는 환경 변수에서 엔드포인트 URL 설정을 위한 사용자 지정 구성 옵션을 읽지 않습니다.
- **false**— SDK 또는 도구는 공유 `config` 파일 또는 환경 변수에서 사용 가능한 사용자 제공 엔드포인트를 사용합니다.

환경 변수를 사용한 엔드포인트 구성

모든 서비스에 대한 요청을 사용자 지정 엔드포인트 URL로 라우팅하려면 `AWS_ENDPOINT_URL` 글로벌 환경 변수를 설정하십시오.

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

특정에 대한 요청을 사용자 지정 엔드포인트 URL AWS 서비스 로 라우팅하려면 `AWS_ENDPOINT_URL_<SERVICE>` 환경 변수를 사용합니다. Amazon DynamoDB 에는 `serviceId`의이 있습니다DynamoDB. 이 서비스의 경우 엔드포인트 URL 환경 변수는 `AWS_ENDPOINT_URL_DYNAMODB`입니다. 이 엔드포인트는 이서비스에 대해 `AWS_ENDPOINT_URL`에 설정된 글로벌 엔드포인트보다 우선합니다.

```
export AWS_ENDPOINT_URL_DYNAMODB=http://localhost:5678
```

또 다른 예로의 `serviceId` AWS Elastic Beanstalk 가 있습니다Elastic Beanstalk. AWS 서비스 식별자는 모든 공백을 밑줄로 바꾸고 모든 문자를 대문자로 대체`serviceId`하여 API

모델의를 기반으로 합니다. 이 서비스에 대한 엔드포인트를 설정하려면 해당 환경 변수는 AWS_ENDPOINT_URL_ELASTIC_BEANSTALK입니다. 모든 서비스별 환경 변수 목록은 서비스별 엔드포인트 식별자을 참조하십시오.

```
export AWS_ENDPOINT_URL_ELASTIC_BEANSTALK=http://localhost:5567
```

공유 **config** 파일을 사용하여 엔드포인트를 구성합니다.

공유 config 파일에서는 다양한 기능을 위해 여러 위치에서 endpoint_url이 사용됩니다.

- profile 내에서 직접 지정된 endpoint_url은 해당 엔드포인트를 글로벌 엔드포인트로 만듭니다.
- services 섹션 내의 서비스 식별자 키 아래에 중첩된 endpoint_url은 해당 엔드포인트가 해당 서비스에 대한 요청에만 적용되로고 만듭니다. 공유 config 파일에서 services 섹션을 정의하는 방법에 대한 자세한 내용은 구성 파일 형식를 참조하십시오.

다음 예제에서는 services 정의를 사용하여 Amazon S3에 대한 서비스별 엔드포인트 URL과 다른 모든 서비스에 사용되는 사용자 지정 글로벌 엔드포인트를 구성합니다.

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

[services s3-specific]
s3 =
  endpoint_url = https://play.min.io:9000
```

단일 프로파일로 여러 서비스에 대한 엔드포인트를 구성할 수 있습니다. 이 예제에서는 동일한 프로파일에서 Amazon S3 및 AWS Elastic Beanstalk 에 대한 서비스별 엔드포인트 URLs을 설정하는 방법을 보여줍니다. serviceId의 AWS Elastic Beanstalk 가 있습니다Elastic Beanstalk. AWS 서비스 식별자는 모든 공백을 밑줄로 바꾸고 모든 문자를 소문자로 대체serviceId하여 API 모델의를 기반으로 합니다. 따라서 서비스 식별자 키가 elastic_beanstalk이(가)되고 이 서비스에 대한 설정이 elastic_beanstalk = 줄에서 시작됩니다. services 섹션에서 사용할 모든 서비스 식별자 키 목록은 서비스별 엔드포인트 식별자을 참조하십시오.

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

```
  endpoint_url = http://localhost:8000

[profile dev]
services = testing-s3-and-eb
```

서비스 구성 섹션은 여러 프로파일에서 사용할 수 있습니다. 예를 들어 두 프로파일이 동일한 services 정의를 사용하면서 다른 프로파일 속성을 변경할 수 있습니다.

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

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

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

역할 기반 보안 인증을 사용하여 프로파일의 엔드포인트 구성

프로파일에 역할 기반 보안 인증 정보가 IAM 가정 역할 기능에 대한 source_profile 파라미터를 통해 구성된 경우 SDK는 지정된 프로파일에 대한 서비스 구성만 사용합니다. 역할이 연결된 프로파일은 사용하지 않습니다. 예를 들어 다음과 같은 공유 config 파일을 사용합니다.

```
[profile A]
credential_source = Ec2InstanceMetadata
endpoint_url = https://profile-a-endpoint.aws/

[profile B]
source_profile = A
role_arn = arn:aws:iam::123456789012:role/roleB
services = profileB

[services profileB]
ec2 =
  endpoint_url = https://profile-b-ec2-endpoint.aws
```

프로파일 B를 사용하고 코드에서 Amazon EC2로 호출하는 경우 엔드포인트는 https://profile-b-ec2-endpoint.aws로 확인됩니다. 코드에서 다른 서비스에 요청을 하는 경우 엔드포인트 확

인은 사용자 지정 로직을 따르지 않습니다. 엔드포인트는 프로파일 A에 정의된 글로벌 엔드포인트로 확인되지 않습니다. 글로벌 엔드포인트가 프로파일 B에 적용되려면 프로파일 B 내에서 직접 `endpoint_url`을 설정해야 합니다. `source_profile` 설정에 대한 자세한 내용은 역할 보안 인증 제공자 수임 단원을 참조하십시오.

설정의 우선 순위

이 기능의 설정은 동시에 사용할 수 있지만 서비스당 하나의 값만 우선합니다. 지정된에 대한 API 호출 AWS 서비스의 경우 다음 순서가 값을 선택하는 데 사용됩니다.

1. 코드나 서비스 클라이언트 자체에 설정된 모든 명시적 설정은 다른 모든 설정보다 우선합니다.
   - 의 경우 `--endpoint-url` 명령줄 파라미터에서 제공하는 값 AWS CLI입니다. SDK의 경우 명시적 할당은 AWS 서비스 클라이언트 또는 구성 객체를 인스턴스화할 때 설정한 파라미터의 형태를 취할 수 있습니다.
2. 서비스별 환경 변수에서 제공하는 값(예: `AWS_ENDPOINT_URL_DYNAMODB`).
3. `AWS_ENDPOINT_URL` 글로벌 엔드포인트 환경 변수에 의해 제공되는 값입니다.
4. 공유 `config` 파일의 `services` 섹션 내에 서비스 식별자 키 아래 중첩된 `endpoint_url` 설정에서 제공하는 값.
5. 공유 `config` 파일의 `profile` 내에서 직접 지정된 `endpoint_url` 설정에 의해 제공되는 값.
6. 각의 기본 엔드포인트 URL AWS 서비스 이 마지막으로 사용됩니다.

AWS SDKs 도구 지원

다음 SDK는 이 주제에서 설명하는 기능 및 설정을 지원합니다. 모든 일부 예외가 기록됩니다. 모든 JVM 시스템 속성 설정은 AWS SDK for Java 및 AWS SDK for Kotlin 에서만 지원됩니다.

SDK	지원	참고 또는 추가 정보
AWS CLI v2	예	
SDK for C++	아니요	
SDK for Go V2 (1.x)	예	

SDK	지원	참고 또는 추가 정보
SDK for Go 1.x (V1)	아니요	
SDK for Java 2.x	예	
SDK for Java 1.x	아니요	
SDK for JavaScript 3.x	예	
SDK for JavaScript 2.x	아니요	
SDK for Kotlin	예	
.NET 4.x용 SDK	예	
SDK for .NET 3.x	예	
SDK for PHP 3.x	예	
SDK for Python (Boto3)	예	
SDK for Ruby 3.x	예	
SDK for Rust	예	
SDK for Swift	예	
PowerShell V5용 도구	예	
PowerShell V4용 도구	예	

서비스별 엔드포인트 식별자

다음 표의 식별자를 사용하는 방법 및 위치에 대한 자세한 내용은 서비스별 엔드포인트 섹션을 참조하십시오.

serviceId	공 유 A **c** 파 일 으 서 비 스 스 별 자 키	**AWS_ENDPOINT_URL_<SERVICE>** 환경 변수
AccessAnalyzer	a l	AWS_ENDPOINT_URL_ACCESSANALYZER
Account	a	AWS_ENDPOINT_URL_ACCOUNT
ACM	a	AWS_ENDPOINT_URL_ACM
ACM PCA	a	AWS_ENDPOINT_URL_ACM_PCA
Alexa For Business	a _	AWS_ENDPOINT_URL_ALEXA_FOR_BUSINESS
amp	a	AWS_ENDPOINT_URL_AMP
Amplify	a	AWS_ENDPOINT_URL_AMPLIFY
AmplifyBackend	a c	AWS_ENDPOINT_URL_AMPLIFYBACKEND

serviceId	공 유 A **c** 파 일 으 사 비 스 스 별 자 키	**AWS_ENDPOINT_URL_<SERVICE>** 환경 변수
AmplifyUIBuilder	ar bu	AWS_ENDPOINT_URL_AMPLIFYUIBUILDER
API Gateway	ap ay	AWS_ENDPOINT_URL_API_GATEWAY
ApiGatewayManagementApi	ap yr nt	AWS_ENDPOINT_URL_APIGATEWAYMANAGEMENTAPI
ApiGatewayV2	ap yv	AWS_ENDPOINT_URL_APIGATEWAYV2
AppConfig	ap	AWS_ENDPOINT_URL_APPCONFIG
AppConfigData	ap da	AWS_ENDPOINT_URL_APPCONFIGDATA
AppFabric	ap	AWS_ENDPOINT_URL_APPFABRIC
Appflow	ap	AWS_ENDPOINT_URL_APPFLOW

serviceId	공 유 A **c** 파 일 으 사 비 스 스 별 자 키	**AWS_ENDPOINT_URL_<SERVICE>** 환경 변수
AppIntegrations	a a	AWS_ENDPOINT_URL_APPINTEGRATIONS
Application Auto Scaling	a o c	AWS_ENDPOINT_URL_APPLICATION_AUTO_SCALING
Application Insights	a o t	AWS_ENDPOINT_URL_APPLICATION_INSIGHTS
ApplicationCostProfiler	a o f	AWS_ENDPOINT_URL_APPLICATIONCOSTPROFILER
App Mesh	a	AWS_ENDPOINT_URL_APP_MESH
AppRunner	a	AWS_ENDPOINT_URL_APPRUNNER
AppStream	a	AWS_ENDPOINT_URL_APPSTREAM

serviceId	공 유 A **c** 파 일 으 사 비 스 스 별 자 키	**AWS_ENDPOINT_URL_<SERVICE>** 환경 변수
AppSync	ap	AWS_ENDPOINT_URL_APPSYNC
ARC Zonal Shift	ar _s	AWS_ENDPOINT_URL_ARC_ZONAL_SHIFT
Artifact	ar	AWS_ENDPOINT_URL_ARTIFACT
Athena	at	AWS_ENDPOINT_URL_ATHENA
AuditManager	au ge	AWS_ENDPOINT_URL_AUDITMANAGER
Auto Scaling	au in	AWS_ENDPOINT_URL_AUTO_SCALING
Auto Scaling Plans	au in	AWS_ENDPOINT_URL_AUTO_SCALING_PLANS
b2bi	b2	AWS_ENDPOINT_URL_B2BI
Backup	ba	AWS_ENDPOINT_URL_BACKUP

serviceId	공 유 AV **co** 파 일 으 서 비 스 스 별 자 키	**AWS_ENDPOINT_URL_<SERVICE>** 환경 변수
Backup Gateway	ba te	AWS_ENDPOINT_URL_BACKUP_GATEWAY
BackupStorage	ba ra	AWS_ENDPOINT_URL_BACKUPSTORAGE
Batch	ba	AWS_ENDPOINT_URL_BATCH
BCM Data Exports	bc ex	AWS_ENDPOINT_URL_BCM_DATA_EXPORTS
Bedrock	be	AWS_ENDPOINT_URL_BEDROCK
Bedrock Agent	be ge	AWS_ENDPOINT_URL_BEDROCK_AGENT
Bedrock Agent Runtime	be ge in	AWS_ENDPOINT_URL_BEDROCK_AGENT_RUNTIME
Bedrock Runtime	be un	AWS_ENDPOINT_URL_BEDROCK_RUNTIME

serviceId	공유 AWS **co** 파일의 서비스별 자 키	**AWS_ENDPOINT_URL_<SERVICE>** 환경 변수
billingconductor	bi no	AWS_ENDPOINT_URL_BILLINGCONDUCTOR
Braket	br	AWS_ENDPOINT_URL_BRAKET
Budgets	bu	AWS_ENDPOINT_URL_BUDGETS
Cost Explorer	ce ox	AWS_ENDPOINT_URL_COST_EXPLORER
chatbot	ch	AWS_ENDPOINT_URL_CHATBOT
Chime	ch	AWS_ENDPOINT_URL_CHIME
Chime SDK Identity	ch _i	AWS_ENDPOINT_URL_CHIME_SDK_IDENTITY
Chime SDK Media Pipelines	ch _m pe	AWS_ENDPOINT_URL_CHIME_SDK_MEDIA_PIPELINES

serviceId	공 유 A **c** 파 일 으 사 비 스 스 별 자 키	**AWS_ENDPOINT_URL_<SERVICE>** 환경 변수
Chime SDK Meetings	ch _r	AWS_ENDPOINT_URL_CHIME_SDK_MEETINGS
Chime SDK Messaging	ch _r g	AWS_ENDPOINT_URL_CHIME_SDK_MESSAGING
Chime SDK Voice	ch _v	AWS_ENDPOINT_URL_CHIME_SDK_VOICE
CleanRooms	cl s	AWS_ENDPOINT_URL_CLEANROOMS
CleanRoomsML	cl sr	AWS_ENDPOINT_URL_CLEANROOMSML
Cloud9	cl	AWS_ENDPOINT_URL_CLOUD9
CloudControl	cl rc	AWS_ENDPOINT_URL_CLOUDCONTROL

serviceId	공 유 A **c** 파 일 으 서 비 스 스 별 자 키	**AWS_ENDPOINT_URL_<SERVICE>** 환경 변수
CloudDirectory	c c	AWS_ENDPOINT_URL_CLOUDDIRECTORY
CloudFormation	c a	AWS_ENDPOINT_URL_CLOUDFORMATION
CloudFront	c t	AWS_ENDPOINT_URL_CLOUDFRONT
CloudFront KeyValueStore	c t_ e	AWS_ENDPOINT_URL_CLOUDFRONT_KEYVALUESTORE
CloudHSM	c	AWS_ENDPOINT_URL_CLOUDHSM
CloudHSM V2	c v	AWS_ENDPOINT_URL_CLOUDHSM_V2
CloudSearch	c c	AWS_ENDPOINT_URL_CLOUDSEARCH

serviceId	공 유 A **c** 파 일 으 서 비 스 스 별 자 키	**AWS_ENDPOINT_URL_<SERVICE>** 환경 변수
CloudSearch Domain	c ch	AWS_ENDPOINT_URL_CLOUDSEARCH_DOMAIN
CloudTrail	c l	AWS_ENDPOINT_URL_CLOUDTRAIL
CloudTrail Data	c l_	AWS_ENDPOINT_URL_CLOUDTRAIL_DATA
CloudWatch	c h	AWS_ENDPOINT_URL_CLOUDWATCH
codeartifact	c a	AWS_ENDPOINT_URL_CODEARTIFACT
CodeBuild	c	AWS_ENDPOINT_URL_CODEBUILD
CodeCatalyst	c y	AWS_ENDPOINT_URL_CODECATALYST

serviceId	공유 AWS **co** 파일 으 서 비 스 스 별 자 키	**AWS_ENDPOINT_URL_<SERVICE>** 환경 변수
CodeCommit	co t	AWS_ENDPOINT_URL_CODECOMMIT
CodeDeploy	co y	AWS_ENDPOINT_URL_CODEDEPLOY
CodeGuru Reviewer	co re	AWS_ENDPOINT_URL_CODEGURU_REVIEWER
CodeGuru Security	co se	AWS_ENDPOINT_URL_CODEGURU_SECURITY
CodeGuruProfiler	co ro	AWS_ENDPOINT_URL_CODEGURUPROFILER
CodePipeline	co in	AWS_ENDPOINT_URL_CODEPIPELINE
CodeStar	co	AWS_ENDPOINT_URL_CODESTAR
CodeStar connections	co co ns	AWS_ENDPOINT_URL_CODESTAR_CONNECTIONS

serviceId	공유 AWS **co** 파일으 서비스 별 자키	**AWS_ENDPOINT_URL_<SERVICE>** 환경 변수
codestar notifications	co no io	AWS_ENDPOINT_URL_CODESTAR_NOTIFICATIONS
Cognito Identity	co de	AWS_ENDPOINT_URL_COGNITO_IDENTITY
Cognito Identity Provider	co de ro	AWS_ENDPOINT_URL_COGNITO_IDENTITY_PROVIDER
Cognito Sync	co yn	AWS_ENDPOINT_URL_COGNITO_SYNC
Comprehend	co d	AWS_ENDPOINT_URL_COMPREHEND
ComprehendMedical	co dr	AWS_ENDPOINT_URL_COMPREHENDMEDICAL
Compute Optimizer	co pt	AWS_ENDPOINT_URL_COMPUTE_OPTIMIZER

serviceId	공 유 A **c** 파 일 으 사 비 스 스 별 자 키	**AWS_ENDPOINT_URL_<SERVICE>** 환경 변수
Config Service	c r	AWS_ENDPOINT_URL_CONFIG_SERVICE
Connect	c	AWS_ENDPOINT_URL_CONNECT
Connect Contact Lens	c o n	AWS_ENDPOINT_URL_CONNECT_CONTACT_LENS
ConnectCampaigns	c m	AWS_ENDPOINT_URL_CONNECTCAMPAIGNS
ConnectCases	c s	AWS_ENDPOINT_URL_CONNECTCASES
ConnectParticipant	c r	AWS_ENDPOINT_URL_CONNECTPARTICIPANT
ControlTower	c w	AWS_ENDPOINT_URL_CONTROLTOWER

serviceId	공유 AWS **co** 파일으 서비스 별 자키	**AWS_ENDPOINT_URL_<SERVICE>** 환경 변수
Cost Optimization Hub	co m: hu	AWS_ENDPOINT_URL_COST_OPTIMIZATION_HUB
Cost and Usage Report Service	co us o: ce	AWS_ENDPOINT_URL_COST_AND_USAGE_REPO RT_SERVICE
Customer Profiles	cu p:	AWS_ENDPOINT_URL_CUSTOMER_PROFILES
DataBrew	da	AWS_ENDPOINT_URL_DATABREW
DataExchange	da ng	AWS_ENDPOINT_URL_DATAEXCHANGE
Data Pipeline	da l:	AWS_ENDPOINT_URL_DATA_PIPELINE
DataSync	da	AWS_ENDPOINT_URL_DATASYNC
DataZone	da	AWS_ENDPOINT_URL_DATAZONE

serviceId	공 유 A **c** 파 일 으 사 비 스 스 별 자 키	**AWS_ENDPOINT_URL_<SERVICE>** 환경 변수
DAX	da	AWS_ENDPOINT_URL_DAX
Detective	de	AWS_ENDPOINT_URL_DETECTIVE
Device Farm	de rn	AWS_ENDPOINT_URL_DEVICE_FARM
DevOps Guru	de ru	AWS_ENDPOINT_URL_DEVOPS_GURU
Direct Connect	di nn	AWS_ENDPOINT_URL_DIRECT_CONNECT
Application Discovery Service	ap or e: ce	AWS_ENDPOINT_URL_APPLICATION_DISCOVE RY_SERVICE
DLM	dl	AWS_ENDPOINT_URL_DLM

serviceId	공유 AWS **co** 파일으 서비스 별 자 키	**AWS_ENDPOINT_URL_<SERVICE>** 환경 변수
Database Migration Service	da m: _S	AWS_ENDPOINT_URL_DATABASE_MIGRATION_ SERVICE
DocDB	do	AWS_ENDPOINT_URL_DOCDB
DocDB Elastic	do st	AWS_ENDPOINT_URL_DOCDB_ELASTIC
drs	dr	AWS_ENDPOINT_URL_DRS
Directory Service	di _S	AWS_ENDPOINT_URL_DIRECTORY_SERVICE
DynamoDB	dy	AWS_ENDPOINT_URL_DYNAMODB
DynamoDB Streams	dy st	AWS_ENDPOINT_URL_DYNAMODB_STREAMS
EBS	eb	AWS_ENDPOINT_URL_EBS
EC2	ec	AWS_ENDPOINT_URL_EC2

serviceId	공유 AWS **co** 파일 으 서 비 스 스 별 자 키	**AWS_ENDPOINT_URL_<SERVICE>** 환경 변수
EC2 Instance Connect	ec nc ct	AWS_ENDPOINT_URL_EC2_INSTANCE_CONNECT
ECR	ec	AWS_ENDPOINT_URL_ECR
ECR PUBLIC	ec c	AWS_ENDPOINT_URL_ECR_PUBLIC
ECS	ec	AWS_ENDPOINT_URL_ECS
EFS	ef	AWS_ENDPOINT_URL_EFS
EKS	el	AWS_ENDPOINT_URL_EKS
EKS Auth	el	AWS_ENDPOINT_URL_EKS_AUTH
Elastic Inference	el nf	AWS_ENDPOINT_URL_ELASTIC_INFERENCE
ElastiCache	el he	AWS_ENDPOINT_URL_ELASTICACHE

serviceId	공 유 A **c** 파 일 으 서 비 스 스 별 자 키	**AWS_ENDPOINT_URL_<SERVICE>** 환경 변수
Elastic Beanstalk	e ea	AWS_ENDPOINT_URL_ELASTIC_BEANSTALK
Elastic Transcoder	e ra	AWS_ENDPOINT_URL_ELASTIC_TRANSCODER
Elastic Load Balancing	e oa c	AWS_ENDPOINT_URL_ELASTIC_LOAD_BALANCING
Elastic Load Balancing v2	e oa c	AWS_ENDPOINT_URL_ELASTIC_LOAD_BALANCING_V2
EMR	er	AWS_ENDPOINT_URL_EMR
EMR containers	er in	AWS_ENDPOINT_URL_EMR_CONTAINERS
EMR Serverless	er r	AWS_ENDPOINT_URL_EMR_SERVERLESS

serviceId	공 유 A **c** 파 일 으 사 비 스 스 별 자 키	**AWS_ENDPOINT_URL_<SERVICE>** 환경 변수
EntityResolution	e o	AWS_ENDPOINT_URL_ENTITYRESOLUTION
Elasticsearch Service	e a i	AWS_ENDPOINT_URL_ELASTICSEARCH_SERVICE
EventBridge	e g	AWS_ENDPOINT_URL_EVENTBRIDGE
Evidently	e	AWS_ENDPOINT_URL_EVIDENTLY
finspace	f	AWS_ENDPOINT_URL_FINSPACE
finspace data	f d	AWS_ENDPOINT_URL_FINSPACE_DATA
Firehose	f	AWS_ENDPOINT_URL_FIREHOSE
fis	f	AWS_ENDPOINT_URL_FIS
FMS	f	AWS_ENDPOINT_URL_FMS

serviceId	공 유 A **c** 파 일 으 사 비 스 스 별 자 키	**AWS_ENDPOINT_URL_<SERVICE>** 환경 변수
forecast	fo	AWS_ENDPOINT_URL_FORECAST
forecastquery	fo ue	AWS_ENDPOINT_URL_FORECASTQUERY
FraudDetector	fr ct	AWS_ENDPOINT_URL_FRAUDDETECTOR
FreeTier	fr	AWS_ENDPOINT_URL_FREETIER
FSx	fs	AWS_ENDPOINT_URL_FSX
GameLift	ga	AWS_ENDPOINT_URL_GAMELIFT
Glacier	gl	AWS_ENDPOINT_URL_GLACIER
Global Accelerator	gl ce	AWS_ENDPOINT_URL_GLOBAL_ACCELERATOR
Glue	gl	AWS_ENDPOINT_URL_GLUE
grafana	gr	AWS_ENDPOINT_URL_GRAFANA

serviceId	공유 AWS **co** 파일의 서비스별 자 키	**AWS_ENDPOINT_URL_<SERVICE>** 환경 변수
Greengrass	gr s	AWS_ENDPOINT_URL_GREENGRASS
GreengrassV2	gr sv	AWS_ENDPOINT_URL_GREENGRASSV2
GroundStation	gr t:	AWS_ENDPOINT_URL_GROUNDSTATION
GuardDuty	gu	AWS_ENDPOINT_URL_GUARDDUTY
Health	he	AWS_ENDPOINT_URL_HEALTH
HealthLake	he e	AWS_ENDPOINT_URL_HEALTHLAKE
Honeycode	ho	AWS_ENDPOINT_URL_HONEYCODE
IAM	ia	AWS_ENDPOINT_URL_IAM

serviceId	공 유 A **c** 파 일 으 서 비 스 스 별 자 키	**AWS_ENDPOINT_URL_<SERVICE>** 환경 변수
identitystore	i t	AWS_ENDPOINT_URL_IDENTITYSTORE
imagebuilder	i d	AWS_ENDPOINT_URL_IMAGEBUILDER
ImportExport	i o	AWS_ENDPOINT_URL_IMPORTEXPORT
Inspector	i	AWS_ENDPOINT_URL_INSPECTOR
Inspector Scan	i _	AWS_ENDPOINT_URL_INSPECTOR_SCAN
Inspector2	i 2	AWS_ENDPOINT_URL_INSPECTOR2
InternetMonitor	i o	AWS_ENDPOINT_URL_INTERNETMONITOR
IoT	i	AWS_ENDPOINT_URL_IOT

serviceId	공 유 A **c** 파 일 으 사 비 스 스 별 자 키	**AWS_ENDPOINT_URL_<SERVICE>** 환경 변수
IoT Data Plane	i p	AWS_ENDPOINT_URL_IOT_DATA_PLANE
IoT Jobs Data Plane	i d e	AWS_ENDPOINT_URL_IOT_JOBS_DATA_PLANE
IoT 1Click Devices Service	i k _	AWS_ENDPOINT_URL_IOT_1CLICK_DEVICES_ SERVICE
IoT 1Click Projects	i k s	AWS_ENDPOINT_URL_IOT_1CLICK_PROJECTS
IoTAnalytics	i i	AWS_ENDPOINT_URL_IOTANALYTICS
IotDeviceAdvisor	i a	AWS_ENDPOINT_URL_IOTDEVICEADVISOR
IoT Events	i s	AWS_ENDPOINT_URL_IOT_EVENTS

serviceId	공 유 A **c** 파 일 으 사 비 스 스 별 자 키	**AWS_ENDPOINT_URL_<SERVICE>** 환경 변수
IoT Events Data	io s_	AWS_ENDPOINT_URL_IOT_EVENTS_DATA
IoTFleetHub	io ub	AWS_ENDPOINT_URL_IOTFLEETHUB
IoTFleetWise	io is	AWS_ENDPOINT_URL_IOTFLEETWISE
IoTSecureTunneling	io tu	AWS_ENDPOINT_URL_IOTSECURETUNNELING
IoTSiteWise	io se	AWS_ENDPOINT_URL_IOTSITEWISE
IoTThingsGraph	io g:	AWS_ENDPOINT_URL_IOTTHINGSGRAPH
IoTTwinMaker	io ke	AWS_ENDPOINT_URL_IOTTWINMAKER

serviceId	공 유 A **c** 파 일 으 서 비 스 스 별 자 키	**AWS_ENDPOINT_URL_<SERVICE>** 환경 변수
IoT Wireless	i e	AWS_ENDPOINT_URL_IOT_WIRELESS
ivs	i	AWS_ENDPOINT_URL_IVS
IVS RealTime	i i	AWS_ENDPOINT_URL_IVS_REALTIME
ivschat	i	AWS_ENDPOINT_URL_IVSCHAT
Kafka	k	AWS_ENDPOINT_URL_KAFKA
KafkaConnect	k e	AWS_ENDPOINT_URL_KAFKACONNECT
kendra	k	AWS_ENDPOINT_URL_KENDRA
Kendra Ranking	k n	AWS_ENDPOINT_URL_KENDRA_RANKING
Keyspaces	k	AWS_ENDPOINT_URL_KEYSPACES
Kinesis	k	AWS_ENDPOINT_URL_KINESIS

serviceId	공유 AWS **co** 파일 으 서 비 스 스 별 자 키	**AWS_ENDPOINT_URL_<SERVICE>** 환경 변수
Kinesis Video Archived Media	k: ic iv a	AWS_ENDPOINT_URL_KINESIS_VIDEO_ARCHIVED_MEDIA
Kinesis Video Media	k: ic a	AWS_ENDPOINT_URL_KINESIS_VIDEO_MEDIA
Kinesis Video Signaling	k: ic a:	AWS_ENDPOINT_URL_KINESIS_VIDEO_SIGNALING
Kinesis Video WebRTC Storage	k: ic to e	AWS_ENDPOINT_URL_KINESIS_VIDEO_WEBRTC_STORAGE
Kinesis Analytics	k: na	AWS_ENDPOINT_URL_KINESIS_ANALYTICS

serviceId	공 유 A **c** 파 일 으 서 비 스 스 별 자 키	**AWS_ENDPOINT_URL_<SERVICE>** 환경 변수
Kinesis Analytics V2	k n v	AWS_ENDPOINT_URL_KINESIS_ANALYTICS_V2
Kinesis Video	k i	AWS_ENDPOINT_URL_KINESIS_VIDEO
KMS	k	AWS_ENDPOINT_URL_KMS
LakeFormation	l t	AWS_ENDPOINT_URL_LAKEFORMATION
Lambda	l	AWS_ENDPOINT_URL_LAMBDA
Launch Wizard	l z	AWS_ENDPOINT_URL_LAUNCH_WIZARD
Lex Model Building Service	l _ _	AWS_ENDPOINT_URL_LEX_MODEL_BUILDING_ SERVICE

serviceId	공 유 A **c** 파 일 으 사 비 스 스 별 자 키	**AWS_ENDPOINT_URL_<SERVICE>** 환경 변수
Lex Runtime Service	le me e	AWS_ENDPOINT_URL_LEX_RUNTIME_SERVICE
Lex Models V2	le s_	AWS_ENDPOINT_URL_LEX_MODELS_V2
Lex Runtime V2	le me	AWS_ENDPOINT_URL_LEX_RUNTIME_V2
License Manager	l: ar	AWS_ENDPOINT_URL_LICENSE_MANAGER
License Manager Linux Subscriptions	l: ar nu r:	AWS_ENDPOINT_URL_LICENSE_MANAGER_LIN UX_SUBSCRIPTIONS
License Manager User Subscriptions	l: ar e: ip	AWS_ENDPOINT_URL_LICENSE_MANAGER_USE R_SUBSCRIPTIONS

serviceId	공 유 A **c** 파 일 으 사 비 스 스 별 자 키	**AWS_ENDPOINT_URL_<SERVICE>** 환경 변수
Lightsail	l	AWS_ENDPOINT_URL_LIGHTSAIL
Location	l	AWS_ENDPOINT_URL_LOCATION
CloudWatch Logs	c h	AWS_ENDPOINT_URL_CLOUDWATCH_LOGS
LookoutEquipment	l u	AWS_ENDPOINT_URL_LOOKOUTEQUIPMENT
LookoutMetrics	l t	AWS_ENDPOINT_URL_LOOKOUTMETRICS
LookoutVision	l s	AWS_ENDPOINT_URL_LOOKOUTVISION
m2	m	AWS_ENDPOINT_URL_M2
Machine Learning	m e	AWS_ENDPOINT_URL_MACHINE_LEARNING
Macie2	m	AWS_ENDPOINT_URL_MACIE2

serviceId	공유 AWS **co** 파일 으 서 비 스 스 별 자 키	**AWS_ENDPOINT_URL_<SERVICE>** 환경 변수
ManagedBlockchain	ma oc	AWS_ENDPOINT_URL_MANAGEDBLOCKCHAIN
ManagedBlockchain Query	ma oc qu	AWS_ENDPOINT_URL_MANAGEDBLOCKCHAIN_QUERY
Marketplace Agreement	ma ce er	AWS_ENDPOINT_URL_MARKETPLACE_AGREEMENT
Marketplace Catalog	ma ce g	AWS_ENDPOINT_URL_MARKETPLACE_CATALOG
Marketplace Deployment	ma ce me	AWS_ENDPOINT_URL_MARKETPLACE_DEPLOYMENT

serviceId	공유 AWS **config** 파일의 서비스별 자격 증명	**AWS_ENDPOINT_URL_<SERVICE>** 환경 변수
Marketplace Entitlement Service	marketplace-entitlement	AWS_ENDPOINT_URL_MARKETPLACE_ENTITLEMENT_SERVICE
Marketplace Commerce Analytics	marketplace-commerce-analytics	AWS_ENDPOINT_URL_MARKETPLACE_COMMERCE_ANALYTICS
MediaConnect	mediaconnect	AWS_ENDPOINT_URL_MEDIACONNECT
MediaConvert	mediaconvert	AWS_ENDPOINT_URL_MEDIACONVERT
MediaLive	medialive	AWS_ENDPOINT_URL_MEDIALIVE
MediaPackage	mediapackage	AWS_ENDPOINT_URL_MEDIAPACKAGE

serviceId	공유 AWS **co** 파일 으 서 비 스 스 별 자 키	**AWS_ENDPOINT_URL_<SERVICE>** 환경 변수
MediaPackage Vod	me ag	AWS_ENDPOINT_URL_MEDIAPACKAGE_VOD
MediaPackageV2	me ag	AWS_ENDPOINT_URL_MEDIAPACKAGEV2
MediaStore	me e	AWS_ENDPOINT_URL_MEDIASTORE
MediaStore Data	me e_	AWS_ENDPOINT_URL_MEDIASTORE_DATA
MediaTailor	me or	AWS_ENDPOINT_URL_MEDIATAILOR
Medical Imaging	me ma	AWS_ENDPOINT_URL_MEDICAL_IMAGING
MemoryDB	me	AWS_ENDPOINT_URL_MEMORYDB
Marketplace Metering	ma ce ng	AWS_ENDPOINT_URL_MARKETPLACE_METERING

serviceId	공 유 A **c** 파 일 으 사 비 스 스 별 자 키	**AWS_ENDPOINT_URL_<SERVICE>** 환경 변수
Migration Hub	m _	AWS_ENDPOINT_URL_MIGRATION_HUB
mgn	m	AWS_ENDPOINT_URL_MGN
Migration Hub Refactor Spaces	m _ c e	AWS_ENDPOINT_URL_MIGRATION_HUB_REFAC TOR_SPACES
MigrationHub Config	m h g	AWS_ENDPOINT_URL_MIGRATIONHUB_CONFIG
MigrationHubOrches trator	m h t	AWS_ENDPOINT_URL_MIGRATIONHUBORCHESTRATOR
MigrationHubStrategy	m h g	AWS_ENDPOINT_URL_MIGRATIONHUBSTRATEGY
Mobile	m	AWS_ENDPOINT_URL_MOBILE

serviceId	공유 AWS **co** 파일 으 서 비 스 스 별 자 키	**AWS_ENDPOINT_URL_<SERVICE>** 환경 변수
mq	mq	AWS_ENDPOINT_URL_MQ
MTurk	mt	AWS_ENDPOINT_URL_MTURK
MWAA	mw	AWS_ENDPOINT_URL_MWAA
Neptune	ne	AWS_ENDPOINT_URL_NEPTUNE
Neptune Graph	ne ra	AWS_ENDPOINT_URL_NEPTUNE_GRAPH
neptunedata	ne ta	AWS_ENDPOINT_URL_NEPTUNEDATA
Network Firewall	ne i:	AWS_ENDPOINT_URL_NETWORK_FIREWALL
NetworkManager	ne na	AWS_ENDPOINT_URL_NETWORKMANAGER
NetworkMonitor	ne n:	AWS_ENDPOINT_URL_NETWORKMONITOR
nimble	n:	AWS_ENDPOINT_URL_NIMBLE

serviceId	공 유 A **c** 파 일 으 서 비 스 스 별 자 키	**AWS_ENDPOINT_URL_<SERVICE>** 환경 변수
OAM	oa	AWS_ENDPOINT_URL_OAM
Omics	om	AWS_ENDPOINT_URL_OMICS
OpenSearch	op h	AWS_ENDPOINT_URL_OPENSEARCH
OpenSearchServerless	op hs ss	AWS_ENDPOINT_URL_OPENSEARCHSERVERLESS
OpsWorks	op	AWS_ENDPOINT_URL_OPSWORKS
OpsWorksCM	op m	AWS_ENDPOINT_URL_OPSWORKSCM
Organizations	or ic	AWS_ENDPOINT_URL_ORGANIZATIONS
OSIS	os	AWS_ENDPOINT_URL_OSIS
Outposts	ou	AWS_ENDPOINT_URL_OUTPOSTS
p8data	p8	AWS_ENDPOINT_URL_P8DATA

serviceId	공유 AWS **co** 파일으 서비스 별 자키	**AWS_ENDPOINT_URL_<SERVICE>** 환경 변수
p8data	p8	AWS_ENDPOINT_URL_P8DATA
Panorama	pa	AWS_ENDPOINT_URL_PANORAMA
Payment Cryptography	pa ry hy	AWS_ENDPOINT_URL_PAYMENT_CRYPTOGRAPHY
Payment Cryptography Data	pa ry hy	AWS_ENDPOINT_URL_PAYMENT_CRYPTOGRAPHY_DATA
Pca Connector Ad	pc ct	AWS_ENDPOINT_URL_PCA_CONNECTOR_AD
Personalize	pe ze	AWS_ENDPOINT_URL_PERSONALIZE
Personalize Events	pe ze	AWS_ENDPOINT_URL_PERSONALIZE_EVENTS

serviceId	공 유 A **c** 파 일 으 사 비 스 스 별 자 키	**AWS_ENDPOINT_URL_<SERVICE>** 환경 변수
Personalize Runtime	pe ze e	AWS_ENDPOINT_URL_PERSONALIZE_RUNTIME
PI	pi	AWS_ENDPOINT_URL_PI
Pinpoint	pi	AWS_ENDPOINT_URL_PINPOINT
Pinpoint Email	pi em	AWS_ENDPOINT_URL_PINPOINT_EMAIL
Pinpoint SMS Voice	pi sm	AWS_ENDPOINT_URL_PINPOINT_SMS_VOICE
Pinpoint SMS Voice V2	pi sm _v	AWS_ENDPOINT_URL_PINPOINT_SMS_VOICE_V2
Pipes	pi	AWS_ENDPOINT_URL_PIPES
Polly	po	AWS_ENDPOINT_URL_POLLY

serviceId	공 유 A **c** 파 일 으 서 비 스 스 별 자 키	**AWS_ENDPOINT_URL_<SERVICE>** 환경 변수
Pricing	p:	AWS_ENDPOINT_URL_PRICING
PrivateNetworks	p: t	AWS_ENDPOINT_URL_PRIVATENETWORKS
Proton	p:	AWS_ENDPOINT_URL_PROTON
QBusiness	q	AWS_ENDPOINT_URL_QBUSINESS
QConnect	q	AWS_ENDPOINT_URL_QCONNECT
QLDB	q	AWS_ENDPOINT_URL_QLDB
QLDB Session	q i	AWS_ENDPOINT_URL_QLDB_SESSION
QuickSight	q t	AWS_ENDPOINT_URL_QUICKSIGHT
RAM	r	AWS_ENDPOINT_URL_RAM
rbin	r	AWS_ENDPOINT_URL_RBIN

serviceId	공유 AWS **co** 파일의 서비스별 자 키	**AWS_ENDPOINT_URL_<SERVICE>** 환경 변수
RDS	rd	AWS_ENDPOINT_URL_RDS
RDS Data	rd	AWS_ENDPOINT_URL_RDS_DATA
Redshift	re	AWS_ENDPOINT_URL_REDSHIFT
Redshift Data	re da	AWS_ENDPOINT_URL_REDSHIFT_DATA
Redshift Serverless	re se s	AWS_ENDPOINT_URL_REDSHIFT_SERVERLESS
Rekognition	re on	AWS_ENDPOINT_URL_REKOGNITION
repostspace	re ce	AWS_ENDPOINT_URL_REPOSTSPACE
resiliencehub	re eh	AWS_ENDPOINT_URL_RESILIENCEHUB

serviceId	공 유 A **c** 파 일 으 서 비 스 스 별 자 키	**AWS_ENDPOINT_URL_<SERVICE>** 환경 변수
Resource Explorer 2	re ex 2	AWS_ENDPOINT_URL_RESOURCE_EXPLORER_2
Resource Groups	re gr	AWS_ENDPOINT_URL_RESOURCE_GROUPS
Resource Groups Tagging API	re gr gg	AWS_ENDPOINT_URL_RESOURCE_GROUPS_TAG GING_API
RoboMaker	ro	AWS_ENDPOINT_URL_ROBOMAKER
RolesAnywhere	ro he	AWS_ENDPOINT_URL_ROLESANYWHERE
Route 53	ro	AWS_ENDPOINT_URL_ROUTE_53
Route53 Recovery Cluster	ro ec lu	AWS_ENDPOINT_URL_ROUTE53_RECOVERY_CLUSTER

serviceId	공유 AWS **co** 파일으 서비스 별 자 키	**AWS_ENDPOINT_URL_<SERVICE>** 환경 변수
Route53 Recovery Control Config	ro ec on nt	AWS_ENDPOINT_URL_ROUTE53_RECOVERY_CONTROL_CONFIG
Route53 Recovery Readiness	ro ec ea	AWS_ENDPOINT_URL_ROUTE53_RECOVERY_READINESS
Route 53 Domains	ro do	AWS_ENDPOINT_URL_ROUTE_53_DOMAINS
Route53Resolver	ro so	AWS_ENDPOINT_URL_ROUTE53RESOLVER
RUM	ru	AWS_ENDPOINT_URL_RUM
S3	s3	AWS_ENDPOINT_URL_S3
S3 Control	s3 l	AWS_ENDPOINT_URL_S3_CONTROL

serviceId	공 유 AV **co** 파 일 으 서 비 스 스 별 자 키	**AWS_ENDPOINT_URL_<SERVICE>** 환경 변수
S3Outposts	s3 s	AWS_ENDPOINT_URL_S3OUTPOSTS
SageMaker	sa	AWS_ENDPOINT_URL_SAGEMAKER
SageMaker A2I Runtime	sa _a in	AWS_ENDPOINT_URL_SAGEMAKER_A2I_RUNTIME
Sagemaker Edge	sa _e	AWS_ENDPOINT_URL_SAGEMAKER_EDGE
SageMaker FeatureSt ore Runtime	sa _f to in	AWS_ENDPOINT_URL_SAGEMAKER_FEATUREST ORE_RUNTIME
SageMaker Geospatial	sa _g ai	AWS_ENDPOINT_URL_SAGEMAKER_GEOSPATIAL

serviceId	공 유 A **c** 파 일 으 서 비 스 스 별 자 키	**AWS_ENDPOINT_URL_<SERVICE>** 환경 변수
SageMaker Metrics	s _	AWS_ENDPOINT_URL_SAGEMAKER_METRICS
SageMaker Runtime	s _	AWS_ENDPOINT_URL_SAGEMAKER_RUNTIME
savingsplans	s a	AWS_ENDPOINT_URL_SAVINGSPLANS
Scheduler	s	AWS_ENDPOINT_URL_SCHEDULER
schemas	s	AWS_ENDPOINT_URL_SCHEMAS
SimpleDB	s	AWS_ENDPOINT_URL_SIMPLEDB
Secrets Manager	s a	AWS_ENDPOINT_URL_SECRETS_MANAGER
SecurityHub	s u	AWS_ENDPOINT_URL_SECURITYHUB

serviceId	공유 AWS **co** 파일 으 서 비 스 스 별 자 키	**AWS_ENDPOINT_URL_<SERVICE>** 환경 변수
SecurityLake	se al	AWS_ENDPOINT_URL_SECURITYLAKE
ServerlessApplicationRepository	se sa io to	AWS_ENDPOINT_URL_SERVERLESSAPPLICATIONREPOSITORY
Service Quotas	se uo	AWS_ENDPOINT_URL_SERVICE_QUOTAS
Service Catalog	se at	AWS_ENDPOINT_URL_SERVICE_CATALOG
Service Catalog AppRegistry	se at pi	AWS_ENDPOINT_URL_SERVICE_CATALOG_APPREGISTRY
ServiceDiscovery	se sc	AWS_ENDPOINT_URL_SERVICEDISCOVERY
SES	se	AWS_ENDPOINT_URL_SES

serviceId	공 유 A **c** 파 일 으 사 비 스 스 별 자 키	**AWS_ENDPOINT_URL_<SERVICE>** 환경 변수
SESv2	se	AWS_ENDPOINT_URL_SESV2
Shield	sh	AWS_ENDPOINT_URL_SHIELD
signer	si	AWS_ENDPOINT_URL_SIGNER
SimSpaceWeaver	si ea	AWS_ENDPOINT_URL_SIMSPACEWEAVER
SMS	sn	AWS_ENDPOINT_URL_SMS
Snow Device Management	sn ce me	AWS_ENDPOINT_URL_SNOW_DEVICE_MANAGEMENT
Snowball	sn	AWS_ENDPOINT_URL_SNOWBALL
SNS	sn	AWS_ENDPOINT_URL_SNS
SQS	sq	AWS_ENDPOINT_URL_SQS
SSM	ss	AWS_ENDPOINT_URL_SSM

serviceId	공 유 AV **co** 파 일 으 서 비 스 스 별 자 키	**AWS_ENDPOINT_URL_<SERVICE>** 환경 변수
SSM Contacts	ss ct	AWS_ENDPOINT_URL_SSM_CONTACTS
SSM Incidents	ss er	AWS_ENDPOINT_URL_SSM_INCIDENTS
Ssm Sap	ss	AWS_ENDPOINT_URL_SSM_SAP
SSO	ss	AWS_ENDPOINT_URL_SSO
SSO Admin	ss	AWS_ENDPOINT_URL_SSO_ADMIN
SSO OIDC	ss	AWS_ENDPOINT_URL_SSO_OIDC
SFN	st	AWS_ENDPOINT_URL_SFN
Storage Gateway	st at	AWS_ENDPOINT_URL_STORAGE_GATEWAY
STS	st	AWS_ENDPOINT_URL_STS
SupplyChain	su in	AWS_ENDPOINT_URL_SUPPLYCHAIN

serviceId	공 유 A **c** 파 일 으 서 비 스 스 별 자 키	**AWS_ENDPOINT_URL_<SERVICE>** 환경 변수
Support	su	AWS_ENDPOINT_URL_SUPPORT
Support App	su pp	AWS_ENDPOINT_URL_SUPPORT_APP
SWF	sw	AWS_ENDPOINT_URL_SWF
synthetics	sy s	AWS_ENDPOINT_URL_SYNTHETICS
Textract	te	AWS_ENDPOINT_URL_TEXTRACT
Timestream InfluxDB	ti m_ b	AWS_ENDPOINT_URL_TIMESTREAM_INFLUXDB
Timestream Query	ti m_	AWS_ENDPOINT_URL_TIMESTREAM_QUERY
Timestream Write	ti m_	AWS_ENDPOINT_URL_TIMESTREAM_WRITE
tnb	tn	AWS_ENDPOINT_URL_TNB

serviceId	공 유 A **c** 파 일 으 사 비 스 스 별 자 키	**AWS_ENDPOINT_URL_<SERVICE>** 환경 변수
Transcribe	t e	AWS_ENDPOINT_URL_TRANSCRIBE
Transfer	t	AWS_ENDPOINT_URL_TRANSFER
Translate	t	AWS_ENDPOINT_URL_TRANSLATE
TrustedAdvisor	t v	AWS_ENDPOINT_URL_TRUSTEDADVISOR
VerifiedPermissions	v e s	AWS_ENDPOINT_URL_VERIFIEDPERMISSIONS
Voice ID	v	AWS_ENDPOINT_URL_VOICE_ID
VPC Lattice	v c	AWS_ENDPOINT_URL_VPC_LATTICE
WAF	w	AWS_ENDPOINT_URL_WAF

serviceId	공유 AWS **co** 파일의 서비스별 자 키	**AWS_ENDPOINT_URL_<SERVICE>** 환경 변수
WAF Regional	wa na	AWS_ENDPOINT_URL_WAF_REGIONAL
WAFV2	wa	AWS_ENDPOINT_URL_WAFV2
WellArchitected	we te	AWS_ENDPOINT_URL_WELLARCHITECTED
Wisdom	wi	AWS_ENDPOINT_URL_WISDOM
WorkDocs	wo	AWS_ENDPOINT_URL_WORKDOCS
WorkLink	wo	AWS_ENDPOINT_URL_WORKLINK
WorkMail	wo	AWS_ENDPOINT_URL_WORKMAIL
WorkMailMessageFlow	wo es w	AWS_ENDPOINT_URL_WORKMAILMESSAGEFLOW
WorkSpaces	wo s	AWS_ENDPOINT_URL_WORKSPACES

serviceId	공 유 A **c** 파 일 으 서 비 스 스 별 자 키	**AWS_ENDPOINT_URL_<SERVICE>** 환경 변수
WorkSpaces Thin Client	w s_ i	AWS_ENDPOINT_URL_WORKSPACES_THIN_CLIENT
WorkSpaces Web	w s_	AWS_ENDPOINT_URL_WORKSPACES_WEB
XRay	x	AWS_ENDPOINT_URL_XRAY

스마트 구성 기본값

> **Note**
>
> 설정 페이지의 레이아웃을 이해하거나 다음에 나오는 Support AWS SDKs이 가이드의 설정 페이지 이해.

스마트 구성 기본값 기능을 사용하면 AWS SDKs 다른 구성 설정에 대해 사전 정의되고 최적화된 기본값을 제공할 수 있습니다.

다음을 사용하여 이 기능을 구성하십시오.

defaults_mode - 공유 AWS **config** 파일 설정, **AWS_DEFAULTS_MODE** - 환경 변수, **aws.defaultsMode** - JVM 시스템 속성: Java/Kotlin만 해당

이 설정을 사용하면 사용자 애플리케이션 아키텍처에 맞는 모드를 선택하여 애플리케이션에 최적화된 기본값을 제공할 수 있습니다. AWS SDK 설정에 명시적으로 설정된 값이 있는 경우 해당 값이 항상 우선합니다. AWS SDK 설정에 명시적으로 설정된 값이 없고 defaults_mode가 레거시와 같지 않은 경우이 기능은 애플리케이션에 최적화된 다양한 설정에 대해 서로 다른 기본값을 제공할 수 있습니다. 설정에는 HTTP 통신 설정, 재시도 동작, 서비스 리전 엔드포인트 설정 및 잠재적으로 모든 SDK 관련 구성이 포함될 수 있습니다. 이 기능을 사용하는 고객은 일반 사용 시나리오에 맞게 조정된 새 구성 기본값을 얻을 수 있습니다. defaults_mode이 legacy같지 않은 경우에 SDK를 업그레이드할 때 사용자 애플리케이션 테스트를 수행하는 것이 좋습니다. 제공된 기본값은 모범 사례가 발전함에 따라 변경될 수 있기 때문입니다.

기본값: legacy

참고: 새 주요 버전의 SDK는 standard로 기본 설정됩니다.

유효값:

- legacy— SDK에 따라 달라지고 defaults_mode의 설정 이전에 존재했던 기본 설정을 제공합니다.
- standard— 대부분의 시나리오에서 안전하게 실행할 수 있는 최신 권장 기본값을 제공합니다.
- in-region - 표준 모드를 기반으로 하며 동일한 내에서 AWS 서비스 를 호출하는 애플리케이션에 맞게 조정된 최적화를 포함합니다 AWS 리전.
- cross-region - 표준 모드를 기반으로 하며 다른 리전 AWS 서비스 에서를 호출하는 애플리케이션에 맞게 조정된 최적화를 포함합니다.
- mobile— 표준 모드를 기반으로 구축하며 모바일 애플리케이션에 맞게 조정된 최적화를 포함합니다.
- auto— 표준 모드를 기반으로 구축하며 실험적 기능을 포함합니다. SDK는 런타임 환경을 검색하여 적절한 설정을 자동으로 결정합니다. 자동 감지는 휴리스틱 기반이며 정확도가 100%는 아닙니다. 런타임 환경을 확인할 수 없는 경우 standard 모드가 사용됩니다. 자동 감지는 인스턴스 메타데이터를 쿼리하여 지연 시간을 초래할 수 있습니다. 시작 지연 시간이 애플리케이션에 중요한 경우에는 명시적 지연 시간을 defaults_mode을 대신 선택하는 것이 좋습니다.

config 파일에서 이 값을 설정하는 예:

```
[default]
defaults_mode = standard
```

다음 파라미터는 defaults_mode의 선택에 따라 최적화될 수 있습니다.

- retryMode— SDK 재시도 방법을 지정합니다. 재시도 동작을(를) 참조하세요.
- stsRegionalEndpoints - SDK가 AWS Security Token Service ()와 통신하는 데 사용하는 AWS 서비스 엔드포인트를 결정하는 방법을 지정합니다AWS STS. AWS STS 리전 엔드포인트을(를) 참조하세요.
- s3UsEast1RegionalEndpoints - SDK가 us-east-1 리전의 Amazon S3와 통신하는 데 사용하는 AWS 서비스 엔드포인트를 결정하는 방법을 지정합니다.
- connectTimeoutInMillis— 소켓에서 처음 연결을 시도한 후 제한 시간이 초과되기까지의 시간. 클라이언트가 연결 핸드셰이크 완료를 수신하지 못하면 클라이언트는 작업을 포기하고 중단합니다.
- tlsNegotiationTimeoutInMillis— CLIENT HELLO 메시지가 전송된 시점부터 클라이언트와 서버가 암호를 완전히 협상하고 키를 교환할 때까지 TLS 핸드셰이크에 걸릴 수 있는 최대 시간입니다.

각 설정의 기본값은 사용자 응용 프로그램에서 선택한 defaults_mode에 따라 달라집니다. 이러한 값은 현재 다음과 같이 설정되어 있습니다(변경될 수 있음).

파라미터	**standard** 모드	**in-region** 모드	**cross-region** 모드	**mobile** 모드
retryMode	standard	standard	standard	standard
stsRegionalEndpoints	regional	regional	regional	regional
s3UsEast1RegionalEndpoints	regional	regional	regional	regional
connectTimeoutInMillis	3100	1100	3100	30000
tlsNegotiationTime	3100	1100	3100	30000

파라미터	**standard** 모드	**in-region** 모드	**cross-region** 모드	**mobile** 모드
`outInMillis`				

예를 들어, 선택한 `defaults_mode` 값이 `standard`이면 `standard`의 값이 같으면 `retry_mode` (유효한 `retry_mode`옵션에서)의 값이 할당되고 `regional`의 값이 `stsRegionalEndpoints`에 할당됩니다(유효한 `stsRegionalEndpoints` 옵션에서).

AWS SDKs 도구 지원

다음 SDK는 이 주제에서 설명하는 기능 및 설정을 지원합니다. 모든 일부 예외가 기록됩니다. 모든 JVM 시스템 속성 설정은 AWS SDK for Java 및 AWS SDK for Kotlin 에서만 지원됩니다.

SDK	지원	참고 또는 추가 정보
AWS CLI v2	아니요	
SDK for C++	예	최적화되지 않은 매개변수:`stsRegionalEndpoints` ,`s3UsEast1RegionalEndpoints` ,`tlsNegotiationTimeoutInMillis` .
SDK for Go V2 (1.x)	예	최적화되지 않은 매개변수:`stsRegionalEndpoints` ,`s3UsEast1RegionalEndpoints` ,`retryMode` .
SDK for Go 1.x (V1)	아니요	
SDK for Java 2.x	예	최적화되지 않은 매개변수:`stsRegionalEndpoints` .
SDK for Java 1.x	아니요	

SDK	지원	참고 또는 추가 정보
SDK for JavaScript 3.x	예	최적화되지 않은 매개변수:stsRegionalEndpoints ,s3UsEast1RegionalEndpoints ,tlsNegotiationTimeoutInMillis . connectTimeoutInMillis 가 connectionTimeout 호출됩니다.
SDK for JavaScript 2.x	아니요	
SDK for Kotlin	아니요	
.NET 4.x용 SDK	예	최적화되지 않은 매개변수:connectTimeoutInMillis ,tlsNegotiationTimeoutInMillis .
SDK for .NET 3.x	예	최적화되지 않은 매개변수:connectTimeoutInMillis ,tlsNegotiationTimeoutInMillis .
SDK for PHP 3.x	예	최적화되지 않은 매개변수:tlsNegotiationTimeoutInMillis .
SDK for Python (Boto3)	예	최적화되지 않은 매개변수:tlsNegotiationTimeoutInMillis .
SDK for Ruby 3.x	예	

SDK	지원	참고 또는 추가 정보
SDK for Rust	아니요	
SDK for Swift	아니요	
PowerShell V5용 도구	예	최적화되지 않은 매개변수:connectTimeoutInMillis ,tlsNegotiationTimeoutInMillis .
PowerShell V4용 도구	예	최적화되지 않은 매개변수:connectTimeoutInMillis ,tlsNegotiationTimeoutInMillis .

AWS 공통 런타임(CRT) 라이브러리

AWS 공통 런타임(CRT) 라이브러리는 SDKs. CRT는 C로 작성된 독립 패키지의 모듈식 제품군으로, 각 패키지는 우수한 성능을 제공하고 다양한 필수 기능을 위한 최소한의 설치 공간을 제공합니다. 이러한 기능은 모든 SDK에서 공통적이며 공유되므로 코드 재사용, 최적화 및 정확성이 향상됩니다. 패키지는 다음과 같습니다.

- awslabs/aws-c-auth: AWS 클라이언트 측 인증(표준 자격 증명 공급자 및 서명(sigv4))
- awslabs/aws-c-cal: 암호화 프리미티브 유형, 해시 (MD5, SHA256, SHA256 HMAC), 서명자, AES
- awslabs/aws-c-common: 기본 데이터 구조, 스레딩/동기화 프리미티브 유형, 버퍼 관리, stdlib 관련 함수
- awslabs/aws-c-compression: 압축 알고리즘 (허프만 인코딩/디코딩)
- awslabs/aws-c-event-stream: 이벤트 스트림 메시지 처리 (헤더, 프렐루드, 페이로드, crc/trailer), 이벤트 스트림을 통한 원격 프로시저 호출 (RPC)구현
- awslabs/aws-c-http: HTTP/1.1 및 HTTP/2 사양의 C99 구현
- awslabs/aws-c-io: 소켓 (TCP, UDP), DNS, 파이프, 이벤트 루프, 채널, SSL/TLS
- awslabs/aws-c-iot: 디바이스와 AWS IoT 클라우드 서비스 통합의 C99 구현
- awslabs/aws-c-mqtt: 사물 인터넷 (IoT)을 위한 간단한 표준 메시징 프로토콜
- awslabs/aws-c-s3: Amazon S3 서비스와의 통신을 위한 C99 라이브러리 구현으로, 고대역폭 Amazon EC2 인스턴스의 처리량을 극대화하도록 설계되었습니다
- awslabs/aws-c-sdkutils: AWS 프로필 구문 분석 및 관리를 위한 유틸리티 라이브러리
- awslabs/aws-checksums: 효율적인 소프트웨어 구현으로 대체되는 크로스 플랫폼 하드웨어 가속 CRC32c 및 CRC32
- awslabs/aws-lc: Google BoringSSL 프로젝트 및 OpenSSL 프로젝트의 코드를 기반으로 AWS 및 해당 고객을 위해 AWS 암호화 팀이 유지 관리하는 범용 암호화 라이브러리
- awslabs/s2n: TLS/SSL 프로토콜의 C99 구현, 보안을 최우선으로 하여 작고 빠르도록 설계되었습니다

CRT는 Go 및 Rust를 제외한 모든 SDKs를 통해 사용할 수 있습니다.

CRT 종속성

CRT 라이브러리는 복잡한 관계와 종속성 네트워크를 형성합니다. 소스에서 직접 CRT를 구축해야 하는 경우 이러한 관계를 아는 것이 도움이 됩니다. 그러나 대부분의 사용자는 언어 SDK(예: C++용 AWS SDK 또는 Java용 AWS SDK) 또는 언어 IoT 디바이스 SDK(예: C++용 AWS IoT SDK 또는 Java용 AWS IoT SDK)를 통해 CRT 기능에 액세스합니다. 다음 다이어그램에서 언어 CRT 바인딩 상자는 특정 언어 SDK의 CRT 라이브러리를 래핑하는 패키지를 나타냅니다. 이 패키지는 다음과 같은 형식의 `aws-crt-*` 패키지 모음입니다. 여기서 '*'는 SDK 언어 (예: `aws-crt-cpp` 혹은 `aws-crt-java`)입니다.

다음은 CRT 라이브러리의 계층적 종속성을 보여줍니다.

AWS SDKs 및 도구 유지 관리 정책

개요

이 문서에서는 모바일 및 IoT SDKs를 포함한 AWS 소프트웨어 개발 키트(SDKs) 및 도구에 대한 유지 관리 정책과 기본 종속성을 간략하게 설명합니다. AWS 는 신규 또는 업데이트된 AWS APIs, 새로운 기능, 개선 사항, 버그 수정, 보안 패치 또는 설명서 업데이트에 대한 지원이 포함될 수 있는 업데이트를 AWS SDKs 및 도구에 제공합니다. 업데이트는 종속성, 언어 런타임 및 운영 체제의 변경 사항을 해결할 수도 있습니다. AWS SDK 릴리스는 패키지 관리자(예: Maven, NuGet, PyPI)에게 게시되며 GitHub에서 소스 코드로 사용할 수 있습니다.

최신 기능, 보안 업데이트 및 기본 종속성을 유지하려면 SDK 릴리스를 최신 상태로 유지하도록 권장됩니다. 지원되지 않는 SDK 버전을 계속 사용하는 것은 권장되지 않으며 사용자의 재량에 따라 수행됩니다.

버전 관리

AWS SDK 릴리스 버전은 X.Y.Z 형식이며, 여기서 X는 메이저 버전을 나타냅니다. SDK의 주 버전이 올라가면 이 SDK가 해당 언어의 새로운 관용구와 패턴을 지원하기 위해 상당하고 상당한 변화를 겪었다는 것을 알 수 있습니다. 메이저 버전은 공용 인터페이스(예: 클래스, 메서드, 유형 등), 동작 또는 의미가 변경될 때 도입됩니다. 애플리케이션이 최신 SDK 버전에서 작동하려면 애플리케이션을 업데이트해야 합니다. AWS이 제공하는 업그레이드 지침에 따라 메이저 버전을 신중하게 업데이트하는 것이 중요합니다.

SDK 메이저 버전 수명 주기

메이SDKs 및 도구 버전의 수명 주기는 아래에 설명된 5단계로 구성됩니다.

- 개발자 미리 보기 (0 단계)- 이 단계에서는 SDK가 지원되지 않으므로 프로덕션 환경에 사용해서는 안 되며 조기 액세스 및 피드백 목적으로만 사용됩니다. 향후 릴리스에서는 단절적 변경이 도입될 수 있습니다. 가 릴리스를 안정적인 제품으로 AWS 식별하면 릴리스 후보로 표시될 수 있습니다. 릴리스 후보는 중대한 버그가 나타나지 않는 한 GA 릴리스가 될 준비가 되어 있으며 완전한 AWS 지원을 받게 됩니다.
- 일반 가용성(GA)(1단계) -이 단계에서는 SDKs 완전히 지원됩니다. AWS 는 새 서비스에 대한 지원, 기존 서비스에 대한 API 업데이트, 버그 및 보안 수정이 포함된 정기적인 SDK 릴리스를 제공합니다.

도구의 경우 AWS 는 새로운 기능 업데이트 및 버그 수정이 포함된 일반 릴리스를 제공합니다. AWS 는 최소 24개월 동안 SDK의 GA 버전을 지원합니다.

- 유지 관리 공지(2단계) - AWS SDK가 유지 관리 모드로 전환되기 최소 6개월 전에 공개 공지를 합니다. 이 기간 동안에도 SDK는 완전히 지원됩니다. 일반적으로 유지 관리 모드는 다음 메이저 버전이 GA로 전환되는 시점에 동시 발표됩니다.
- 유지 관리 (3 단계)- 유지 관리 모드에서는 AWS 는 SDK 릴리스를 중요한 버그 수정 및 보안 문제에 한정합니다. SDK는 신규 또는 기존 서비스에 대한 API 업데이트를 수신하거나 새 리전 지역을 지원하도록 업데이트 되지 않습니다. 달리 명시되지 않는 한 유지 관리 모드의 기본 기간은 12개월입니다.
- 지원 종료 (4 단계)- SDK가 지원 종료 되면 더 이상 업데이트나 릴리스가 없습니다. 이전에 게시된 릴리스는 공개 패키지 관리자를 통해 계속 사용할 수 있으며 코드는 GitHub에 그대로 유지됩니다. GitHub 리포지토리는 보관될 수 있습니다. end-of-support에 도달한 SDK 사용은 사용자의 재량에 따라 수행됩니다. 새 메이저 버전으로 업그레이드할 것을 당사는 권장합니다.

다음은 SDK 메이저 버전 수명 주기의 시각적 그림입니다. 아래 표시된 타임라인은 예시용이며 구속력이 없다는 점에 유의하십시오.

종속성 수명 주기

AWS SDKs 언어 런타임, 운영 체제 또는 타사 라이브러리 및 프레임워크와 같은 기본 종속성이 있습니다. 이러한 종속성은 일반적으로 언어 커뮤니티 또는 해당 특정 구성 요소를 소유한 공급업체와 연동되어 있습니다. 각 커뮤니티 또는 공급업체는 해당 제품에 대한 자체 지원 종료 일정을 게시합니다.

다음 용어는 기본 타사 종속성을 분류하는 데 사용됩니다.

- 운영 체제 (OS): 아마존 리눅스 AMI, 아마존 리눅스 2, 윈도우 2008, 윈도우 2012, 윈도우 2016 등을 예로 들 수 있습니다.
- 언어 런타임: 자바 7, 자바 8, 자바 11, .NET 코어, 표준 .NET, .NET PCL 등을 예로 들 수 있습니다.
- 타사 라이브러리/프레임워크: OpenSSL, .NET 프레임워크 4.5, Java EE 등을 예로 들 수 있습니다.

커뮤니티 또는 벤더가 종속성에 대한 지원을 종료한 후에도 최소 6개월 동안 SDK 종속성을 계속 지원하는 것이 정책입니다. 하지만 이 정책은 특정 종속성에 따라 달라질 수 있습니다.

> **Note**
>
> AWS 는 메이저 SDK 버전을 늘리지 않고 기본 종속성에 대한 지원을 중지할 수 있는 권한을 보유합니다.

통신 메서드

유지 관리 공고는 여러 방법으로 전달됩니다.

- 영향 받는 계정에는 특정 SDK 버전에 대한 지원 종료 계획을 알리는 이메일 공지가 발송됩니다. 이메일은 지원 종료 경로를 설명하고, 캠페인 타임 라인을 정하고, 업그레이드 지침을 제공합니다.
- AWS API 참조 설명서, 사용 설명서, SDK 제품 마케팅 페이지 및 GitHub readme(s)과 같은 SDK 설명서가 캠페인 타임라인을 표시하고 영향을 받는 애플리케이션 업그레이드에 대한 지침을 제공하도록 업데이트됩니다.
- end-of-support 경로를 간략하게 설명하고 캠페인 타임라인을 반복하는 AWS 블로그 게시물이 게시됩니다.
- 사용 중단 경고가 SDK에 추가되어 지원 종료 경로를 설명하고 SDK 설명서가 링크로 연결됩니다.

사용 가능한 AWS SDKs 및 도구의 메이저 버전 목록과 유지 관리 수명 주기의 현재 위치를 보려면 섹션을 참조하세요버전 수명 주기.

AWS SDKs 및 도구 버전 수명 주기

아래 표에는 사용 가능한 AWS 소프트웨어 개발 키트(SDK) 메이저 버전 목록과 관련 타임라인과 함께 유지 관리 수명 주기의 현재 위치가 나와 있습니다. 주요 버전의 AWS SDKs 및 도구 수명 주기와 기본 종속성에 대한 자세한 내용은 섹션을 참조하세요유지 관리 정책.

SDK	메이저 버전	현재 단계	일반 출시 날짜	주석
AWS CLI	1.x	정식 출시	9/2/2013	
AWS CLI	2.x	정식 출시	2/10/2020	
SDK for C++	1.x	정식 출시	9/2/2015	
SDK for Go V2	V2 1.x	정식 출시	1/19/2021	
SDK for Go	1.x	지원 종료	11/19/2015	
SDK for Java	1.x	유지 관리	3/25/2010	세부 정보 및 날짜는 공지를 참조하세요.
SDK for Java	2.x	정식 출시	11/20/2018	
SDK for JavaScript	1.x	지원 종료	5/6/2013	
SDK for JavaScript	2.x	지원 종료	6/19/2014	
SDK for JavaScript	3.x	정식 출시	12/15/2020	
SDK for Kotlin	1.x	정식 출시	11/27/2023	
SDK for .NET	1.x	지원 종료	2009년 11월	
SDK for .NET	2.x	지원 종료	11/8/2013	
SDK for .NET	3.x	정식 출시	7/28/2015	

SDK	메이저 버전	현재 단계	일반 출시 날짜	주석
SDK for .NET	4.x	정식 출시	4/28/2025	
SDK for PHP	2.x	지원 종료	11/2/2012	
SDK for PHP	3.x	정식 출시	5/27/2015	
SDK for Python (Boto2)	1.x	지원 종료	7/13/2011	
SDK for Python (Boto3)	1.x	정식 출시	6/22/2015	
SDK for Python (Botocore)	1.x	정식 출시	6/22/2015	
SDK for Ruby	1.x	지원 종료	7/14/2011	
SDK for Ruby	2.x	지원 종료	2/15/2015	
SDK for Ruby	3.x	정식 출시	8/29/2017	
SDK for Rust	1.x	정식 출시	11/27/2023	
SDK for Swift	1.x	정식 출시	9/17/2024	
PowerShell용 도구	2.x	지원 종료	11/8/2013	
PowerShell용 도구	3.x	지원 종료	7/29/2015	
PowerShell용 도구	4.x	정식 출시	11/21/2019	
PowerShell용 도구	5.x	정식 출시	6/23/2025	

언급되지 않은 SDK 또는 도구를 검색하시나요? 예를 들어 암호화 SDKs, IoT 디바이스 SDKs 및 모바일 SDKs는이 가이드에 포함되지 않습니다. 이러한 다른 도구에 대한 설명서를 찾으려면 에서 빌드할 도구를 AWS참조하세요.

AWS SDKs 도구에 대한 문서 기록 참조 가이드

다음 표에는 AWS SDK 및 도구 참조 가이드의 중요한 추가 및 업데이트가 설명되어 있습니다. 이 설명서에 대한 업데이트 알림을 받으려면 RSS 피드에 가입하면 됩니다.

변경 사항	설명	날짜
새 S3 Express One Zone 설정 추가	세션 인증을 비활성화하기 위해 새 S3 Express One Zone 설정을 추가합니다.	2025년 10월 13일
새 인증 의사 결정 트리 추가	옵션 간 인증 결정을 지원하는 새 의사 결정 트리를 추가합니다.	2025년 9월 23일
새 인증 체계 기능 추가	새 인증 체계 기능 추가. AWS STS 리전 엔드포인트 업데이트.	2025년 8월 18일
새로운 버전의 Tools for PowerShell 추가	모든 설정 참조 AWS SDKs와의 호환성 테이블에 최신 버전의 Tools for PowerShell 지원을 추가합니다. 호스트 접두사 삽입 기능이 추가되었습니다.	2025년 6월 23일
페이지 제목 업데이트	더 많은 제목, 테이블 제목, 추상 및 SEO 업데이트.	2025년 3월 5일
페이지 제목 업데이트	더 설명적인 제목을 사용하도록 콘텐츠를 업데이트합니다.	2025년 2월 24일
설정 참조에 Swift SDK 추가	모든 Setting reference Compatibility with AWS SDKs 테이블에 Swift SDK 지원을 추가합니다.	2024년 9월 17일

SDK for Java 1.x 시스템 속성	AWS SDK for Java 1.x로 지원되는 JVM 시스템 구성 설정에 대한 세부 정보를 추가합니다.	2024년 5월 30일
설정 업데이트	JVM 시스템 구성 설정을 추가합니다.	2024년 3월 27일
호환성 테이블 업데이트	SDK 지원 호환성 업데이트, IAM Identity Center 절차 업데이트.	2024년 2월 20일
컨테이너 보안 인증 업데이트. IMDS 업데이트.	Amazon EKS 지원 추가. IMDSv1 폴백을 비활성화하기 위한 설정 추가.	2023년 12월 29일
요청 압축	요청 압축 기능에 대한 설정 추가.	2023년 12월 27일
호환성 테이블	SDK 및 도구 기능에 대한 호환성 테이블이 SDK for Kotlin, SDK for Rust, AWS Tools for PowerShell등을 포함하도록 업데이트되었습니다.	2023년 12월 10일
인증 업데이트	SDK 및 도구에 지원되는 인증 방법이 업데이트되었습니다.	2023년 7월 1일
IAM 모범 사례 업데이트	IAM 모범 사례에 따라 가이드가 업데이트되었습니다. 자세한 내용은 IAM의 보안 모범 사례를 참조하십시오.	2023년 2월 27일
SSO 업데이트	새 SSO 토큰 구성을 위한 SSO 보안 인증 업데이트.	2022년 11월 19일
설정 업데이트	일반 구성 및 Amazon S3 다중 리전 액세스 포인트 지원 표 업데이트.	2022년 11월 17일

설정 업데이트	IMDS 클라이언트 및 IMDS 보안 인증의 명확성 업데이트. 환경 변수 업데이트.	2022년 11월 4일
시작 페이지 업데이트	Amazon CodeWhisperer 출시.	2022년 9월 22일
싱글 사인온 서비스 이름 변경	AWS SSO가 이제 참조됨을 반영하기 위한 업데이트입니다 AWS IAM Identity Center.	2022년 7월 26일
설정 업데이트	구성 파일 세부 정보 및 지원 설정에 대한 사소한 업데이트.	2022년 6월 15일
업데이트	이 안내서의 거의 모든 부분이 업데이트되었습니다.	2022년 2월 1일
최초 릴리스	이 설명서의 최초 릴리스가 일반에 공개되었습니다.	2020년 3월 13일

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