



참조 안내서

AWS SDKs 및 도구



AWS SDKs 및 도구: 참조 안내서

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

Amazon의 상표 및 트레이드 드레스는 Amazon 외 제품 또는 서비스와 함께, Amazon 브랜드 이미지를 떨어뜨리거나 고객에게 혼동을 일으킬 수 있는 방식으로 사용할 수 없습니다. Amazon이 소유하지 않은 기타 모든 상표는 Amazon과 제휴 관계이거나 관련이 있거나 후원 관계와 관계없이 해당 소유자의 자산입니다.

Table of Contents

AWS SDKs 및 도구 참조 가이드	1
개발자 리소스	2
도구 키트 원격 측정 알림	2
구성	4
공유 config 및 credentials 파일	5
프로파일	5
구성 파일 형식	6
보안 인증 파일의 형식	9
공유 파일의 위치	10
홈 디렉터리 해상도	10
이러한 파일의 기본 위치를 변경하십시오.	11
환경 변수	12
환경 변수를 설정하는 방법	12
서버리스 환경 변수 설정	13
JVM 시스템 속성	14
JVM 시스템 속성을 설정하는 방법	14
인증 및 액세스	16
AWS Builder ID	17
IAM Identity Center 인증	18
IAM Identity Center를 사용하여 프로그래밍 방식 액세스 구성	18
IAM Identity Center 인증 이해하기	21
IAM Roles Anywhere	25
1단계: IAM Roles Anywhere 구성	25
2단계: IAM Roles Anywhere 사용	25
역할 수입	26
IAM 역할 수입	27
역할 수입(웹)	28
웹 자격 증명 또는 OpenID Connect와 페더레이션	29
AWS 액세스 키	30
단기 보안 인증 정보를 사용합니다	30
장기 보안 인증 정보 사용	31
단기 보안 인증	32
장기 보안 인증	33
IAM EC2인스턴스용 역할	36

IAM 역할 생성	37
Amazon EC2 인스턴스를 시작하고 IAM 역할을 지정합니다.	37
EC2인스턴스에 연결	38
인스턴스에서 애플리케이션을 실행합니다. EC2	38
설정 참조	39
서비스 클라이언트 생성	39
설정의 우선 순위	39
설정 페이지	40
Config 파일 설정 목록	42
Credentials 파일 설정 목록	45
환경 변수 목록	46
JVM 시스템 속성 목록	50
표준화된 보안 인증 공급자	52
자격 증명 공급자 체인 이해	53
SDK-특정 및 도구별 자격 증명 공급자 체인	54
AWS 액세스 키	55
역할 공급자 수입	58
컨테이너 제공업체	64
IAM Identity Center 공급자	67
IMDS 공급자	73
프로세스 공급자	78
표준화된 기능	82
계정 기반 엔드포인트	82
애플리케이션 ID	85
Amazon EC2 인스턴스 메타데이터	88
Amazon S3 액세스 포인트	90
Amazon S3 다중 리전 액세스 포인트	92
AWS 리전	95
AWS STS 리전 엔드포인트	98
듀얼 스택 및 엔드포인트 FIPS	102
엔드포인트 검색	104
일반 구성	106
IMDS 클라이언트	110
재시도 동작	113
요청 압축	119
서비스별 엔드포인트	121

스마트 구성 기본값	176
공통 런타임	181
CRT 종속성	181
유지 관리 정책	183
개요	183
버전 관리	183
SDK 메이저 버전 라이프사이클	183
종속성 라이프사이클	184
통신 메서드	185
버전 지원	186
문서 기록	189
AWS 용어집	191
.....	cxcii

AWS SDKs 및 도구 참조 가이드

많은 SDKs 및 도구가 공유 설계 사양 또는 공유 라이브러리를 통해 몇 가지 공통 기능을 공유합니다.

이 안내서에는 다음과 관련된 정보가 포함되어 있습니다.

- [구성](#) - 공유 config 및 credentials 파일 또는 환경 변수를 사용하여 및 도구를 구성하는 AWS SDKs 방법.
- [인증 및 액세스](#) -를 개발할 AWS 때 코드 또는 도구를 인증하는 방법을 설정합니다 AWS 서비스.
- [설정 참조](#) - 인증 및 구성에 사용할 수 있는 모든 표준 설정에 대한 참조
- [AWS 공통 런타임\(CRT\) 라이브러리](#) - 거의 모든에서 사용할 수 있는 공유 AWS 공통 런타임(CRT) 라이브러리의 개요입니다 SDKs.
- [AWS SDK 및 도구 유지 관리 정책](#)에서는 모바일 및 사물 인터넷(IoT SDKs)를 포함한 소프트웨어 개발 키트() 및 도구에 대한 AWS 유지 관리 정책 및 버전 관리 SDKs와 기본 종속성을 다룹니다.

이 AWS SDKs 및 도구 참조 가이드는 여러 SDKs 및 도구에 적용할 수 있는 정보의 기반이 되기 위한 것입니다. 사용 중인 SDK 또는 도구에 대한 특정 가이드는 여기에 제시된 모든 정보 외에 추가로 사용해야 합니다. 다음은 이 가이드의 관련 자료 섹션이 있는 SDK 및 도구입니다.

사용 중인 제품:	설명서 관련 섹션:
<ul style="list-style-type: none"> • SDK 또는 도구 	AWS SDK 및 도구 유지 관리 정책
<ul style="list-style-type: none"> • 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 • Amazon CodeCatalyst 	구성 인증 및 액세스 AWS SDK 및 도구 유지 관리 정책

사용 중인 제품:	설명서 관련 섹션:
<ul style="list-style-type: none"> • AWS CodeCommit • AWS CodeDeploy • AWS CodePipeline 	
<ul style="list-style-type: none"> • 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 	<ul style="list-style-type: none"> • 구성 • 인증 및 액세스 • 설정 참조 • AWS 공통 런타임(CRT) 라이브러리 • AWS SDK 및 도구 유지 관리 정책 • AWS SDKs 및 툴 버전 지원

개발자 리소스

에서 애플리케이션을 개발하는 데 도움이 될 수 있는 도구에 대한 개요는 [빌드 기반 도구를 AWS](#) AWS 참조하세요. 지원에 대한 자세한 내용은 [AWS 지식 센터](#)를 참조하십시오.

Amazon Q Developer는 AWS 애플리케이션을 이해, 구축, 확장 및 운영하는 데 도움이 되는 생성형 AI 기반 대화형 어시스턴트입니다. 빌드를 가속화하기 위해 Amazon Q AWS를 지원하는 모델을 고품질 AWS 콘텐츠로 보강하여 더 완전하고 실행 가능하며 참조된 답변을 생성합니다. 자세한 내용은 Amazon Q Developer 사용 설명서의 [What is Amazon Q Developer?](#)를 참조하세요.

도구 키트 원격 측정 알림

AWS 통합 개발 환경(IDE) 도구 키트는의 AWS 서비스에 액세스할 수 있는 플러그인 및 확장 프로그램입니다IDE. Amazon Q IDE 플러그인 및 확장을 사용하면에서 생성형 AI 지원을 사용할 수 있습니다IDE. 각 IDE 도구 키트에 대한 자세한 내용은 이전 표의 도구 키트 사용 설명서를 참조하세요. 에서

Amazon Q를 사용하는 방법에 대한 자세한 내용은 [Amazon Q 개발자 안내서의 주제에서 Amazon Q 사용을 IDE](#) IDE참조하세요.

AWS IDE 도구 키트와 Amazon Q는 클라이언트 측 원격 측정 데이터를 수집하고 저장하여 향후 도구 AWS 키트 및 Amazon Q 릴리스와 관련된 결정을 내릴 수 있습니다. 수집된 데이터는 AWS 도구 키트 및 Amazon Q의 사용량을 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 도구 키트 및 Amazon Q 확장에서 AWS IDE 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 코드용 Amazon Q 확장](#)
- [AWS Toolkit for JetBrains 및 용 Amazon Q 플러그인 JetBrains](#)
- [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 서비스에 대한 [AWS 설명서](#) 주제를 참조하세요.

구성

AWS SDK 및 기타 AWS 개발자 도구 AWS Command Line Interface (예:AWS CLI) 를 사용하여 AWS 서비스 API와 상호 작용할 수 있습니다. 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 서버에 대한 요청을 암호화하는 데 사용됩니다. AWS 이 정보를 사용하여 ID를 AWS 확인하고 이와 관련된 권한 정책을 검색할 수 있습니다. 그러면 사용자에게 허용된 작업을 결정할 수 있습니다.
- 요청 처리 방법, 요청을 보낼 위치 (AWS 서비스 엔드포인트), 응답을 AWS CLI 해석하거나 표시하는 방법을 또는 SDK에 지시하는 데 사용하는 기타 구성 세부 정보

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.

하지만 대부분의 AWS SDK 및 도구는 코드 자체를 제외한 두 가지 기본 소스의 공통 설정을 지원합니다.

- [공유 AWS 구성 및 자격 증명 파일](#) — 공유 config 및 credentials 파일은 AWS SDK 또는 도구에 인증 및 구성을 지정할 수 있는 가장 일반적인 방법입니다. 이러한 파일을 사용하여 도구 및 애플리케이션에서 사용할 수 있는 설정을 저장할 수 있습니다. 공유 config 및 credentials 파일 내의 설정은 특정 프로파일과 연결됩니다. 프로파일이 여러 개 있는 경우 다양한 설정 구성을 만들어 다양한 시나리오에 적용할 수 있습니다. AWS 도구를 사용하여 명령을 호출하거나 SDK를 사용하여 AWS API를 호출하는 경우 해당 작업에 사용할 프로필, 즉 구성 설정을 지정할 수 있습니다. 프로파일 중 하나는 default 프로파일로 지정되며 사용할 프로파일을 명시적으로 지정하지 않으면 자동으로 이를 사용합니다. 이러한 파일에 저장할 수 있는 설정은 이 참조 안내서에 설명되어 있습니다.
- [환경 변수](#) - 일부 설정은 별도로 사용자 운영 체제의 환경 변수에 저장할 수도 있습니다. 한 번에 한 세트의 환경 변수만 적용할 수 있지만 프로그램이 실행되고 요구 사항이 변경되면 환경 변수를 능동적으로 수정할 수 있습니다.

이 섹션의 추가 주제

- [공유 config 및 credentials 파일](#)
- [공유 config 및 credentials 파일 위치](#)

- [환경 변수 지원](#)
- [JVM 시스템 속성 지원](#)

공유 config 및 credentials 파일

공유 AWS config 및 credentials 파일에는 프로파일 세트가 포함되어 있습니다. 프로파일은 키-값 페어로 구성된 구성 설정 세트로, AWS Command Line Interface (AWS CLI), 및 AWS SDKs 기타 도구에서 사용됩니다. 구성 값은 프로파일 사용될 때 SDK/도구의 일부 측면을 구성하기 위해 프로파일에 연결됩니다. 이러한 파일은 값이 사용자의 모든 애플리케이션, 프로세스 또는 SDKs 로컬 환경에 영향을 미친다는 점에서 “공유”됩니다.

공유 파일 config와 credentials 파일은 모두 ASCII 문자(UTF-8 인코딩)만 포함하는 일반 텍스트 파일입니다. 일반적으로 [INI 파일](#)이라고 하는 형식을 취합니다.

프로파일

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

[default] 프로파일에는 이름이 지정된 특정 프로파일이 지정되지 않은 경우 SDK 또는 도구 작업에서 사용하는 값이 포함됩니다. 이름으로 명시적으로 참조할 수 있는 별도의 프로파일을 만들 수도 있습니다. 각 프로파일은 애플리케이션 및 시나리오에 따라 다양한 설정과 값을 사용할 수 있습니다.

Note

[default]은(는)단순히 이름이 지정되지 않은 프로파일입니다. 사용자가 프로파일을 지정하지 않는 SDK 경우 이 프로파일의 이름은 에서 사용하는 기본 프로파일이기 default 때문에 지정됩니다. 상속된 기본값을 다른 프로파일에 제공하지 않습니다. [default] 프로파일에서 무언가를 설정하고 명명된 프로파일에서 설정하지 않으면 명명된 프로파일을 사용할 때 값이 설정되지 않습니다.

명명된 프로파일 설정

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

다음 중 하나를 설정하여 이 기능을 구성합니다.

AWS_PROFILE - 환경 변수

이 환경 변수가 명명된 프로파일 또는 “기본값”으로 설정된 경우 모든 SDK 코드와 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

SDK for Kotlin on JVM 및 SDK for Java 2.x의 경우 [aws.profile 시스템 속성](#)을 설정할 수 있습니다. 가 서비스 클라이언트를 SDK 생성할 때 코드에서 설정을 재정의하지 않는 한 명명된 프로파일의 설정을 사용합니다. SDK for Java 1.x는 이 시스템 속성을 지원하지 않습니다.

Note

애플리케이션이 여러 애플리케이션을 실행하는 서버에 있는 경우 기본 프로파일 대신 항상 이름이 지정된 프로파일을 사용하는 것이 좋습니다. 기본 프로파일은 환경의 모든 AWS 애플리케이션에서 자동으로 픽업되며 그 간에 공유됩니다. 따라서 다른 사용자가 애플리케이션의 기본 프로파일을 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. 범위만 애플리케이션에 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 프로파일입니다.

다음은 `[default]` 프로파일이 있는 기본 `config` 파일을 보여주는 예입니다. [region](#) 설정값을 설정합니다. 다른 섹션 정의가 발생할 때까지 이 줄을 따르는 모든 설정은 이 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 인 프로파일이 있는 `config` 파일을 보여줍니다. `foo`. 다른 섹션 정의가 발생할 때까지 이 줄을 따르는 모든 설정은 이 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 줄에는 를 사용하여 AWS 보안 인증 정보를 확인하도록 프로필을 구성하는데 사용하는 설정 그룹의 이름이 지정됩니다 AWS IAM Identity Center. Single Sign-On 인증 구성에 대한 자세한 내용은 [IAM SDK 또는 도구에 대한 Identity Center 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.

다음 예제에서는 'my-ssoSampleRole'의 토큰을 사용하여 '111122223333' 계정의 " IAM 역할에 대한 단기 AWS 자격 증명을 가져오는 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 지정을 지원하며 이 기능이 포함된 SDKs 및 도구에서만 사용할 수 있습니다. 에서 이 기능을 사용할 수 있는지 확인하려면 서비스별 엔드포인트와의 호환성 [AWS SDKs](#)는 섹션을 SDK참조하세요.

services 섹션 정의 라인은 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, 여기서 `<SERVICE>` 는 AWS 서비스 식별자 키입니다. AWS 서비스 식별자는 모든 공백을 밑줄로 바꾸고 모든 문자를 소문자로 표시하여 API 모델의 를 기반으로 합니다. `services` 섹션에서 사용할 모든 서비스 식별자 키 목록은 [서비스별 엔드포인트 식별자](#)을 참조하십시오. 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 `dynamodb`입니다. Amazon DynamoDB 서비스 하위 섹션은 라인에서 시작됩니다 `dynamodb =` . 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 프로파일 이름 자체만 사용합니다. 다음 예제에서는 이름이 `foo`인 프로파일 섹션이 있는 `credentials` 파일을 보여줍니다.

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

'비밀' 또는 민감한 것으로 간주되는 다음 설정만 `credentials` 파일에 저장할 수 있습니다.

`aws_access_key_id`, `aws_secret_access_key` 및 `aws_session_token`. 이러한 설정을 공유 `config` 파일에 배치할 수도 있지만 이러한 민감한 값은 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_key](#) 및 [aws_session_token](#) 전역 설정을 설정합니다.

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 위치를 변경하십시오.

다음 중 하나를 사용하여 SDK 또는 도구가 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 [을 참조하십시오. 환경 변수 지원](#)

JVM 시스템 속성 사용

JVM에서 실행되는 Kotlin용 SDK와 Java 2.x용 SDK의 경우 다음과 같은 JVM 시스템 속성을 설정하여 이러한 파일의 위치 또는 이름을 기본값에서 사용자 지정 값으로 변경할 수 있습니다.

- config파일 JVM 시스템 속성: **aws.configFile**
- credentials 파일 환경 변수:**aws.sharedCredentialsFile**

JVM 시스템 속성을 설정하는 방법에 대한 지침은 [을 참조하십시오. the section called “JVM 시스템 속성을 설정하는 방법”](#) Java 1.x용 SDK는 이러한 시스템 속성을 지원하지 않습니다.

환경 변수 지원

환경 변수는 구성 옵션과 보안 인증을 지정하는 다른 방법을 제공하며, 스크립팅을 수행하거나 명명된 프로파일을 임시로 기본값으로 설정할 때 유용할 수 있습니다. 대부분 지원되는 환경 변수 목록은 SDKs [을 참조하십시오.환경 변수 목록.](#)

옵션의 우선 순위

- 환경 변수를 사용하여 설정을 지정하면 공유 프로필에서 로드된 모든 값이 무시됩니다. AWS config및 파일. credentials
- 의 매개 변수를 사용하여 설정을 지정하는 경우 AWS CLI 명령줄은 해당 환경 변수 또는 구성 파일에 있는 프로필의 모든 값을 재정의합니다.

환경 변수를 설정하는 방법

다음은 기본 사용자에게 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...zrkuWJ0gQs8IZZaIv2BXIa2R40Lgk
$ 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 wJa1rXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
C:\> setx
AWS_SESSION_TOKEN AQoEXAMPLEH4aoAH0gNCAPy...truncated...zrkuWJ0gQs8IZZaIv2BXIa2R40Lgk
C:\> setx AWS_REGION us-west-2
```

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 셸에는 영향을 주지 않습니다.

PowerShell

```
PS C:\> $Env:AWS_ACCESS_KEY_ID="AKIAIOSFODNN7EXAMPLE"
PS C:\> $Env:AWS_SECRET_ACCESS_KEY="wJa1rXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY"
PS C:\>
PS C:\> $Env:AWS_SESSION_TOKEN="AQoEXAMPLEH4aoAH0gNCAPy...truncated...zrkuWJ0gQs8IZZaIv2BXIa2R40Lgk"
PS C:\> $Env:AWS_REGION="us-west-2"
```

이전 예와 같이 PowerShell 프롬프트에서 환경 변수를 설정하면 현재 세션 기간 동안만 값이 저장됩니다. 모든 PowerShell 세션과 명령 프롬프트 세션에서 환경 변수 설정을 유지하려면 제어판의 시스템 응용 프로그램을 사용하여 환경 변수 설정을 저장하십시오. 또는 PowerShell 프로파일에 변수를 추가하여 향후 모든 PowerShell 세션에 사용할 변수를 설정할 수 있습니다. 환경 변수를 저장하거나 여러 세션에서 유지하는 방법에 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 들어, AWS Lambda 환경 변수를 직접 설정할 수 있습니다. 자세한 내용은 [사용을 참조하십시오. AWS Lambda](#)의 환경 변수 AWS Lambda 개발자 가이드.

서버리스 프레임워크에서는 SDK 환경 설정의 공급자 키 아래에 있는 `serverless.yml` 파일에 환경 변수를 설정할 수 있는 경우가 많습니다. `serverless.yml` 파일에 대한 자세한 내용은 서버리스 프레임워크 설명서의 [일반 함수 설정](#)을 참조하십시오.

컨테이너 환경 변수를 설정하는 데 사용하는 메커니즘에 관계없이 [정의된 런타임 환경 변수](#)에 Lambda에 대해 문서화된 것과 같이 컨테이너에서 예약한 것도 있습니다. 환경 변수가 처리되는 방식과 제한 사항이 있는지 확인하려면 항상 사용 중인 컨테이너의 공식 설명서를 참조하십시오.

JVM 시스템 속성 지원

[JVM 시스템 속성](#)은 JVM에서 실행되는 SDK의 구성 옵션 및 자격 증명을 지정하는 또 다른 방법을 제공합니다 (예: 및). AWS SDK for Java AWS SDK for Kotlin [SDK에서 지원하는 JVM 시스템 속성 목록](#)은 [설정 참조](#)를 참조하십시오.

옵션의 우선 순위

- JVM 시스템 속성을 사용하여 설정을 지정하면 환경 변수에 있거나 공유 AWS config 및 파일의 프로파일에서 로드된 모든 값이 무시됩니다. `credentials`
- 환경 변수를 사용하여 설정을 지정하면 공유 AWS config 및 `credentials` 파일의 프로파일에서 로드된 모든 값이 무시됩니다.

JVM 시스템 속성을 설정하는 방법

여러 가지 방법으로 JVM 시스템 속성을 설정할 수 있습니다.

커맨드 라인에서

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

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

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

환경 변수 사용

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

다음 예제에서는 코드에서 값을 명시적으로 재정의하지 않는 한 모든 서비스 클라이언트에 대해 AWS 리전 전역적으로 구성합니다.

Linux, macOS, or Unix

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

환경 변수를 설정하면 사용되는 값이 변경되어 셸 세션이 종료될 때까지 또는 변수를 다른 값으로 설정할 때까지 유지됩니다. 셸의 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 셸에는 영향을 주지 않습니다.

런타임 시

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

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

Important

SDK 서비스 클라이언트를 초기화하기 전에 JVM 시스템 속성을 설정하십시오. 그렇지 않으면 서비스 클라이언트가 다른 값을 사용할 수 있습니다.

인증 및 액세스

를 사용하여 개발할 AWS 때 코드가 를 인증하는 방법을 설정해야 합니다 AWS 서비스. 환경 및 사용 가능한 액세스에 따라 다양한 방식으로 AWS 리소스에 대한 프로그래밍 방식 AWS 액세스를 구성할 수 있습니다.

로컬(AWS가 아닌 지역)에서 실행되는 코드의 인증 옵션

- [IAM SDK 또는 도구에 대한 Identity Center 인증](#) - 보안 모범 사례로 IAM Identity Center와 AWS Organizations 함께 를 사용하여 모든 에서 액세스를 관리하는 것이 좋습니다 AWS 계정. 에서 사용자 생성하거나 AWS IAM Identity Center, Microsoft Active Directory를 사용하거나, SAML 2.0 자격 증명 공급자(IdP)를 사용하거나, IdP를 에 개별적으로 페더레이션할 수 있습니다 AWS 계정. 리전이 IAM Identity Center를 지원하는지 확인하려면 의 [AWS IAM Identity Center 엔드포인트 및 할당량을 참조하세요](#) Amazon Web Services 일반 참조.
- [IAM Roles Anywhere](#) - IAM Roles Anywhere를 사용하여 외부에서 실행되는 서버, 컨테이너 및 애플리케이션과 같은 워크로드에 IAM 대해 에서 임시 보안 자격 증명을 얻을 수 있습니다 AWS. IAM Roles Anywhere를 사용하려면 워크로드가 X.509 인증서를 사용해야 합니다.
- [AWS 자격 증명으로 역할 수임](#) - 그렇지 않으면 액세스할 수 없는 AWS 리소스에 일시적으로 액세스 하기 위해 IAM 역할을 수임할 수 있습니다.
- [AWS 액세스 키](#) - 편리하지 않거나 AWS 리소스에 대한 보안 위험을 증가시킬 수 있는 다른 옵션.

AWS 환경 내에서 실행되는 코드에 대한 인증 옵션

코드가 에서 실행되는 경우 애플리케이션에서 AWS보안 인증 정보를 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 의 자격 증명 공급자 체인을 통해 컨테이너 내에서 실행되는 코드에서 IAM 역할에 대한 SDK자격 증명을 자동으로 가져올 수 있습니다.

- [Amazon EC2 인스턴스용 IAM 역할 사용](#) - IAM 역할을 사용하여 Amazon EC2 인스턴스에서 애플리케이션을 안전하게 실행합니다.
- 다음과 같은 방법으로 IAM Identity Center를 AWS 사용하여 프로그래밍 방식으로 상호 작용할 수 있습니다.
 - [AWS CloudShell](#) 를 사용하여 콘솔에서 AWS CLI 명령을 실행합니다.
 - 소프트웨어 개발 팀을 위한 클라우드 기반 협업 공간을 시도하려면 [Amazon CodeCatalyst](#)를 사용하는 것이 좋습니다.

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

에 액세스해야 하는 모바일 애플리케이션 또는 클라이언트 기반 웹 애플리케이션을 생성하는 경우 AWS 웹 ID 페더레이션을 사용하여 임시 AWS 보안 자격 증명을 동적으로 요청하도록 앱을 구축합니다.

웹 ID 페더레이션을 사용하면 사용자 정의 로그인 코드를 생성하거나 자신의 사용자 보안 인증을 관리할 필요가 없습니다. 대신 앱 사용자는 Amazon, Facebook, Google 또는 기타 OpenID Connect() 호환 IdP 를 사용한 로그인과 같은 잘 알려진 외부 자격 증명 공급자(IdP OIDC)를 사용하여 로그인할 수 있습니다. 인증 토큰을 받은 다음 해당 토큰을 의 리소스를 사용할 권한이 있는 IAM 역할에 매핑 AWS 된 임시 보안 자격 증명으로 교환할 수 있습니다 AWS 계정.

SDK 또는 도구에 대해 이를 구성하는 방법을 알아보려면 섹션을 참조하세요 [웹 자격 증명 또는 OpenID Connect로 역할 수입](#).

모바일 애플리케이션의 경우 Amazon Cognito를 사용하는 것이 좋습니다. Amazon Cognito는 ID 브로커로 활동하며 사용자를 대신하여 상당한 페더레이션을 합니다. 자세한 내용은 IAM 사용 설명서의 [모바일 앱용 Amazon Cognito 사용](#)을 참조하세요.

액세스 관리에 대한 추가 정보

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

- [IAM 자격 증명\(사용자, 사용자 그룹 및 역할\)](#) - 에서 자격 증명의 기본 사항을 이해합니다 AWS.
- [의 보안 모범 사례 IAM - 공동 책임 모델](#) 에 따라 AWS 애플리케이션을 개발할 때 따라야 할 보안 권장 사항입니다.

Amazon Web Services 일반 참조에는 다음에 대한 기본 사항이 있습니다.

- [AWS 보안 인증 정보 이해 및 가져오기](#) - 콘솔 및 프로그래밍 방식 액세스 모두에 대한 키 옵션 및 관리 관행에 액세스합니다.

AWS Builder ID

는 AWS 계정 이미 소유하거나 생성하려는 모든 를 AWS Builder ID 보완합니다. AWS 계정은 사용자가 생성하는 AWS 리소스의 컨테이너 역할을 하고 해당 리소스의 보안 경계를 제공하지만 는 사용자를 개인으로 AWS Builder ID 나타냅니다. 로 로그인하여 Amazon 및 Amazon 과 같은 개발자 도구 CodeWhisperer 및 서비스에 AWS Builder ID 액세스할 수 있습니다 CodeCatalyst.

- AWS 로그인 사용 설명서에서 [로 로그인 AWS Builder ID](#) - 를 생성 및 사용하는 방법과 Builder ID가 제공하는 기능을 AWS Builder ID 알아봅니다.
- CodeWhisperer 사용 설명서의 [CodeWhisperer 및 AWS Toolkit - Builder ID로 인증](#) - 가 를 CodeWhisperer 사용하는 방법을 알아봅니다 AWS Builder ID.
- [CodeCatalyst 개념 - AWS Builder ID](#) Amazon CodeCatalyst 사용 설명서 - 가 를 CodeCatalyst 사용하는 방법을 알아봅니다 AWS Builder ID.

IAM SDK 또는 도구에 대한 Identity Center 인증

AWS IAM Identity Center 는 컴퓨팅이 아닌AWS 서비스에서 개발할 때 자격 AWS 증명을 제공하는 권장 방법입니다. 예를 들어 이는 사용자의 로컬 개발 환경과 같은 것입니다. Amazon Elastic Compute Cloud(AmazonEC2) 또는와 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. 의 형식은 `d-xxxxxxxxxx.awsapps.com/start`또는 URL입니다 `your_subdomain.awsapps.com/start`. AWS 액세스 포털에 로그인하면 해당 사용자에 대해 구성된 AWS 계정 및 역할을 볼 수 있습니다. 이 절차에서는 AWS 액세스 포털을 사용하여 SDK/tool 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.

IAM Identity Center를 사용하여 프로그래밍 방식 액세스 구성

1 단계: 액세스 설정 및 적절한 권한 세트 선택

IAM Identity Center를 아직 활성화하지 않은 경우 AWS IAM Identity Center 사용 설명서의 [IAM Identity Center 활성화](#)를 참조하세요.

자격 AWS 증명에 액세스하려면 다음 방법 중 하나를 선택합니다.

IAM Identity Center를 통해 액세스를 설정하지 않았습니다.

1. 사용 AWS IAM Identity Center 설명서의 [기본 IAM Identity Center 디렉토리를 사용하여 사용자 액세스 구성](#) 절차에 따라 사용자를 추가하고 관리 권한을 추가합니다.

2. AdministratorAccess 권한 세트는 정기적인 개발에 사용해서는 안 됩니다. 대신 고용주가이 용도로 사용자 지정 PowerUserAccess 권한 세트를 생성하지 않은 한 사전 정의된 권한 세트를 사용하는 것이 좋습니다.

[기본 IAM Identity Center 디렉터리 절차를 다시 사용하여 동일한 사용자 액세스 구성 절차를 따르](#) 되 이번에는 다음과 같이 합니다.

- *Admin team* 그룹을 생성하는 대신 *Dev team* 그룹을 생성하고 그 후 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 완료하면 다음 사항이 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 확인합니다PowerUserAccess. 도구/로 작업할 PowerUserAccess 때를 선택합니다SDK.

고용주가 관리하는 페더레이션 자격 증명 공급자(예: Microsoft Entra 또는 Okta)를 AWS 통해에 이미 액세스할 수 있음

자격 증명 공급자의 포털을 AWS 통해에 로그인합니다. 클라우드 관리자가 PowerUserAccess (개발자) 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 문의하십시오.

고용주가 관리하는 액세스 포털을 AWS 통해에 AWS 이미 액세스할 수 있음

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.

고용주가 관리하는 페더레이션 사용자 지정 자격 증명 공급자를 AWS 통해 이미 액세스할 수 있음
IT 팀에 문의하십시오.

2단계: IAM Identity Center를 사용하도록 SDKs 및 도구 구성

1. 사용자 개발 시스템에 최신 AWS CLI를 설치합니다.
 - a. 자세한 내용은 AWS Command Line Interface 사용 설명서의 [AWS CLI 최신 버전의 설치 또는 업데이트](#)를 참조하십시오.
 - b. (선택 사항) AWS CLI 이 작동하는지 확인하려면 명령 프롬프트를 열고 `aws --version` 명령을 실행합니다.
2. AWS 액세스 포털에 로그인합니다. 고용주가 이를 제공하거나 1단계: 액세스 설정 다음에 이메일로 받을 URL 수 있습니다. 그렇지 않은 경우의 대시보드에서 AWS 액세스 포털 URL을 찾습니다 <https://console.aws.amazon.com/singlesignon/>.
 - a. AWS 액세스 포털의 계정 탭에서 관리할 개별 계정을 선택합니다. 사용자의 역할이 표시됩니다. 액세스 키를 선택하여 적절한 권한 세트에 대한 명령줄 또는 프로그래밍 방식 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 자격 증명 방법을 선택하여 다음 단계에 필요한 Issuer URL 및 SSO Region 값을 가져옵니다. 참고:와 호환하여 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. 프롬프트에 대해 얻은 값을 SSO Start URL 입력합니다 Issuer URL.
 - b. CLI 프로필 이름의 경우 시작할 `default` 때를 입력하는 것이 좋습니다. 기본이 아닌 (명명된) 프로필 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 구성된 IAM Identity Center 권한 세트가 표시되어야 합니다.

5. 를 사용하는 경우 개발 환경에서 SDK에 대한 애플리케이션을 AWS SDK생성합니다.
 - a. 일부의 경우 IAM Identity Center 인증을 사용하려면 SSO 및와 같은 SDKs추가 패키지를 애플리케이션에 추가해야 SSOIDC 합니다. 자세한 내용은 특징를 참조하세요SDK.
 - b. 이전에에 대한 액세스를 구성한 경우 공유 AWS credentials 파일에서를 AWS검토합니다AWS 액세스 키. SDK 또는 도구가 [자격 증명 공급자 체인 이해](#) 우선 순위로 인해 IAM Identity Center 자격 증명을 사용하려면 정적 자격 증명을 제거해야 합니다.

SDKs 및 도구가이 구성을 사용하여 자격 증명을 사용하고 새로 고치는 방법에 대한 자세한 내용은 섹션을 참조하세요IAM Identity Center 인증 이해하기.

구성된 세션 길이에 따라 액세스가 결국 만료되고 SDK 또는 도구에 인증 오류가 발생합니다. 필요할 때 액세스 포털 세션을 다시 새로 고치려면 AWS CLI 를 사용하여 `aws sso login` 명령을 실행합니다.

IAM Identity Center 액세스 포털 세션 기간과 권한 세트 세션 기간을 모두 확장할 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 기간 구성](#)
- 권한 설정 세션 기간 - 세션 기간을 [설정합니다](#)

SDKs 및 도구에 대한 모든 IAM Identity Center 공급자 설정에 대한 자세한 내용은이 설명서IAM Identity Center 자격 증명 공급자의 섹션을 참조하세요.

IAM Identity Center 인증 이해하기

IAM Identity Center 관련 용어

다음 용어는 기본 AWS IAM Identity Center 프로세스와 구성을 이해하는 데 도움이 됩니다. AWSSDK API 설명서에서는 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	Description
Identity Center	sso	AWS싱글 사인 온으로 이름이 변경되더라도 sso API 이름 공간에는 이전 버전과의 호환성을 위해 원래 이름을 유지합니다. 자세한 내용은 AWS IAM Identity Center 사용 설명서의 IAM Identity Center 이름 변경 을 참조하세요.
IAM Identity Center 콘솔 관리자 콘솔		싱글 사인온을 구성하는 데 사용하는 콘솔입니다.
AWS 액세스 포털 URL		사용자 IAM ID 센터 계정의 고유한 URL (예https://xxx.awsapps.com/start) 사용자 IAM ID 센터 로그인 보안 인증을 사용하여 이 포털에 로그인합니다.
IAM Identity Center 액세스 포털 세션	인증 세션	발신자에게 베어러 액세스 토큰을 제공합니다.
권한 집합 세션		SDK가 내부적으로 AWS 서비스호출에 사용하는 IAM 세션입니다. 비공식 토론에서는 이를 “역할 세션”이라고 잘못 지칭하는 것을 볼 수 있습니다.
권한 세트 보안 인증	AWS 보안 인증 sigv4 보안 인증	SDK가 대부분의 AWS 서비스 호출 (특히, 모든 AWS 서비스 sigv4 호출) 에 실제로 사용하는 보안 인증입니다. 비공식 토론에서는 이를 “역할 보안 인

IAM Identity Center 이름	SDK API 이름	Description
		증"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

AWS 서비스에 대한 SDK 보안 인증 확정을 이해하십시오.

IAM ID 센터 API는 베어러 토큰 보안 인증을 sigv4 보안 인증으로 교환합니다. Amazon CodeWhisperer와 Amazon CodeCatalyst 같은 몇 가지 예외를 제외하면 대부분 AWS 서비스은 sigv4 API입니다. 다음은 AWS IAM Identity Center를 통한 사용자 애플리케이션 코드에 대한 대부분의 AWS 서비스 호출을 지원하기 위한 보안 인증 확정 프로세스를 설명합니다.

AWS 액세스 포털 세션 시작

- 사용자 보안 인증으로 세션에 로그인하여 프로세스를 시작하십시오.
 - AWS Command Line Interface(AWS CLI) 에 있는 `aws sso login` 명령을 사용합니다. 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 또한 새 액세스 토큰과 새로 고침 토큰으로 SSO 캐시 JSON 파일을 업데이트하여 SDK에서 사용할 수 있도록 합니다.
- 이미 활성 세션이 있는 경우 이 AWS CLI 명령은 기존 세션을 재사용하며 기존 세션이 종료될 때 종료됩니다. IAM Identity Center 세션 기간 설정 방법을 알아보려면 AWS IAM Identity Center 사용 설명서의 [사용자의 AWS 액세스 포털 세션 기간 구성](#)을 참조하십시오.
 - 자주 로그인해야 하는 필요성을 줄이기 위해 최대 세션 길이가 90일로 연장되었습니다.

SDK가 AWS 서비스호출 보안 인증을 가져오는 방법

SDK는 사용자가 서비스 별로 클라이언트 객체를 인스턴스화 하는 시점에 AWS 서비스에 대한 액세스를 제공합니다. 공유 AWS config 파일의 선택된 프로필이 IAM Identity Center 보안 인증 확인으로 구성된 경우 IAM Identity Center를 사용하여 애플리케이션의 보안 인증을 확인합니다.

- [보안 인증 확인 프로세스](#)는 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 설정 세션이라고 하며 IAM 역할을 맡아 SDK에 AWS대한 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 설정](#)을 참조하십시오.
- 대부분의 AWS SDK API 설명서에서는 권한 집합 보안 인증을 보안 인증 및 sigv4 보안 인증이라고도 합니다.

IAM ID 센터 API의 [GetRole Credentials](#)를 호출하면 권한 세트 보안 인증이 SDK로 반환됩니다. SDK의 클라이언트 객체는 위임된 IAM 역할을 사용하여 AWS 서비스를 호출합니다. 예를 들어 Amazon S3에게 계정의 버킷을 나열하도록 요청합니다. 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 구성](#)를 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 세션 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 Anywhere

IAM Roles Anywhere를 사용하여 AWS 외부에서 실행되는 서버, 컨테이너 및 애플리케이션과 같은 워크로드에 대한 IAM의 임시 보안 인증을 획득할 수 있습니다. 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 AWS 외부에서 실행되는 워크로드 또는 프로세스에 대한 임시 자격 증명을 얻을 수 있는 방법을 제공합니다. 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.

트러스트 앵커, IAM 역할 및 IAM Roles Anywhere 프로필 설정 단계는 IAM Roles Anywhere 사용 설명서의 [AWS Identity and Access Management Roles Anywhere의 트러스트 앵커 및 프로필 생성](#)을 참조하십시오.

Note

IAM Roles Anywhere 사용 설명서의 프로필은 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.

보안 인증 도우미 도구를 다운로드하는 방법에 대한 지침은 IAM Roles Anywhere 사용 설명서의 [AWS Identity and Access Management Roles Anywhere에서 임시 보안 인증 획득](#)을 참조하십시오.

IAM Roles Anywhere의 임시 보안 인증을 AWS SDK 및 AWS CLI 함께 사용하려면 공유 AWS config 파일에서 `credential_process` 설정을 구성하면 됩니다. SDK와 AWS CLI은 인증에 사용하는 프로

세스 보안 인증 공급자를 `credential_process` 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 제공자](#)를 참조하십시오.

AWS 자격 증명으로 역할 수입

역할 수입에는 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 토큰

큰으로 구성됩니다. AWS Security Token Service (AWS STS) API 요청에 대한 자세한 내용은 AWS Security Token Service API 참조의 [작업을](#) 참조하세요.

역할을 수입하도록 SDK 또는 도구를 설정하려면 먼저 수입할 특정 역할을 생성하거나 식별해야 합니다. IAM 역할은 Amazon 리소스 이름() 역할로 고유하게 식별됩니다 [ARN](#). 역할은 다른 엔티티와 신뢰 관계를 구축합니다. 역할을 사용하는 신뢰할 수 있는 엔티티는 AWS 서비스 또는 다른 일 수 있습니다 AWS 계정. IAM 역할에 대한 자세한 내용은 IAM 사용 설명서의 [IAM 역할 사용](#)을 참조하세요.

IAM 역할이 식별되면 해당 역할에서 신뢰할 수 있는 경우 역할에서 부여한 권한을 사용하도록 SDK 또는 도구를 구성할 수 있습니다.

Note

가능하면 리전 엔드포인트를 사용하고 를 구성하는 것이 AWS 가장 좋습니다 [AWS 리전](#).

IAM 역할 수입

역할을 수입할 때는 임시 보안 자격 증명 세트를 AWS STS 반환합니다. 이러한 보안 인증은 다른 프로 파일이나 코드가 실행되는 인스턴스 또는 컨테이너에서 제공됩니다. 가장 일반적으로 이러한 유형의 역할 가정은 한 계정에 대한 AWS 자격 증명에 있지만 애플리케이션에 다른 계정의 리소스에 대한 액세스 권한이 필요한 경우에 사용됩니다.

1단계: IAM 역할 설정

역할을 수입하도록 SDK 또는 도구를 설정하려면 먼저 수입할 특정 역할을 생성하거나 식별해야 합니다. IAM 역할은 역할 를 사용하여 고유하게 식별됩니다 [ARN](#). 역할은 일반적으로 계정 내에서 또는 크 로스 계정 액세스를 위해 다른 엔티티와 신뢰 관계를 구축합니다. 이를 설정하려면 IAM 사용 설명서의 [IAM 역할 생성](#)을 참조하세요.

2단계: SDK 또는 도구 구성

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

Amazon ECS 컨테이너, Amazon EC2 인스턴스 또는 환경 변수에서 보안 인증을 소 싱 `credential_source`하는 데 사용합니다.

다른 프로파일에서 보안 인증을 소싱하는 데 `source_profile`을(를) 사용합니다.

`source_profile`은 또한 수입된 역할을 사용하여 다른 역할을 수입하는 프로파일 계층 구조인 역할 체인을 지원합니다.

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

- `role_arn` - 1단계에서 생성한 IAM 역할에서
- `source_profile` 또는 `credential_source` 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 수입 옵션 모두의 구성을 보여줍니다.

```
role_arn = arn:aws:iam::<123456789012>:role/my-role-name
source_profile = profile-name-with-user-that-can-assume-role
```

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

모든 역할 수입 보안 인증 공급자 설정에 대한 자세한 내용은 이 안내서의 [역할 보안 인증 제공자 수입](#)를 참조하세요.

웹 자격 증명 또는 OpenID Connect로 역할 수입

역할 수입에는 액세스 권한이 없을 수 있는 AWS 리소스에 액세스하기 위해 일련의 임시 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 대한 자세한 내용은 AWS Security Token Service API 참조의 [작업을](#) 참조하세요.

역할을 수입하도록 SDK 또는 도구를 설정하려면 먼저 수입할 특정 역할을 생성하거나 식별해야 합니다. IAM 역할은 Amazon 리소스 이름([ARN](#)) 역할로 고유하게 식별됩니다. 역할은 다른 엔티티와 신뢰 관계를 구축합니다. 역할을 사용하는 신뢰할 수 있는 엔티티는 웹 자격 증명 공급자 또는 OpenID Connect(OIDC) 또는 SAML 페더레이션일 수 있습니다. IAM 역할에 대한 자세한 내용은 IAM 사용 설명서의 [역할 수입 방법](#)을 참조하세요.

에서 IAM 역할을 구성한 후 해당 역할이 자격 증명 공급자를 신뢰하도록 구성된 SDK 경우 임시 AWS 자격 증명을 가져오기 위해 해당 역할을 수임 SDK하도록 추가로 구성할 수 있습니다.

Note

가능하면 리전 엔드포인트를 사용하고를 구성하는 것이 AWS 좋습니다 [AWS 리전](#).

웹 자격 증명 또는 OpenID Connect와 페더레이션

Login With Amazon, Facebook, Google과 같은 퍼블릭 자격 증명 공급자의 JSON 웹 토큰(JWTs)을 사용하여 사용하여 임시 AWS 자격 증명을 가져올 수 있습니다 `AssumeRoleWithWebIdentity`. 사용 방법에 따라 ID 토큰 또는 액세스 토큰이라고 할 JWTs 수 있습니다. EntraID 또는와 같이 OIDC의 검색 프로토콜과 호환되는 자격 증명 공급자(IdPs)에서 JWTs 발급한를 사용할 수도 있습니다 `PingFederate`.

Amazon Elastic Kubernetes Service를 사용하는 경우 이 기능은 Amazon EKS 클러스터의 각 서비스 계정에 대해 서로 다른 IAM 역할을 지정하는 기능을 제공합니다. 이 Kubernetes 기능은 포드 JWTs에 배포되며, 그러면 이 자격 증명 공급자가 임시 AWS 자격 증명을 얻는 데 사용합니다. 이 Amazon EKS 구성에 대한 자세한 내용은 Amazon EKS 사용 설명서의 [IAM 서비스 계정 역할을](#) 참조하세요. 그러나 더 간단한 옵션을 위해에서 [SDK 지원하는](#) 경우 [Amazon EKS Pod 자격 증명을](#) 사용하는 것이 좋습니다.

1단계: 자격 증명 공급자 및 IAM 역할 설정

외부 IdP를 사용하여 페더레이션을 구성하려면 IAM 자격 증명 공급자를 사용하여 외부 IdP 및 해당 구성에 AWS 대해 알립니다. 이렇게 하면 AWS 계정 와 외부 IdP 간의 신뢰가 설정됩니다. 인증에 JSON 웹 토큰(JWT)을 사용하도록 SDK를 구성하기 전에 먼저 자격 증명 공급자(IdP)와 액세스에 사용되는 IAM 역할을 설정해야 합니다. 이를 설정하려면 IAM 사용 설명서의 [웹 자격 증명 또는 OpenID Connect Federation\(콘솔\)에 대한 역할 생성을](#) 참조하세요.

2단계: SDK 또는 도구 구성

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

프로필에서 이를 지정하면 SDK 또는 도구가 자동으로 해당 AWS STS `AssumeRoleWithWebIdentity` API 호출을 수행합니다. 웹 자격 증명 페더레이션을 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 수입하는 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. 그러나 Amazon Cognito 자격 증명 공급자는 다른 자격 증명 공급자와 같은 SDKs 및 도구 코어 라이브러리에 포함되지 않습니다. Amazon Cognito에 액세스하려면 SDK 또는 도구의 빌드 또는 라이브러리에 Amazon Cognito 서비스 클라이언트를 API포함합니다. 에서 사용하려면 Amazon Cognito 개발자 안내서의 [코드 예제](#)를 AWS SDKs참조하세요.

모든 역할 수입 보안 인증 공급자 설정에 대한 자세한 내용은 이 안내서의 [역할 보안 인증 제공자 수입](#)를 참조하세요.

AWS 액세스 키

단기 보안 인증 정보를 사용합니다

연장된 세션 기간 옵션 사용에 [IAM SDK 또는 도구에 대한 Identity Center 인증](#)을 사용하도록 SDK 또는 도구를 구성하는 것이 좋습니다.

하지만 SDK 또는 도구의 임시 보안 인증을 직접 설정하려면 [단기 보안 인증을 사용한 인증](#)을 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 Center](#)과 같은 보안 인증 공급자를 통한 페더레이션을 사용하십시오.

전체 액세스 관리 AWS 계정

보안 모범 사례로서, IAM Identity Center를 AWS Organizations 사용하여 모든 사용자의 AWS 계정 액세스를 관리하는 것이 좋습니다. 자세한 내용은 IAM 사용 설명서의 [IAM의 보안 모범 사례](#)를 참조하십시오.

IAM ID 센터에서 사용자를 만들거나, Microsoft Active Directory를 사용하거나, SAML 2.0 ID 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. 또한 멀티 팩터 인증 (MFA) 을 적용하고 액세스를 위한 임시 자격 증명을 사용할 수 있습니다. AWS 계정 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

처음 사용하는 경우 테스트 IAM 사용자를 만든 다음 이를 사용하여 자습서를 실행하고 제공되는 기능을 탐색할 수 있습니다. AWS AWS 학습 중에는 이러한 유형의 보안 인증 정보를 사용해도 괜찮지만 샌드박스 환경 밖에서는 사용하지 않는 것이 좋습니다.

다음과 같은 사용 사례에서는 IAM 사용자와 함께 시작하는 것이 합리적일 수 있습니다. AWS

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

이러한 사용 사례 이외의 IAM 사용자를 사용하는 경우 가능한 한 빨리 IAM Identity Center로 전환하거나 ID 공급자를 페더레이션하십시오. AWS 계정 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 교체](#)에 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-secrets](#)와 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 보안 인증을 사용한 인증](#)을 참조하십시오.

단기 보안 인증을 사용한 인증

세션 기간 연장 [IAM SDK 또는 도구에 대한 Identity Center 인증](#) 옵션과 함께 사용할 수 있도록 SDK 또는 도구를 구성하는 것이 좋습니다. 하지만 에서 사용할 수 있는 임시 자격 증명은 복사하여 사용할 수 있습니다. AWS 포털에 접속하세요. 보안 인증이 만료되면 새 보안 인증을 복사해야 합니다. 프로파일 에서 임시 보안 인증을 사용하거나 이를 시스템 속성 및 환경 변수의 값으로 사용할 수 있습니다.

모범 사례: 자격 증명 파일의 액세스 키와 토큰을 수동으로 관리하는 대신 애플리케이션에서 다음에서 전달된 임시 자격 증명을 사용하는 것이 좋습니다.

- 원래 요청 ping에 대한 AWS 컴퓨팅 서비스 (예: Amazon Elastic Compute Cloud 또는 내부에서 애플리케이션 실행) AWS Lambda.
- 자격 증명 공급자 체인의 또 다른 옵션 (예 [IAM SDK 또는 도구에 대한 Identity Center 인증](#)).
- 또는 를 [프로세스 보안 인증 제공자](#) 사용하여 임시 자격 증명을 검색할 수 있습니다.

에서 검색한 단기 자격 증명을 사용하여 자격 증명 파일을 설정합니다. AWS 액세스 포털

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>
```


IAM사용자를 사용하여 코드를 실행하는 경우 개발 환경의 SDK or 도구는 공유된 코드의 장기 IAM 사용자 자격 증명을 사용하여 인증합니다. AWS credentials파일. IAM주제의 [보안 모범 사례를 검토하고 가능한 한](#) 빨리 IAM Identity Center 또는 기타 임시 자격 증명으로 전환하십시오.

보안 인증에 대한 중요 경고 및 지침

보안 인증에 대한 경고

- 액세스하려면 계정의 루트 자격 증명을 NOT 사용하세요. AWS 있습니다. 이 보안 인증은 계정 액세스에 제한이 없고 취소하기 어렵습니다.
- 애플리케이션 파일에 리터럴 액세스 키 또는 자격 증명 정보를 NOT 입력하세요. 이를 어기는 경우, 예를 들어 프로젝트를 퍼블릭 리포지토리에 업로드하면 뜻하지 않게 보안 인증이 노출될 위험이 있습니다.
- 프로젝트 영역에 자격 증명 정보가 포함된 파일을 NOT 포함하세요.
- 모든 자격 증명 정보는 공유에 저장되어 있다는 점에 유의하세요. AWS credentials파일은 일반 텍스트로 저장됩니다.

보안 인증 정보를 안전하게 관리하기 위한 추가 지침

안전한 관리 방법에 대한 일반적인 설명 AWS 자격 증명 관리 [모범 사례를 참조하십시오](#). AWS의 액세스 키 [AWS 일반 참조](#). 이 논의 외에도 다음 사항도 고려해 보십시오.

- Amazon 엘라스틱 컨테이너 서비스 (AmazonECS) [작업을 위한 작업에는 IAM 역할을](#) 사용하십시오.
- Amazon EC2 인스턴스에서 실행되는 애플리케이션의 [IAM 역할을](#) 사용하십시오.

사전 요구 사항: 생성 AWS account

IAM사용자를 사용하여 액세스하려면 AWS 서비스에는 다음이 필요합니다. AWS 계정 및 AWS 자격 증명.

1. 계정을 생성합니다.

생성하려면 AWS 계정을 참조하십시오. [시작하기: 처음이신가요? AWS 사용자이신가요?](#) 에서 AWS Account Management 참조 가이드.

2. 관리 사용자를 생성합니다.

관리 콘솔 및 서비스에 액세스하기 위해 루트 사용자 계정(사용자가 생성하는 초기 계정)을 사용하지 않습니다. 대신 사용 설명서의 관리 사용자 [만들기에 설명된 대로 관리자](#) 계정을 만드십시오. IAM

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

이 두 계정 모두 다음을 개발하는 데 적합하지 않습니다. AWS 또는 에서 애플리케이션을 실행하는 데 사용할 수도 있습니다. AWS. 가장 좋은 방법은 이러한 작업에 적합한 사용자, 권한 집합 또는 서비스 역할을 만드는 것입니다. 자세한 내용은 사용 [설명서의 최소 권한 적용을](#) 참조하십시오. IAM

1단계: 사용자 생성 IAM

- IAM사용 설명서의 사용자 [생성 IAM \(콘솔\)](#) 절차에 따라 IAM사용자를 생성합니다. IAM사용자를 생성할 때:
 - 에 대한 사용자 액세스 제공을 선택하는 것이 좋습니다. AWS Management Console. 이렇게 하면 볼 수 있습니다. AWS 서비스 시각적 환경에서 실행 중인 코드 (예: 검사) 와 관련이 있습니다. AWS CloudTrail 진단 로그를 작성하거나 Amazon 심플 스토리지 서비스에 파일을 업로드하면 코드를 디버깅할 때 유용합니다.
 - 권한 설정 - 권한 옵션에서 이 사용자에게 권한을 할당하려는 방법에 대해 직접 정책 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 관련된 해당 절차의 선택적 단계는 무시해도 됩니다.

2단계: 액세스 키 가져오기

1. IAM콘솔의 탐색 창에서 사용자를 선택한 다음 이전에 만든 사용자의 이름을 선택합니다. **User name**
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 동일한 유형의 키를 생성하여 두 옵션 모두에 사용할 수 있습니다. AWS CLI 그리고SDKs.

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. 공유를 생성하거나 엽니다. AWS credentials 파일. 이 파일은 Linux 및 macOS 시스템의 경우 ~/.aws/credentials이며, Windows의 경우 %USERPROFILE%\aws\credentials입니다. 자세한 내용은 [보안 인증 파일 위치](#)를 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 자격 증명으로 전환하는 것이 좋습니다. 장기 보안 인증을 사용하지 않도록 전환한 후에는 공유 credentials 파일에서 해당 보안 인증을 삭제해야 합니다.

Amazon EC2 인스턴스용 IAM 역할 사용

이 예제에서는 Amazon EC2 인스턴스에 배포된 애플리케이션에서 사용할 Amazon S3 액세스 권한을 가진 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 부여하십시오. 자세한 내용은 [내용은 Amazon EC2 EC2 사용 설명서의 Amazon IAM 역할을 참조하십시오.](#)

IAM 역할 생성

개발하는 AWS SDK 애플리케이션은 작업을 수행하기 AWS 서비스 위해 하나 이상의 애플리케이션에 액세스할 가능성이 높습니다. 애플리케이션을 실행하는 데 필요한 권한을 부여하는 IAM 역할을 생성하십시오.

이 절차는 예를 들어 Amazon S3에 대한 읽기 전용 액세스 권한을 부여하는 역할을 생성합니다. 대부분의 AWS SDK 안내서에는 Amazon S3에서 읽을 수 있는 “시작하기” 자습서가 있습니다.

1. 에 AWS Management Console 로그인하고 에서 IAM <https://console.aws.amazon.com/iam/> 콘솔을 여십시오.
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 EC2 Amazon을 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 필요하므로 이 이름을 기억해 두십시오.

Amazon EC2 인스턴스를 시작하고 IAM 역할을 지정합니다.

다음과 같이 IAM 역할을 사용하여 Amazon EC2 인스턴스를 생성하고 시작할 수 있습니다.

- Amazon EC2 사용 설명서에서 [인스턴스를 빠르게 시작하는](#) 방법을 따르십시오. 하지만 최종 제출 단계에 앞서 다음 작업도 수행하십시오.
 - 고급 세부 정보에서 IAM 인스턴스 프로파일의 경우 이전 단계에서 만든 역할을 선택합니다.

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

EC2인스턴스에 연결

Amazon EC2 인스턴스에 연결하여 애플리케이션을 Amazon 인스턴스로 전송한 다음 애플리케이션을 실행할 수 있습니다. 인스턴스를 생성할 때 Key pair (로그인) 에서 사용한 키 쌍의 비공개 부분, 즉 파일이 들어 있는 PEM 파일이 필요합니다.

인스턴스 유형에 대한 지침 ([Linux 인스턴스에 연결 또는 Windows 인스턴스에 연결](#)) 에 따라 이 작업을 수행할 수 있습니다. 연결할 때는 개발 머신에서 인스턴스로 파일을 전송할 수 있는 방식으로 연결합니다.

Note

Linux 또는 macOS 터미널에서는 보안 복사 명령을 사용하여 애플리케이션을 복사할 수 있습니다. 키 scp 페어와 함께 사용하려면 다음 명령을 사용할 수 `scp -i path/to/key file/to/copy ec2-user@ec2-xx-xx-xxx-xxx.compute.amazonaws.com:~` 있습니다. Windows에 대한 자세한 내용은 [Windows 인스턴스로 파일 전송](#)을 참조하십시오.

툴킷을 사용하는 경우 AWS 툴킷을 사용하여 인스턴스에 연결할 수도 있는 경우가 많습니다. 자세한 내용은 사용하는 툴킷의 특정한 사용 설명서를 참조하십시오.

인스턴스에서 애플리케이션을 실행합니다. EC2

1. 로컬 드라이브에서 Amazon EC2 인스턴스로 애플리케이션 파일을 복사합니다.
2. 애플리케이션을 시작하고 개발 머신에서와 동일한 결과로 실행되는지 확인합니다.
3. (선택 사항) 애플리케이션이 IAM 역할에서 제공한 자격 증명을 사용하는지 확인하십시오.
 - a. 에서 Amazon EC2 콘솔에 AWS Management Console 로그인하고 엽니다 <https://console.aws.amazon.com/ec2/>.
 - b. 인스턴스를 선택합니다.
 - c. 작업, 보안을 선택한 다음 IAM역할 수정을 선택합니다.
 - d. IAM역할의 경우 역할 없음을 IAM 선택하여 IAM 역할을 분리합니다.
 - e. IAM역할 업데이트를 선택합니다.
 - f. 애플리케이션을 다시 실행하고 인증 오류가 반환되는지 확인합니다.

설정 참조

SDKs 는 APIs에 대한 언어별 를 제공합니다 AWS 서비스. 인증, 재시도 동작 등을 포함하여 성공적으로 API 전화를 거는 데 필요한 몇 가지 헤비 리프팅을 처리합니다. 이렇게 하려면 SDKs에 요청에 사용할 자격 증명을 얻고, 각 서비스에 사용할 설정을 유지하고, 전역 설정에 사용할 값을 얻는 유연한 전략이 있습니다.

구성 설정에 대한 자세한 내용은 다음 섹션에서 확인할 수 있습니다.

- [AWS SDKs 및 도구 표준화된 자격 증명 공급자](#) - 여러 에 걸쳐 표준화된 공통 자격 증명 공급자입니다 SDKs.
- [AWS SDKs 및 도구 표준화 기능](#) - 여러 에 걸쳐 표준화된 공통 기능 SDKs.

서비스 클라이언트 생성

프로그래밍 방식으로 에 액세스하려면 각 에 대해 클라이언트 클래스/객체를 AWS 서비스 SDKs 사용합니다 AWS 서비스. 예를 들어 애플리케이션이 Amazon 에 액세스해야 하는 경우 EC2 애플리케이션은 해당 서비스와 인터페이스할 Amazon EC2 클라이언트 객체를 생성합니다. 그런 다음 서비스 클라이언트를 사용하여 요청을 AWS 서비스에 보내면 됩니다. 대부분의 에서 SDKs 서비스 클라이언트 객체는 변경할 수 없으므로 요청을 하는 각 서비스와 다른 구성을 사용하여 동일한 서비스에 대한 요청을 할 새 클라이언트를 생성해야 합니다.

설정의 우선 순위

전역 설정은 대부분의 에서 지원되고 에서 광범위한 영향을 미치는 기능, 자격 증명 공급자 SDKs 및 기타 기능을 구성합니다 AWS 서비스. 모든 SDKs 에는 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 또는 의 경우 AWS Tools for PowerShell 명령줄에 입력하는 작업당 파라미터의 형태를 취합니다. 의 경우 SDK 명시적 할당은 AWS 서비스 클라이언트 또는 구성 객체를 인스턴스화할 때 또는 때로는 개별 를 호출할 때 설정한 파라미터의 형태를 취할 수 있습니다 API.
2. Java/Kotlin만 해당: 설정에 대한 JVM 시스템 속성이 확인됩니다. 설정이 되면 클라이언트 구성에 해당 값이 사용됩니다.
3. 환경 변수를 확인합니다. 설정이 되면 클라이언트 구성에 해당 값이 사용됩니다.

4. 는 공유 credentials 파일의 설정을 SDK 확인합니다. 설정하면 클라이언트가 이를 사용합니다.
5. 설정에 대한 공유 config 파일입니다. 설정이 있는 경우 는 설정을 SDK 사용합니다.
 - AWS_PROFILE 환경 변수 또는 aws.profile JVM 시스템 속성을 사용하여 SDK 로드할 프로파일을 지정할 수 있습니다.
6. SDK 소스 코드 자체에서 제공하는 모든 기본값은 마지막으로 사용됩니다.

Note

일부 SDKs 및 도구는 다른 순서로 확인할 수 있습니다. 또한 일부 SDKs 및 도구는 파라미터를 저장하고 검색하는 다른 방법을 지원합니다. 예를 들어는 [SDK 스토어](#) 라는 추가 소스를 AWS SDK for .NET 지원합니다. SDK 또는 도구에 고유한 공급자에 대한 자세한 내용은 사용 중인 SDK 또는 도구에 대한 특정 가이드를 참조하세요.

순서에 따라 어떤 방법이 다른 메서드보다 우선 적용되는지 결정됩니다. 예를 들어 공유 config 파일에서 프로파일을 설정하는 경우 SDK 또는 도구가 다른 위치를 먼저 확인한 후에만 검색되고 사용됩니다. 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 재정의합니다. 마지막으로 개별 작업(AWS CLI 명령줄 파라미터 또는 API 파라미터) 또는 코드의 설정은 해당 명령에 대한 다른 모든 값을 재정의합니다.

설정 페이지

이 가이드의 설정 참조 섹션 내 페이지에는 다양한 메커니즘을 통해 설정할 수 있는 사용 가능한 설정이 자세히 나와 있습니다. 다음 표에는 구성 및 보안 인증 파일 설정, 환경 변수 및 (Java 및 Kotlin의 경우 SDKs) 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 나열된 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 설정에는 설명 세부 정보 위에 블록에서 값을 설정하는 지원되는 모든 방법이 나열됩니다. [우선 순위](#)는 다르지만 결과 기능은 설정 방식에 관계없이 동일합니다.

설명에는 아무것도 하지 않으면 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 전달합니다.

- 코드베이스 외부에서 요청 압축을 제어하는 세 가지 동등한 방법이 있습니다. 다음 작업 중 하나를 수행할 수 있습니다.
 - 를 사용하여 구성 파일에 설정 `disable_request_compression`
 - 를 사용하여 환경 변수로 설정 `AWS_DISABLE_REQUEST_COMPRESSION`
 - 또는 Java 또는 Kotlin을 사용하는 경우 를 사용하여 JVM 시스템 속성으로 SDK 설정합니다. `aws.disableRequestCompression`

Note

코드에서 직접 동일한 기능을 구성하는 방법도 있을 수 있지만, 이 참조는 각 에 고유하기 때문에 이 문제를 다루지 않습니다. SDK 코드 자체에서 구성을 설정하려면 특정 SDK 가이드 또는 API 참조를 참조하세요.

- 아무것도 하지 않으면 값은 기본적으로 로 설정됩니다 `false`.
- 이 부울 설정에 유효한 유일한 값은 `true` 및 `false`입니다.

각 기능 페이지 하단에는 테이블과의 호환성 AWS SDKs이 있습니다.

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

- Yes - 작성된 SDK 대로 설정은 에서 완전히 지원됩니다.
- Partial - 일부 설정이 지원되거나 동작이 설명과 다릅니다. Partial의 경우 추가 참고는 편차를 나타냅니다.
- No - 지원되는 설정은 없습니다. 이는 코드에서 동일한 기능을 달성할 수 있는지 여부에 대한 주장을 하지 않습니다. 나열된 외부 구성 설정이 지원되지 않음을 나타냅니다.

Config 파일 설정 목록

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

설정 이름	Details
account_id_endpoint_mode	계정 기반 엔드포인트
api_versions	일반 구성 설정
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
credential_source	역할 보안 인증 제공자 수입
defaults_mode	스마트 구성 기본값
disable_request_compression	요청 압축

설정 이름	Details
duration_seconds	역할 보안 인증 공급자 위임
ec2_metadata_service_endpoint	IMDS 자격 증명 공급자
ec2_metadata_service_endpoint_mode	IMDS 자격 증명 공급자
ec2_metadata_v1_disabled	IMDS 자격 증명 공급자
endpoint_discovery_enabled	엔드포인트 검색
endpoint_url	서비스별 엔드포인트
external_id	역할 보안 인증 제공자 수입
ignore_configured_endpoint_urls	서비스별 엔드포인트
max_attempts	재시도 동작
metadata_service_number_attempts	Amazon EC2 인스턴스 메타데이터
metadata_service_timeout	Amazon EC2 인스턴스 메타데이터

설정 이름	Details
mfa_serial	역할 보안 인증 제공자 수입
output	일반 구성 설정
parameter_validation	일반 구성 설정
region	AWS 리전
request_max_in_compression_size_bytes	요청 압축
retry_mode	재시도 동작
role_arn	역할 보안 인증 제공자 수입
role_session_name	역할 보안 인증 제공자 수입
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
source_profile	역할 보안 인증 공급자 위임
sso_account_id	IAM Identity Center 자격 증명 공급자
sso_region	IAM Identity Center 자격 증명 공급자

설정 이름	Details
sso_registration_scopes	IAM Identity Center 자격 증명 공급자
sso_role_name	IAM Identity Center 자격 증명 공급자
sso_start_url	IAM Identity Center 자격 증명 공급자
sts_regional_endpoints	AWS STS 리전 엔드포인트
use_dualstack_endpoint	듀얼 스택 및 FIPS 엔드포인트
use_fips_endpoint	듀얼 스택 및 FIPS 엔드포인트
web_identity_token_file	역할 보안 인증 제공자 수입

Credentials 파일 설정 목록

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

설정 이름	Details
aws_access_key_id	AWS 액세스 키
aws_secret_access_key	AWS 액세스 키
aws_session_token	AWS 액세스 키

환경 변수 목록

대부분의 에서 지원하는 환경 변수 SDKs는 다음 표에 나열되어 있습니다. 이는 전 세계에 적용되며 모든 AWS 서비스에 영향을 미칩니다. SDKs 및 도구는 고유한 설정 및 환경 변수도 지원할 수 있습니다. 개인 SDK 또는 도구에서만 지원되는 설정 및 환경 변수를 보려면 해당 특정 SDK 또는 도구 안내서를 참조하세요.

설정 이름	Details
AWS_ACCESS_KEY_ID	AWS 액세스 키
AWS_ACCOUNT_ID	계정 기반 엔드포인트
AWS_ACCOUNT_ID_ENDPOINT_MODE	계정 기반 엔드포인트
AWS_CA_BUNDLE	일반 구성 설정
AWS_CONFIG_FILE	공유 config 및 credentials 파일 위치
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_CREDENTIALS	컨테이너 보안 인증 제공업체

설정 이름	Details
ENTIALS_RELATIVE_URI	
AWS_DEFAULTS_MODE	스마트 구성 기본값
AWS_DISABLE_REQUEST_COMPRESSION	요청 압축
AWS_EC2_METADATA_DISABLED	IMDS 자격 증명 공급자
AWS_EC2_METADATA_SERVICE_ENDPOINT	IMDS 자격 증명 공급자
AWS_EC2_METADATA_SERVICE_ENDPOINT_MODE	IMDS 자격 증명 공급자
AWS_EC2_METADATA_V1_DISABLED	IMDS 자격 증명 공급자
AWS_ENABLE_ENDPOINT_DISCOVERY	엔드포인트 검색
AWS_ENDPOINT_URL	서비스별 엔드포인트

설정 이름	Details
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 파일 credentials
AWS_REGION	AWS 리전
AWS_REQUEST_COMPRESSION_SIZE_BYTES	요청 압축
AWS_RETRY_MODE	재시도 동작
AWS_ROLE_ARN	역할 보안 인증 제공자 수입
AWS_ROLE_SESSION_NAME	역할 보안 인증 제공자 수입

설정 이름	Details
AWS_S3_DISTRIBUTABLE_REGIONAL_ACCESS_POINTS	Amazon S3 다중 리전 액세스 포인트
AWS_S3_US_E_ASN_REGION	Amazon S3 액세스 포인트
AWS_SDK_USER_APP_ID	애플리케이션 ID
AWS_SECRET_ACCESS_KEY	AWS 액세스 키
AWS_SESSION_TOKEN	AWS 액세스 키
AWS_SHARED_CREDENTIALS_FILE	공유 config 및 credentials 파일 위치
AWS_STS_REGIONAL_ENDPOINTS	AWS STS 리전 엔드포인트
AWS_USE_DUALSTACK_ENDPOINT	듀얼 스택 및 FIPS 엔드포인트
AWS_USE_FIPS_ENDPOINT	듀얼 스택 및 FIPS 엔드포인트
AWS_WEB_IDENTITY_TOKEN_FILE	역할 보안 인증 공급자 위임

JVM 시스템 속성 목록

AWS SDK for Java 및 AWS SDK for Kotlin (대상)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	Details
<code>aws.accessKeyId</code>	AWS 액세스 키
<code>aws.accountId</code>	계정 기반 엔드포인트
<code>aws.accountIdEndpointMode</code>	계정 기반 엔드포인트
<code>aws.configFile</code>	공유 config 및 credentials 파일 위치
<code>aws.defaultsMode</code>	스마트 구성 기본값
<code>aws.disableEc2MetadataV1</code>	IMDS 자격 증명 공급자
<code>aws.disableRequestCompression</code>	요청 압축
<code>aws.ec2MetadataServiceEndpoint</code>	IMDS 자격 증명 공급자
<code>aws.ec2MetadataServiceEndpointMode</code>	IMDS 자격 증명 공급자

설정 이름	Details
<code>aws.endpointDiscoveryEnabled</code>	엔드포인트 검색
<code>aws.endpointUrl</code>	서비스별 엔드포인트
<code>aws.endpointUrl<ServiceName></code>	서비스별 엔드포인트
<code>aws.ignoreConfiguredEndpointUrls</code>	서비스별 엔드포인트
<code>aws.maxAttempts</code>	재시도 동작
<code>aws.profile</code>	공유 config 및 파일 credentials
<code>aws.region</code>	AWS 리전
<code>aws.requestMinCompressionSizeBytes</code>	요청 압축
<code>aws.retryMode</code>	재시도 동작
<code>aws.roleArn</code>	역할 보안 인증 제공자 수입
<code>aws.roleSessionName</code>	역할 보안 인증 제공자 수입
<code>aws.s3DisableMultiRegionAccessPoints</code>	Amazon S3 다중 리전 액세스 포인트

설정 이름	Details
aws.s3UseArnRegion	Amazon S3 액세스 포인트
aws.secretAccessKey	AWS 액세스 키
aws.sessionToken	AWS 액세스 키
aws.sharedCredentialsFile	공유 config 및 credentials 파일 위치
aws.useDualstackEndpoint	듀얼 스택 및 FIPS 엔드포인트
aws.useFipsEndpoint	듀얼 스택 및 FIPS 엔드포인트
aws.userAgentAppId	애플리케이션 ID
aws.webIdentityTokenFile	역할 보안 인증 공급자 위임

AWS SDKs 및 도구 표준화된 자격 증명 공급자

많은 자격 증명 공급자가 일관된 기본값으로 표준화되었으며 많은 에서 동일한 방식으로 작동합니다 SDKs. 이러한 일관성은 여러 에서 코딩할 때 생산성과 명확성을 높입니다 SDKs. 코딩으로 모든 설정을 재정의할 수 있습니다. 자세한 내용은 특정 SDK 를 참조하세요 API.

⚠ Important

모든 공급자 또는 공급자 내의 모든 측면을 SDKs 모두 지원하는 것은 아닙니다.

주제

- [자격 증명 공급자 체인 이해](#)
- [SDK-특정 및 도구별 자격 증명 공급자 체인](#)
- [AWS 액세스 키](#)
- [역할 보안 인증 제공자 수입](#)
- [컨테이너 보안 인증 제공업체](#)
- [IAM Identity Center 자격 증명 공급자](#)
- [IMDS 자격 증명 공급자](#)
- [프로세스 보안 인증 제공자](#)

자격 증명 공급자 체인 이해

모든 SDKs 에는 에 요청할 때 사용할 유효한 보안 인증 정보를 찾기 위해 확인하는 일련의 위치(또는 소스)가 있습니다 AWS 서비스. 유효한 보안 인증 정보를 찾은 후에는 검색이 중지됩니다. 이러한 체계적 검색을 자격 증명 공급자 체인이라고 합니다.

표준화된 자격 증명 공급자 중 하나를 사용할 때는 AWS SDKs 자격 증명이 만료되면 항상 자동으로 갱신을 시도합니다. 기본 제공 보안 인증 공급자 체인은 체인에서 사용 중인 공급자에 관계없이 보안 인증을 새로 고칠 수 있는 기능을 애플리케이션에 제공합니다. 이를 SDK 위해 에 추가 코드가 필요하지 않습니다.

각 에서 사용하는 고유한 체인은 SDK 다양하지만 가장 자주 다음과 같은 소스가 포함됩니다.

보안 인증 제공업체	설명
AWS 액세스 키	AWS IAM 사용자의 액세스 키(예: 및 <code>AWS_ACCESS_KEY_ID</code> <code>AWS_SECRET_ACCESS_KEY</code>).
웹 자격 증명 또는 OpenID Connect 와 페더레이션- 역할 보안 인증 제공자 수입	Amazon, Facebook, Google 또는 기타 OpenID Connect() 호환 IdP 를 사용한 로그인과 같은 잘 알려진 외부 자격 증명 공급자(IdP OIDC)를 사용하여 로그인합니다. AWS

보안 인증 제공업체	설명
	Security Token Service ()의 JSON 웹 토큰(JWT)을 사용하여 IAM 역할의 권한을 맡습니다AWS STS.
IAM Identity Center 자격 증명 공급자	에서 자격 증명을 가져옵니다 AWS IAM Identity Center.
역할 보안 인증 제공자 수입	IAM 역할의 권한을 맡아 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 정보. 컨테이너 보안 인증 공급자는 고객의 컨테이너 애플리케이션에 대한 보안 인증을 가져옵니다.
프로세스 보안 인증 제공자	사용자 정의 보안 인증 공급자. IAM Roles Anywhere를 포함한 외부 소스 또는 프로세스에서 보안 인증을 가져옵니다.
IMDS 자격 증명 공급자	Amazon Elastic Compute Cloud(AmazonEC2) 인스턴스 프로필 보안 인증 정보입니다. IAM 역할을 각 EC2 인스턴스와 연결합니다. 그러면 인스턴스에서 실행되는 코드에 대해 해당 역할의 임시 보안 인증을 사용할 수 있게 됩니다. 자격 증명은 Amazon EC2 메타데이터 서비스를 통해 전달됩니다.

체인의 각 단계마다 설정 값을 할당하는 여러 방법이 있습니다. 코드에 지정된 설정값이 항상 우선합니다. 그러나 [환경 변수](#)와 [공유 config 및 credentials 파일](#)같은 경우도 있습니다. 자세한 내용은 [설정의 우선 순위](#) 단원을 참조하십시오.

SDK-특정 및 도구별 자격 증명 공급자 체인

또는 도구SDK의 특정 자격 증명 공급자 체인 세부 정보로 직접 이동하려면 다음에서 SDK 또는 도구를 선택합니다.

- [AWS CLI](#)
- [SDK C++용](#)
- [SDK Go용](#)

- [SDK Java용](#)
- [SDK 에 대한 JavaScript](#)
- [SDK Kotlin용](#)
- [SDK 용.NET](#)
- [SDK 에 대한 PHP](#)
- [SDK Python용\(Boto3\)](#)
- [SDK Ruby용](#)
- [SDK Rust용](#)
- [SDK Swift용](#)
- [용 도구 PowerShell](#)

AWS 액세스 키

Warning

보안 위험을 피하려면 전용 소프트웨어를 개발하거나 실제 데이터로 작업할 때 IAM 사용자를 인증에 사용하지 마세요. 대신 다음과 같은 ID 공급자와의 페더레이션을 사용하십시오. [AWS IAM Identity Center](#).

AWS IAM사용자의 액세스 키는 다음과 같이 사용할 수 있습니다. AWS 자격 증명. 더 AWS SDK다음 을 자동으로 사용합니다. AWS API요청에 서명하기 위한 자격 증명 AWS워크로드가 다음 데이터에 액세스할 수 있도록 AWS 리소스와 데이터를 안전하고 편리하게. 보안 인증이 일시적 유효하지 않거나 만료 후에는 더 이상 유효하지 않도록 항상 `aws_session_token`을 사용하는 것이 좋습니다. 장기 자격 증명은 사용하지 않는 것이 좋습니다.

Note

If AWS 이러한 임시 자격 증명을 새로 고칠 수 없게 됩니다. AWS 워크로드에 영향을 주지 않도록 자격 증명의 유효 기간을 연장할 수 있습니다.

공유 AWS credentials파일은 애플리케이션 소스 디렉토리 외부에 안전하게 보관되고 공유 config 파일의 SDK 특정 설정과 분리되므로 자격 증명 정보를 저장하는 데 사용하는 것이 좋습니다.

알아볼 내용 AWS 자격 증명 및 액세스 키 사용에 대한 자세한 내용은 을 참조하십시오. [AWSIAM사용자 안내서의 보안 자격 증명 및 사용자 액세스 키 관리 IAM](#)

다음을 사용하여 이 기능을 구성하십시오.

aws_access_key_id- 공유 AWS config파일 설정, **aws_access_key_id**- 공유 AWS **credentials**파일 설정 (권장 방법), **AWS_ACCESS_KEY_ID** - 환경 변수, **aws.accessKeyId**- JVM 시스템 속성: 자바/코틀린 전용

다음을 지정합니다. AWS 사용자를 인증하기 위한 자격 증명의 일부로 사용되는 액세스 키.

aws_secret_access_key- 공유 AWS config파일 설정, **aws_secret_access_key**- 공유 AWS **credentials**파일 설정 (권장 방법), **AWS_SECRET_ACCESS_KEY** - 환경 변수, **aws.secretAccessKey**- JVM 시스템 속성: 자바/코틀린 전용

다음을 지정합니다. AWS 사용자를 인증하기 위한 자격 증명의 일부로 사용되는 비밀 키입니다.

aws_session_token- 공유 AWS config파일 설정, **aws_session_token**- 공유 AWS **credentials**파일 설정 (권장 방법), **AWS_SESSION_TOKEN** - 환경 변수, **aws.sessionToken**- JVM 시스템 속성: 자바/코틀린 전용

를 지정합니다. AWS 사용자를 인증하기 위한 자격 증명의 일부로 사용되는 세션 토큰입니다. 역할 수임 요청이 성공하면 반환되는 임시 보안 인증 정보의 일부로 이 값을 받습니다. 세션 토큰은 수동으로 임시 보안 인증을 지정하는 경우에만 필요합니다. 하지만 장기 보안 인증 정보를 사용하는 대신 항상 임시 보안 인증을 사용하는 것이 좋습니다. 보안 권장 사항은 의 [보안 모범 사례](#) 를 참조하십시오. IAM

이러한 값을 구하는 방법에 대한 자세한 내용은 [단기 보안 인증을 사용한 인증](#) 단원을 참조하십시오.

config또는 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...zrkuWJ0gQs8IZZaIv2BXIa2R40lgk
```

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...zrkuWJ0gQs8IZZaIv2BXIa2R40lgk
```

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...zrkuWJ0gQs8IZZaIv2BXIa2R40lgk
```

와의 호환성 AWS SDKs

다음은 이 항목에 설명된 기능 및 설정을 SDKs 지원합니다. 모든 일부 예외가 기록됩니다. 모든 JVM 시스템 속성 설정은 에서 지원됩니다. AWS SDK for Java 그리고 AWS SDK for Kotlin 오직.

SDK	지 운	참고 또는 추가 정보
AWS CLI v2	예	
SDKC++의 경우	예	공유 config 파일은 지원되지 않습니다.
SDK고 V2 (1.x) 용	예	
SDKGo 1.x (V1) 용	예	공유 config 파일 설정을 사용하려면 구성 파일에서 로드를 켜야 합니다. 세션 을 참조하십시오.
SDK자바 2.x의 경우	예	
SDK자바 1.x의 경우	예	
SDK3.x용 JavaScript	예	
SDK2.x의 경우 JavaScript	예	
SDK코틀린의 경우	예	
SDK왜냐하면. NET3.x	예	환경 변수는 지원되지 않습니다.
SDK3.x의 경우 PHP	예	

SDK	지 운	참고 또는 추가 정보
SDK파이썬용 (보토3)	예	
SDK루비 3.x의 경우	예	
SDK러스트용	예	
SDK스위프트용	예	
를 위한 도구 PowerShell	예	환경 변수는 지원되지 않습니다.

역할 보안 인증 제공자 수입

역할 수입에는 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 수입하도록 SDK 또는 도구를 설정하려면 먼저 수입할 특정 역할을 생성하거나 식별해야 합니다. IAM 역할은 Amazon 리소스 이름(ARN) 역할로 고유하게 식별됩니다. 역할은 다른 엔티티와 신뢰 관계를 구축합니다. 역할을 사용하는 신뢰할 수 있는 엔티티는 AWS 서비스, 다른 AWS 계정, 웹 자격 증명 공급자 또는 OIDC 또는 SAML 연동일 수 있습니다.

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

이러한 설정 사용을 시작하는 방법에 대한 지침은 이 안내서의 [AWS 자격 증명으로 역할 수입](#)를 참조하세요.

역할 보안 인증 제공자 수입 설정

다음을 사용하여 이 기능을 구성하십시오.

credential_source - 공유 AWS **config** 파일 설정

Amazon EC2 인스턴스 또는 Amazon Elastic Container Service 컨테이너 내에서 SDK 또는 도구가 `role_arn` 파라미터로 지정한 역할을 수입할 권한이 있는 자격 증명을 찾을 수 있는 위치를 지정하는 데 사용됩니다.

기본값: 없음

유효한 값:

- 환경 - SDK 또는 도구가 환경 변수 [AWS_ACCESS_KEY_ID](#) 및 [AWS_SECRET_ACCESS_KEY](#)에서 소스 자격 증명을 검색하도록 지정합니다.
- Ec2InstanceMetadata - SDK 또는 도구가 [IAM EC2 인스턴스 프로파일에 연결된 역할을](#) 사용하여 소스 자격 증명을 가져오도록 지정합니다.
- EcsContainer - SDK 또는 도구가 [IAM ECS 컨테이너에 연결된 역할을](#) 사용하여 소스 자격 증명을 가져오도록 지정합니다.

credential_source과 source_profile 모두를 동일한 프로파일에서 지정할 수 없습니다.

Amazon에서 자격 증명을 소싱해야 함을 나타내도록 config 파일에서 이를 설정하는 예EC2:

```
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 수입을 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 설명서 [의 역할의 최대 세션 기간 설정 보기를](#) 참조하세요.

config 파일에서 이를 설정하는 예:

```
duration_seconds = 43200
```

external_id - 공유 AWS config 파일 설정

타사에서 고객 계정의 역할을 수입하는 데 사용하는 고유한 식별자를 지정합니다.

이 설정은 프로파일에서 역할을 수입하도록 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 역할을 수입할 때 사용해야 하는 다중 인증(MFA) 디바이스의 식별 또는 일련 번호를 지정합니다.

해당 역할의 신뢰 정책에 MFA 인증이 필요한 조건이 포함된 역할을 수입할 때 필요합니다. 에 대한 자세한 내용은 IAM 사용 설명서의 [AWS 에서 다중 인증 IAM](#) 섹션을 MFA참조하세요.

기본값: 없음.

유효한 값: 값은 하드웨어 디바이스의 일련 번호(예: GAHT12345678) 또는 가상 MFA 디바이스의 Amazon 리소스 이름(ARN)일 수 있습니다. 의 형식은 다음과 ARN 같습니다.

```
arn:aws:iam::account-id:mfa/mfa-device-name
```

config 파일에서 이를 설정하는 예:

이 예제에서는 계정에 대해 생성MyMFADevice되고 사용자에게 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: 값은 다음과 같은 형식ARN의 IAM 역할의 이어야 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 자격 증명 또는 Amazon EC2 인스턴스 프로파일에 연결된 자격 증명 또는 Amazon ECS 컨테이너 인스턴스를 사용합니다.
- `web_identity_token_file` - 모바일 또는 웹 애플리케이션에서 인증된 사용자를 위해 퍼블릭 자격 증명 공급자 또는 OpenID Connect(OIDC) 호환 자격 증명 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. 이 값은 쿼리할 수 있는 수임 역할 사용자 Amazon 리소스 이름(ARN)의 일부가 되며 프로파일에서 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 프로파일이 사용되는 방법을 알아보려면 섹션을 참조하세요 [공유 config 및 credentials 파일](#).

역할 수임 프로파일이기도 한 프로파일을 지정하는 경우 보안 인증을 완전히 확인하기 위해 각 역할이 순차적으로 수임됩니다. 이 체인은가 자격 증명이 있는 프로필을 SDK 발견하면 중지됩니다. 역할 연결은 AWS CLI 또는 AWS API 역할 세션을 최대 1시간으로 제한하며 늘릴 수 없습니다. 자세한 내용은 IAM 사용 설명서의 [역할 용어 및 개념](#)을 참조하세요.

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 프로파일은 SDK 또는 도구에 연결된 B 프로파일의 자격 증명을 자동으로 조회하도록 지시합니다. 이 경우 B 프로파일은에서 제공하는 자격 증명 헬퍼 도구를 사용하여 AWS 대한 자격 증명을 [IAM Roles Anywhere](#) 가져옵니다. 이러한 임시 보안 인증은 코드에서 AWS 리소스에 액세스하기 위해 사용됩니다. 지정된 역할에는 명령 AWS 서비스 또는 API 메서드와 같이 요청된 코드가 실행되도록 허용하는 IAM 권한 정책이 연결되어 있어야 합니다. 프로파일에서 수행하는 모든 작업에는 CloudTrail 로그에 역할 세션 이름이 포함됩니다.

역할 체인의 두 번째 예제에서는 Amazon Elastic Compute Cloud 인스턴스에 애플리케이션이 있고 해당 애플리케이션이 다른 역할을 맡도록 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
```

프로파일A은 Amazon EC2 인스턴스의 자격 증명을 사용하여 지정된 역할을 수입하고 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 OAuth2.0 공급자](#) 또는 [OpenID Connect ID 자격 증명 공급자](#)의 액세스 토큰이 포함된 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 이를 WebIdentityToken 인수로 전달합니다.

기본값: 없음.

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

와의 호환성 AWS SDKs

다음은 이 주제에 설명된 기능 및 설정을 SDKs 지원합니다. 모든 일부 예외가 기록됩니다. 모든 JVM 시스템 속성 설정은 AWS SDK for Java 및 AWS SDK for Kotlin 에서만 지원됩니다.

SDK	지	참고 또는 추가 정보
AWS CLI v2	예	
SDK C++용	부	credential_source 이 지원되지 않음. duration_부분 seconds 이 지원되지 않음. mfa_serial 이 지원되지 않음.
SDK Go V2용(1.x)	예	
SDK Go 1.x(V1)용	예	공유 config 파일 설정을 사용하려면 구성 파일에서 로드를 켜야 합니다. 세션 을 참조하십시오.
SDK Java 2.x용	부	mfa_serial 지원되지 않습니다. 지원되지 duration_부분 seconds 않습니다.
SDK Java 1.x용	부	credential_source 지원되지 않음. 지원되지 mfa_serial 1 않음. JVM 시스템 속성이 지원되지 않음.

SDK	지 운	참고 또는 추가 정보
SDK for JavaScript 3.x	예	
SDK for JavaScript 2.x	부 분	credential_source 이 지원되지 않음.
SDK Kotlin용	예	
SDK 용.NET 3.x	예	
SDK 3.xPHP용	예	
SDK Python용(Boto3)	예	
SDK Ruby 3.x용	예	
SDK Rust용	예	
SDK Swift용	예	
용 도구 PowerShell	예	

컨테이너 보안 인증 제공업체

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

Amazon 를 사용하는 경우 보안 인증 정보 격리ECS, 권한 부여 및 감사 가능성을 개선하기 위해 작업 IAM 역할을 사용하는 것이 좋습니다. 구성되면 Amazon은 SDKs 및 도구가 자격 증명을 얻는 데 사용하는 `AWS_CONTAINER_CREDENTIALS_RELATIVE_URI` 환경 변수를 ECS 설정합니다. 이 기능에 ECS 대해 Amazon을 구성하려면 Amazon Elastic Container Service 개발자 안내서의 [태스크 IAM 역할을 참조하세요](#).

Amazon 를 사용하는 경우 Amazon EKS Pod Identity를 사용하여 보안 인증 정보 격리EKS, 최소 권한, 감사 가능성, 독립적인 운영, 재사용 가능성 및 확장성을 개선하는 것이 좋습니다. Pod와 IAM 역할 모두 Kubernetes 서비스 계정과 연결되어 애플리케이션의 보안 인증을 관리합니다. Amazon EKS Pod

자격 증명에 대한 자세한 내용은 Amazon EKS 사용 설명서의 [Amazon EKS Pod 자격 증명](#)을 참조하세요. 구성되면 Amazon은 `AWS_CONTAINER_CREDENTIALS_FULL_URI` 및 도구가 자격 증명을 얻는데 사용하는 SDKs 및 `AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE` 환경 변수를 EKS 설정합니다. 설정 정보는 Amazon EKS 사용 설명서의 [Amazon EKS Pod Identity Agent 설정](#)을 참조하거나 AWS 블로그 웹 사이트의 [Amazon EKS 클러스터에 있는 애플리케이션에 대한 IAM 권한을 EKS 단순화](#)합니다.

다음을 사용하여 이 기능을 구성하십시오.

`AWS_CONTAINER_CREDENTIALS_FULL_URI` - 환경 변수

자격 증명을 요청할 때 SDK 사용할 의 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`이 설정되지 않은 경우에만 사용됩니다.

명령 행을 통한 환경 변수 설정의 Linux/macOS 예제:

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

또는

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

`AWS_CONTAINER_CREDENTIALS_RELATIVE_URI` - 환경 변수

자격 증명을 요청할 때 SDK 사용할 의 상대 HTTP URL 엔드포인트를 지정합니다. 값은 의 기본 Amazon ECS 호스트 이름에 추가됩니다 `169.254.170.2`.

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 변수가 설정되면 SDK는 환경 변수의 값으로 HTTP 요청에 권한 부여 헤더를 설정합니다.

기본값: 없음.

유효한 값: 문자열.

참고: 이 설정은 `AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE` 설정의 대안이며 `AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE`이 설정되지 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
```

와의 호환성 AWS SDKs

다음은 이 주제에 설명된 기능 및 설정을 SDKs 지원합니다. 모든 일부 예외가 기록됩니다. 모든 JVM 시스템 속성 설정은 AWS SDK for Java 및 AWS SDK for Kotlin 에서만 지원됩니다.

SDK	지참고 또는 추가 정보
AWS CLI v2	예
SDK C++용	예

SDK	지 은	참고 또는 추가 정보
SDK Go V2(1.x)용	예	
SDK Go 1.x(V1)용	예	
SDK Java 2.x용	예	AWS_CONTAINER_CREDENTIALS_FULL_URI 및 AWS_CONTAINER_AUTHORIZATION_TOKEN 는 Lambda SnapStart for Java 에도 사용됩니다.
SDK Java 1.x용	예	AWS_CONTAINER_CREDENTIALS_FULL_URI 및 AWS_CONTAINER_AUTHORIZATION_TOKEN 는 Lambda SnapStart for Java 에도 사용됩니다.
SDK JavaScript 3.x용	예	
SDK JavaScript 2.x용	예	
SDK Kotlin용	예	
SDK 용.NET 3.x	예	
SDK 3.xPHP용	예	
SDK Python용(Boto3)	예	
SDK Ruby 3.x용	예	
SDK Rust용	예	
SDK Swift용	예	
용 도구 PowerShell	예	

IAM Identity Center 자격 증명 공급자

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

Note

API 설명서에서는 AWS SDK IAM Identity Center 자격 증명 공급자를 SSO 자격 증명 공급자라고 합니다.

IAM Identity Center를 활성화한 후 공유 AWS config 파일에서 해당 설정에 대한 프로파일을 정의합니다. 이 프로파일은 IAM Identity Center 액세스 포털에 연결하는 데 사용됩니다. 사용자가 IAM Identity Center로 성공적으로 인증하면 포털은 해당 사용자와 연결된 IAM 역할에 대한 단기 자격 증명을 반환합니다. 가 구성에서 임시 자격 증명을 SDK 가져와 AWS 서비스 요청에 사용하는 방법을 알아보려면 섹션을 참조하세요 [IAM Identity Center 인증 이해하기](#).

config 파일을 통해 IAM Identity Center를 구성하는 방법에는 두 가지가 있습니다.

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

두 구성 모두 세션이 만료되면 다시 로그인해야 합니다.

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

- [AWS IAM Identity Center 사용 설명서](#)
- [AWS IAM Identity Center 포털 API 참조](#)

SDKs 및 도구가 구성을 사용하여 자격 증명을 사용하고 새로 고치는 방법에 대한 자세한 내용은 섹션을 참조하세요 [IAM Identity Center 인증 이해하기](#).

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 Center](#)를 참조하세요.

Note

또는 이 페이지에 자세히 설명된 전체 사전 조건 및 필요한 공유 config 파일 구성은 설정에 대한 안내 지침을 참조하세요 [IAM SDK 또는 도구에 대한 Identity Center 인증](#).

SSO 토큰 공급자 구성

SSO 토큰 공급자 구성을 AWS SDK 사용하면 또는 도구가 세션 기간을 연장할 때까지 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 구성](#)을 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 내의 섹션에 대한 자세한 내용은 섹션을 참조하세요 [구성 파일 형식](#).

다음 공유 config 파일 예제에서는 dev 프로파일을 사용하여 IAM Identity Center 자격 증명을 요청하는 SDK 또는 도구를 구성합니다.

```
[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 정의하여 프로파일에 연결했음을 보여줍니다. 일반적으로 AWS 자격 증명을 요청할 SDK 수 있도록 profile 섹션에서 sso_account_id 및 sso_role_name을 설정해야 합니다. sso-session 섹션 내에서 sso_start_url, 및 sso_region를 설정해야 sso_registration_scopes 합니다.

sso_account_id SSO 토큰 구성의 모든 시나리오에 및 sso_role_name가 필요하지는 않습니다. 애플리케이션에서 무기명 인증을 지원하는 AWS 서비스 만 사용하는 경우 기존 AWS 자격 증명도 필요하지 않습니다. 무기명 인증은 무기명 토큰이라는 보안 토큰을 사용하는 HTTP 인증 체계입니다. 이 시나리오에서는 sso_account_id 및 sso_role_name은 필수가 아닙니다. 서비스가 베어러 토큰 인증을 지원하는지 확인하려면 개별 AWS 서비스 가이드를 참조하세요.

등록 범위는 sso-session의 일부로 구성됩니다. 범위는의 메커니즘입니다.OAuth 2.0 사용자 계정에 대한 애플리케이션의 액세스를 제한합니다. 이전 예제에서는 계정 및 역할을 나열sso_registration_scopes하는 데 필요한 액세스를 제공하도록 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 구성](#)을(를)사용하는 것이 좋습니다.

새로 고칠 수 없는 기존 구성을 사용하려면 프로파일에 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 자격 증명 공급자 설정

다음을 사용하여 이 기능을 구성하십시오.

`sso_start_url` - 공유 AWS `config` 파일 설정

조직의 IAM Identity Center 발급자 URL 또는 액세스 포털을 가리키는 URL은 `입니다URL`. 자세한 내용은 AWS IAM Identity Center 사용 설명서의 [AWS 액세스 포털 사용](#)을 참조하세요.

이 값을 찾으려면 [IAM Identity Center 콘솔](#)을 열고 대시보드를 보고 AWS 액세스 포털을 URL 찾습니다.

- 또는 버전 2.22.0부터 AWS 는 발급자 URL의 값을 사용할 AWS CLI 수 있습니다.

`sso_region` - 공유 AWS `config` 파일 설정

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

AWS 리전 및 해당 코드의 전체 목록은 [리전 엔드포인트](#)를 참조하세요 Amazon Web Services 일반 참조. 이 값을 찾으려면 [IAM Identity Center 콘솔](#)을 열고 대시보드를 보고 리전을 찾습니다.

`sso_account_id` - 공유 AWS `config` 파일 설정

인증에 사용할 AWS Organizations 서비스를 통해 추가된의 숫자 ID AWS 계정입니다.

사용 가능한 계정 목록을 보려면 [IAM Identity Center 콘솔](#)로 이동하여 AWS 계정 페이지를 엽니다. AWS IAM Identity Center 포털 API 참조에서 [ListAccounts](#) API 메서드를 사용하여 사용 가능한 계정 목록을 볼 수도 있습니다. 예를 들어 AWS CLI 메서드 [list-accounts](#)를 호출할 수 있습니다.

`sso_role_name` - 공유 AWS `config` 파일 설정

사용자의 결과 권한을 정의하는 IAM 역할로 프로비저닝된 권한 세트의 이름입니다. 역할은에서 AWS 계정 지정함에 있어야 합니다 `sso_account_id`. Amazon 리소스 이름() 역할이 아닌 역할 이름을 사용합니다ARN.

권한 세트에는 IAM 정책 및 사용자 지정 권한 정책이 연결되어 있으며 사용자가 할당된에 대한 액세스 수준을 정의합니다 AWS 계정.

에 따라 사용 가능한 권한 세트 목록을 보려면 [IAM Identity Center 콘솔](#)로 AWS 계정이동하여 AWS 계정 페이지를 엽니다. AWS 계정 테이블에 나열된 올바른 권한 세트 이름을 선택합니다. AWS IAM

Identity Center 포털 API 참조에서 [ListAccountRoles](#) API 메서드를 사용하여 사용 가능한 권한 세트 목록을 볼 수도 있습니다. 예를 들어 AWS CLI 메서드를 호출할 수 있습니다 [list-account-roles](#).

sso_registration_scopes - 공유 AWS config 파일 설정

sso-session에 대해 인증될 심표로 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 가져오려면의 최소 범위가 부여되어야 sso:account:access 합니다. 사용 가능한 액세스 범위 옵션 목록은 AWS IAM Identity Center 사용 설명서의 [액세스 범위를](#) 참조하세요.

이러한 범위는 등록된 OIDC 클라이언트 및 클라이언트가 검색한 액세스 토큰에 대해 권한을 부여하도록 요청된 권한을 정의합니다. 범위는 IAM Identity Center 베어러 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.

와의 호환성 AWS SDKs

다음은이 주제에 설명된 기능 및 설정을 SDKs 지원합니다. 모든 일부 예외가 기록됩니다. 모든 JVM 시스템 속성 설정은 AWS SDK for Java 및 AWS SDK for Kotlin 에서만 지원됩니다.

SDK	지	참고 또는 추가 정보
AWS CLI v2	예	
SDK C++용	예	
SDK Go V2(1.x)용	예	
SDK Go 1.x(V1)용	예	공유 config 파일 설정을 사용하려면 구성 파일에서 로드를 켜야 합니다. 세션 을 참조하십시오.
SDK Java 2.x용	예	구성 값은 credentials 파일에서도 지원됩니다.
SDK Java 1.x용	아 니 요	

SDK	지참고 또는 추가 정보 온
SDK for JavaScript 3.x	예
SDK for JavaScript 2.x	예
SDK Kotlin용	예
SDK 용.NET 3.x	예
SDK 3.xPHP용	예
SDK Python용(Boto3)	예
SDK Ruby 3.x용	예
SDK Rust용	부 분 새로 고칠 수 없는 레거시 구성에만 해당.
SDK Swift용	예
용 도구 PowerShell	예

IMDS 자격 증명 공급자

인스턴스 메타데이터 서비스(IMDS)는 실행 중인 인스턴스를 구성하거나 관리하는 데 사용할 수 있는 인스턴스에 대한 데이터를 제공합니다. 사용 가능한 데이터에 대한 자세한 내용은 Amazon EC2 사용 설명서의 [인스턴스 메타데이터 작업을](#) 참조하세요. AmazonEC2은 인스턴스에 다양한 정보 비트를 제공할 수 있는 인스턴스에 사용할 수 있는 로컬 엔드포인트를 제공합니다. 인스턴스에 역할이 연결된 경우 해당 역할에 유효한 보안 인증 세트를 제공할 수 있습니다. 는 해당 엔드포인트를 사용하여 [기본 자격 증명 공급자 체인의 일부로 자격 증명을](#) 확인할 SDKs 수 있습니다. 인스턴스 메타데이터 서비스 버전 2(IMDSv2)는 세션 토큰을 IMDS 사용하는 보다 안전한 버전이며 기본적으로 사용됩니다. 재시도할 수 없는 조건(HTTP오류 코드 403, 404, 405)으로 인해 실패하는 경우 IMDSv1가 대체로 사용됩니다.

다음을 사용하여 이 기능을 구성하십시오.

AWS_EC2_METADATA_DISABLED - 환경 변수

Amazon EC2 Instance Metadata Service(IMDS)를 사용하여 자격 증명을 얻으려는지 여부입니다.


기본값: `false`

유효값:

- **true** - IMDS를 사용하여 자격 증명을 가져오지 마세요.
- **false** - IMDS를 사용하여 자격 증명을 가져옵니다.

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

IMDSv2 가 실패할 경우 인스턴스 메타데이터 서비스 버전 1(IMDSv1)을 대체로 사용할지 여부입니다.

 Note

새 SDKs 는 이 설정을 지원하지 않으므로 지원하지 않습니다. 자세한 내용은 테이블 [와의 호환성 AWS SDKs](#)를 참조하세요.

기본값: `false`

유효값:

- **true** - 폴백IMDSv1으로 사용하지 마세요.
- **false** - 을 대체IMDSv1로 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

의 엔드포인트입니다IMDS. 이 값은 및 도구가 Amazon EC2 인스턴스 메타데이터를 검색할 기본 위치를 AWS SDKs 재정의합니다.

기본값: `ec2_metadata_service_endpoint_mode`와 IPv4이 같으면 기본 엔드포인트는 `http://169.254.169.254`입니다. `ec2_metadata_service_endpoint_mode`와 IPv6이 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

의 엔드포인트 모드입니다IMDS.

기본값:IPv4.

유효한 값: IPv4, IPv6.

Note

IMDS 자격 증명 공급자는 의 일부입니다[자격 증명 공급자 체인 이해](#). 그러나 자격 IMDS 증명 공급자는 이 시리즈에 속한 다른 여러 공급자 이후에만 확인됩니다. 따라서 프로그램에서 이 공급자의 보안 인증을 사용하려면 구성에서 다른 유효한 보안 인증 공급자를 제거하거나 다른 프로파일을 사용해야 합니다. 또는 자격 증명 공급자 체인에 의존하여 유효한 자격 증명을 반환하는 공급자를 자동으로 검색하는 대신 코드에 자격 IMDS 증명 공급자 사용을 지정합니다. 서비스 클라이언트를 생성할 때 보안 인증 소스를 직접 지정할 수 있습니다.

IMDS 보안 인증 정보 보안

기본적으로 AWS SDK가 유효한 보안 인증 정보로 구성되지 않은 경우 SDK는 Amazon EC2 인스턴스 메타데이터 서비스(IMDS)를 사용하여 AWS 역할에 대한 보안 인증을 검색하려고 시도합니다. `AWS_EC2_METADATA_DISABLED`환경 변수를 `true`로 설정하여 이 동작을 비활성화할 수 있습니다. 이렇게 하면 불필요한 네트워크 활동이 방지되고 Amazon EC2 인스턴스 메타데이터 서비스가 가장될 수 있는 신뢰할 수 없는 네트워크의 보안이 강화됩니다.

Note

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

Amazon EC2 IMDS 자격 증명 사용 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 셸 세션 및 시스템 재시작 시에도 이 설정을 유지하려면 위의 명령을 `.bash_profile`, `.zsh_profile`, 혹은 `.profile` 등의 셸 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 셸 세션 및 시스템 재시작에 걸쳐 지속되게 하려면 다음 명령을 대신 사용할 수 있습니다.

```
$ setx AWS_EC2_METADATA_DISABLED=true
```

Note

이 `setx` 명령은 현재 셸 세션에 값을 적용하지 않으므로 변경 내용을 적용하려면 셸을 다시 로드하거나 다시 열어야 합니다.

와의 호환성 AWS SDKs

다음은 이 주제에 설명된 기능 및 설정을 SDKs 지원합니다. 모든 일부 예외가 기록됩니다. 모든 JVM 시스템 속성 설정은 AWS SDK for Java 및 AWS SDK for Kotlin 에서만 지원됩니다.

SDK	지참고 또는 추가 정보
AWS CLI v2	예
SDK C++용	예
SDK Go V2용(1.x)	예

SDK	지 운	참고 또는 추가 정보
SDK Go 1.x(V1)용	예	공유 config 파일 설정을 사용하려면 구성 파일에서 로드를 켜야 합니다. 세션 을 참조하십시오.
SDK Java 2.x용	예	
SDK Java 1.x용	부 분	JVM 시스템 속성: <code>com.amazonaws.sdk.disableEc2MetadataV1</code> 대신 <code>aws.disableEc2MetadataV1</code> <code>aws.ec2MetadataServiceEndpoint</code> 를 사용하고 지원되지 <code>aws.ec2MetadataServiceEndpointMode</code> 않습니다.
SDK JavaScript 3.x용	예	
SDK JavaScript 2.x용	예	
SDK Kotlin용	예	IMDSv1 대체를 사용하지 않습니다.
SDK 용.NET 3.x	예	
SDK 3.xPHP용	예	
SDK Python용(Boto3)	예	
SDK Ruby 3.x용	예	
SDK Rust용	예	IMDSv1 대체를 사용하지 않습니다.
SDK Swift용	예	
용 도구 PowerShell	예	를 사용하여 코드에서 IMDSv1 대체를 명시적으로 비활성화할 수 있습니다[<code>Amazon.Util.EC2InstanceMetadata>::EC2MetadataV1Disabled = \$true</code> .

프로세스 보안 인증 공급자

SDKs사용자 지정 사용 사례를 위해 자격 증명 공급자 체인을 확장할 수 있는 방법을 제공합니다. 이 공급자를 사용하여 온프레미스 자격 증명 저장소에서 자격 증명을 검색하거나 온-프레미스 ID 공급자와 통합하는 등의 사용자 지정 구현을 제공할 수 있습니다.

예를 들어 IAM Roles Anywhere는 애플리케이션을 `credential_process` 대신하여 임시 자격 증명을 가져오는 데 사용합니다. 이 용도로 `credential_process`를 구성하려면 [IAM Roles Anywhere](#)을 (를)참조하십시오.

Note

다음은 외부 프로세스에서 자격 증명을 소싱하는 방법을 설명하며, 외부에서 소프트웨어를 실행하는 경우 사용할 수 있습니다. AWS. 건물을 지을 경우 AWS 컴퓨팅 리소스를 사용하려면 다른 자격 증명 공급자를 사용하세요. 이 옵션을 사용하는 경우 운영 체제의 보안 모범 사례를 사용하여 구성 파일이 최대한 잠겨 있는지 확인해야 합니다. 사용자 지정 자격 증명 도구가 비밀 정보를 기록하지 않는지 확인하십시오. StdErr SDKs AWS CLI 이러한 정보를 캡처하고 기록할 수 있어 권한이 없는 사용자에게 정보가 노출될 수 있습니다.

다음을 사용하여 이 기능을 구성하십시오.

`credential_process`- 공유 AWS `config`파일 설정

사용할 인증 자격 증명을 SDK 생성하거나 검색하기 위해 또는 도구가 사용자를 대신하여 실행하는 외부 명령을 지정합니다. 설정은 에서 호출할 프로그램/명령의 이름을 지정합니다. SDK 프로세스를 SDK 호출하면 프로세스가 데이터를 쓸 때까지 기다립니다. JSON stdout 사용자 지정 공급자는 특정 형식으로 정보를 반환해야 합니다. 이 정보에는 SDK 또는 도구가 사용자를 인증하는 데 사용할 수 있는 자격 증명이 들어 있습니다.

Note

프로세스 보안 인증 공급자는 [자격 증명 공급자 체인 이해](#)의 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 사

용을 지정하십시오. 서비스 클라이언트를 생성할 때 보안 인증 소스를 직접 지정할 수 있습니다.

보안 인증 프로그램 경로 지정

설정 값은 사용자 대신 SDK 또는 개발 도구가 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 묶으십시오.
- 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 달라집니다.

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

는 프로필에 지정된 대로 명령을 SDK 실행한 다음 표준 출력 스트림에서 데이터를 읽습니다. 스크립트 프로그램이든 바이너리 프로그램이든 관계없이 지정하는 명령은 다음 구문과 STDOUT 일치하는 JSON 출력을 생성해야 합니다.

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

Expiration키는 RFC3339 형식이 지정된 타임스탬프입니다. Expiration키가 도구 출력에 없는 경우는 자격 증명이 새로 고쳐지지 않는 장기 자격 SDK 증명이라고 가정합니다. 그렇지 않은 경우 보안 인증은 임시 보안 인증으로 간주되며, 기간이 만료되기 전에 credential_process 명령을 다시 실행하면 자동으로 새로 고침됩니다.

Note

는 역할 수입 자격 증명과 달리 외부 프로세스 자격 증명을 캐시하지 SDK 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.

와의 호환성 AWS SDKs

다음은 이 항목에 설명된 기능 및 설정을 SDKs 지원합니다. 모든 일부 예외가 기록됩니다. 모든 JVM 시스템 속성 설정은 에서 지원됩니다. AWS SDK for Java 그리고 AWS SDK for Kotlin 오직.

SDK	지	참고 또는 추가 정보
AWS CLI v2	예	
SDKC++의 경우	예	
SDKGo V2 (1.x) 용	예	
SDKGo 1.x (V1) 용	예	공유 config 파일 설정을 사용하려면 구성 파일에서 로드를 켜야 합니다. 세션 을 참조하십시오.
SDK자바 2.x의 경우	예	
SDK자바 1.x의 경우	예	
SDK3.x용 JavaScript	예	
SDK2.x의 경우 JavaScript	예	
SDK코틀린의 경우	예	
SDK왜냐하면. NET3.x	예	
SDK3.x의 경우 PHP	예	
SDK파이썬용 (보토3)	예	
SDK루비 3.x의 경우	예	
SDK러스트용	예	
SDK스위프트용	예	
를 위한 도구 PowerShell	예	

AWS SDKs 및 도구 표준화 기능

많은 기능이 기본값을 일관되게 유지하고 여러 기능에서 동일한 방식으로 작동하도록 표준화되었습니다. SDKs 이러한 일관성은 여러 곳에 걸쳐 코딩할 때 생산성과 명확성을 높여줍니다. SDKs 코드에서 모든 설정을 재정의할 수 있습니다. 자세한 내용은 특정 SDK API 설정을 참조하십시오.

Important

모든 기능을 SDKs 지원하는 것은 아니며 기능 내의 모든 측면을 지원하는 것은 아닙니다.

주제

- [계정 기반 엔드포인트](#)
- [애플리케이션 ID](#)
- [Amazon EC2 인스턴스 메타데이터](#)
- [Amazon S3 액세스 포인트](#)
- [Amazon S3 다중 리전 액세스 포인트](#)
- [AWS 리전](#)
- [AWS STS 리전 엔드포인트](#)
- [듀얼 스택 및 엔드포인트 FIPS](#)
- [엔드포인트 검색](#)
- [일반 구성 설정](#)
- [IMDS 클라이언트](#)
- [재시도 동작](#)
- [요청 압축](#)
- [서비스별 엔드포인트](#)
- [스마트 구성 기본값](#)

계정 기반 엔드포인트

계정 기반 엔드포인트는 AWS 계정 ID를 사용하여 이 기능을 지원하는 서비스에 대한 AWS 서비스 요청 라우팅을 간소화하여 고성능 및 확장성을 보장하는 데 도움이 됩니다. 계정 기반 엔드포인트

를 지원하는 AWS SDK, 자격 증명 공급자 및 서비스를 사용하는 경우 SDK는 리전 엔드포인트가 아닌 계정 기반 엔드포인트를 자동으로 구성하고 사용합니다. 계정 기반 엔드포인트는 의 형태를 취하며 `https://<account-id>.ddb.<region>.amazonaws.com`, 여기서 `<account-id>`는 AWS 계정 ID로 대체되고 `<region>`는 로 대체됩니다. AWS 리전

기본적으로 계정 ID는 요청이 처리되고 엔드포인트를 구성하는 데 사용될 때 수집됩니다. 자격 증명 확인은 요청이 처리될 때도 발생하며 엔드포인트 확인 방법을 변경할 수 있습니다. 사용 중인 자격 증명 공급자에 따라 계정 ID가 다르게 소싱될 수 있습니다.

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** – 엔드포인트에는 사용 가능한 경우 계정 ID가 포함되어야 합니다.
- **disabled** – 확인된 엔드포인트에는 계정 ID가 포함되지 않습니다.
- **required** – 엔드포인트에는 계정 ID가 포함되어야 합니다. 계정 ID를 사용할 수 없는 경우에서 오류가 SDK 발생합니다.

와의 호환성 AWS SDKs

다음은 이 주제에 설명된 기능 및 설정을 SDKs 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	아니요		
SDK C++용	아니요		
SDK Go V2(1.x)용	예	v1.35.0	
SDK Go 1.x(V1)용	아니요		
SDK Java 2.x용	예	v2.28.4	
SDK Java 1.x용	예	v1.12.771	
SDK JavaScript 3.x용	예	v3.656.0	
SDK JavaScript 2.x용	아니요		
SDK Kotlin용	예	v1.3.37	
SDK 용.NET 3.x	아니요		
SDK 3.xPHP용	예	v3.318.0	

SDK	지원	SDK 버전에서 릴리스됨	참고 또는 추가 정보
SDK Python용(Boto3)	아니요		
SDK Ruby 3.x용	예	v1.123.0	
SDK Rust용	아니요		
SDK Swift용	아니요		
용 도구 PowerShell	아니요		

애플리케이션 ID

A 단일 AWS 계정 여러 고객 애플리케이션에서 전화를 거는 데 사용할 수 있습니다. AWS 서비스. 애플리케이션 ID는 고객이 어떤 소스 애플리케이션이 애플리케이션을 사용하여 일련의 호출을 수행했는지 식별할 수 있는 방법을 제공합니다. AWS 계정. AWS SDKs 그리고 서비스에서는 이 값을 고객 커뮤니케이션에 다시 표시하는 것 외에는 사용하지 않습니다. 예를 들어, 이 값은 운영 이메일이나 이메일에 포함될 수 있습니다. AWS Health Dashboard 알림과 관련된 애플리케이션을 고유하게 식별합니다.

다음을 사용하여 이 기능을 구성하십시오.

sdk_ua_app_id- 공유 AWS config 파일 설정, **AWS_SDK_UA_APP_ID** - 환경 변수,
aws.userAgentAppId- JVM 시스템 속성: 자바/코틀린 전용

이 설정은 특정 애플리케이션 내에 있는 애플리케이션을 식별하기 위해 애플리케이션에 할당하는 고유한 문자열입니다. AWS 계정을 호출합니다. AWS.

기본값: None

유효한 값: 최대 길이가 50인 문자열. 문자, 숫자 및 특수 문자 (!, ,, ,, ,, \$, %, &, *, +, -, ., , ^ _ ` |,) 를 사용할 수 ~ 있습니다.

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 셸에 특별한 의미가 있는 기호를 포함하는 경우 값을 적절하게 이스케이프 처리하십시오.

와의 호환성 AWS SDKs

다음은 이 항목에 설명된 기능 및 설정을 SDKs 지원합니다. 모든 일부 예외가 기록됩니다. 모든 JVM 시스템 속성 설정은 에서 지원됩니다. AWS SDK for Java 그리고 AWS SDK for Kotlin 오직.

SDK	지	참고 또는 추가 정보
AWS CLI v2	예	
SDKC++의 경우	예	공유 config 파일은 지원되지 않습니다.
SDK고 V2 (1.x) 용	예	
SDKGo 1.x (V1) 용	아 니 요	

SDK	지 운	참고 또는 추가 정보
SDK자바 2.x의 경우	부	공유 config 파일 설정이 지원되지 않습니다. 환경 변수는 지원되지 않습니다.
SDKJava 1.x의 경우	아 니 요	
SDK3.x용 JavaScript	예	
SDK2.x의 경우 JavaScript	아 니 요	
SDK코틀린의 경우	예	
SDK왜냐하면. NET3.x	예	환경 변수는 지원되지 않습니다.
SDK3.x의 경우 PHP	예	
SDK파이썬용 (보토3)	예	
SDK루비 3.x의 경우	예	
SDK러스트용	예	
SDK스위프트용	아 니 요	
를 위한 도구 PowerShell	아 니 요	

Amazon EC2 인스턴스 메타데이터

EC2Amazon은 인스턴스 메타데이터 서비스 (IMDS) 라는 인스턴스 기반 서비스를 제공합니다. 이 서비스에 대한 자세한 내용은 Amazon EC2 사용 설명서의 [인스턴스 메타데이터](#) 사용을 참조하십시오. IAM역할로 구성된 Amazon 인스턴스에서 자격 증명을 검색하려는 경우 EC2 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
```

와의 호환성 AWS SDKs

다음은 이 항목에 설명된 기능 및 설정을 SDKs 지원합니다. 모든 일부 예외가 기록됩니다. 모든 JVM 시스템 속성 설정은 에서 지원됩니다. AWS SDK for Java 그리고 AWS SDK for Kotlin 오직.

SDK	지 원	참고 또는 추가 정보
AWS CLI v2	예	
SDKC++의 경우	아 니 요	
SDK고 V2 (1.x) 용	아 니 요	
SDK고 1.x (V1) 용	아 니 요	
SDK자바 2.x의 경우	아 니 요	
SDK자바 1.x의 경우	부 편	AWS_METADATA_SERVICE_TIMEOUT 만 지원됩니다.
SDK3.x용 JavaScript	아 니 요	
SDK2.x의 경우 JavaScript	아 니 요	

SDK	자 문	참고 또는 추가 정보
SDK코틀린의 경우	아 니 요	
SDK왜냐하면. NET3.x	아 니 요	
SDK3.x의 경우 PHP	예	
SDK파이썬용 (보토3)	예	
SDK루비 3.x의 경우	아 니 요	
SDK러스트용	아 니 요	
SDK스위프트용	아 니 요	
를 위한 도구 PowerShell	아 니 요	

Amazon S3 액세스 포인트

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. 와 함께 AWS SDKs, 버킷 이름을 명시적으로 지정하는 대신 버킷 필드에 있는 액세스 포인트 Amazon Resource Names (ARNs) 를 API 작업에 사용할 수 있습니다. 액세스 포인트를 ARN 사용하여 버킷에

서 객체를 가져오거나 [GetObject](#)를 사용하여 버킷에 객체를 [PutObject](#) 추가하는 등의 특정 작업에 사용됩니다. ARN

Amazon S3 액세스 포인트에 대해 자세히 알아보려면 Amazon S3 사용 설명서의 [액세스 포인트 사용](#)을 참조하십시오. ARNs

다음을 사용하여 이 기능을 구성하십시오.

s3_use_arn_region- 공유 AWS config파일 설정, **AWS_S3_USE_ARN_REGION** - 환경 변수, **aws.s3UseArnRegion**- JVM 시스템 속성: 자바/코틀린 전용, 코드에서 값을 직접 구성하려면 해당 값을 직접 참조하십시오. SDK

이 설정은 액세스 포인트 SDK 사용 여부를 제어합니다. ARN AWS 리전 요청에 대한 리전 엔드포인트를 구축합니다. 는 SDK 다음을 검증합니다. ARN AWS 리전 동일한 서비스를 제공합니다. AWS 클라이언트가 구성한 파티션 AWS 리전 실패할 가능성이 높은 파티션 간 호출을 방지하기 위해서입니다. 다중 정의의 경우 코드로 구성된 설정이 우선 적용되고 환경 변수 설정이 그 뒤를 따릅니다.

기본값: false

유효한 값:

- **true**— 's를 SDK 사용합니다. ARN AWS 리전 구성된 클라이언트 대신 엔드포인트를 구성하는 경우 AWS 리전. 예외: 클라이언트가 구성된 경우 AWS 리전 는 FIPS AWS 리전그러면 해당 값과 일치해야 합니다. ARN AWS 리전. 그렇지 않으면 오류가 발생합니다.
- **false**— 클라이언트의 구성을 SDK 사용합니다. AWS 리전 엔드포인트를 구성할 때.

와의 호환성 AWS SDKs

다음은 이 항목에 설명된 기능 및 설정을 SDKs 지원합니다. 모든 일부 예외가 기록됩니다. 모든 JVM 시스템 속성 설정은 에서 지원됩니다. AWS SDK for Java 그리고 AWS SDK for Kotlin 오직.

SDK	지 참고 또는 추가 정보 운
AWS CLI v2	예
SDKC++의 경우	예
SDK고 V2 (1.x) 용	예

SDK	지 운	참고 또는 추가 정보
SDK고 1.x (V1) 용	예	공유 config 파일 설정을 사용하려면 구성 파일에서 로드를 켜야 합니다. 세션 을 참조하십시오.
SDK자바 2.x의 경우	예	
SDK자바 1.x의 경우	예	JVM시스템 속성은 지원되지 않습니다.
SDK JavaScript 3.x용	예	
SDK2.x의 경우 JavaScript	예	
SDK코틀린의 경우	예	
SDK왜냐하면. NET3.x	예	표준 우선 순위를 따르지 않습니다. 공유 config 파일 값이 환경 변수보다 우선합니다.
SDK3.x의 경우 PHP	예	
SDK파이썬용 (보토3)	예	
SDK루비 3.x의 경우	예	
SDK러스트용	아 니 요	
SDK스위프트용	아 니 요	
를 위한 도구 PowerShell	예	표준 우선 순위를 따르지 않습니다. 공유 config 파일 값이 환경 변수보다 우선합니다.

Amazon S3 다중 리전 액세스 포인트

Amazon S3 다중 지역 액세스 포인트는 애플리케이션이 여러 곳에 위치한 Amazon S3 버킷의 요청을 처리하는 데 사용할 수 있는 글로벌 엔드포인트를 제공합니다. AWS 리전. 다중 지역 액세스 포인트를

사용하여 단일 지역에서 사용되는 것과 동일한 아키텍처로 다중 지역 애플리케이션을 구축한 다음 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 포인트](#)를 참조하십시오.

다중 지역 액세스 포인트 Amazon 리소스 이름 (ARNs) 에 대한 자세한 내용은 Amazon S3 사용 설명서의 [다중 지역 액세스 포인트를 사용하여 요청하기](#)를 참조하십시오.

다중 리전 액세스 포인트 생성에 대한 자세한 내용을 알아보려면 Amazon S3 사용 설명서의 [Amazon S3의 다중 리전 액세스 포인트 관리](#)를 참조하십시오.

SigV4A 알고리즘은 글로벌 리전 요청에 서명하는 데 사용되는 서명 구현입니다. 이 알고리즘은 에 대한 종속성을 SDK 통해 얻을 수 있습니다. [AWS 공통 런타임\(CRT\) 라이브러리](#)

다음을 사용하여 이 기능을 구성하십시오.

s3_disable_multiregion_access_points- 공유 AWS config파일 설정, **AWS_S3_DISABLE_MULTIREGION_ACCESS_POINTS** - 환경 변수, **aws.s3DisableMultiRegionAccessPoints**- JVM 시스템 속성: 자바/코틀린 전용, 코드에서 값을 직접 구성하려면 해당 값을 직접 참조하십시오. SDK

이 설정은 SDK 사용자가 잠재적으로 지역 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.

와의 호환성 AWS SDKs

다음은 이 항목에 설명된 기능 및 설정을 SDKs 지원합니다. 모든 일부 예외가 기록됩니다. 모든 JVM 시스템 속성 설정은 에서 지원됩니다. AWS SDK for Java 그리고 AWS SDK for Kotlin 오직.

SDK	지 참고 또는 추가 정보 은
AWS CLI v2	예

SDK	지 운	참고 또는 추가 정보
SDKC++의 경우	예	
SDK고 V2 (1.x) 용	예	
SDKGo 1.x (V1) 용	아 니 요	
SDK자바 2.x의 경우	예	
SDK자바 1.x의 경우	아 니 요	
SDK3.x용 JavaScript	예	
SDK2.x의 경우 JavaScript	아 니 요	
SDK코틀린의 경우	예	
SDK왜냐하면. NET3.x	예	
SDK3.x의 경우 PHP	예	
SDK파이썬용 (보토3)	예	
SDK루비 3.x의 경우	예	
SDK러스트용	예	
SDK스위프트용	아 니 요	
를 위한 도구 PowerShell	예	

AWS 리전

AWS 리전은 로 작업할 때 이해해야 할 중요한 개념입니다 AWS 서비스.

를 사용하면 특정 지리적 영역에 물리적으로 AWS 서비스 있는 에 액세스할 AWS 리전 수 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 중단에 영향을 받지 않는 중복 리소스를 생성할 수 있습니다.

대부분의 AWS 서비스 요청은 특정 지리적 리전과 연결됩니다. 한 리전에서 생성한 리소스는 AWS 서비스에서 제공하는 복제 기능을 명시적으로 사용하지 않는 한 다른 리전에 존재하지 않습니다. 예를 들어 Amazon S3 및 Amazon은 리전 간 복제를 EC2 지원하지 않습니다. 와 같은 일부 서비스에는 리전 리소스 IAM가 없습니다.

AWS 일반 참조에는 다음 정보가 포함됩니다.

- 리전과 엔드포인트 간의 관계를 이해하고 기존 리전 엔드포인트 목록을 보려면 [AWS 서비스 엔드포인트](#)를 참조하십시오.
- 각각의 AWS 서비스에 대해 지원되는 모든 리전 및 엔드포인트의 현재 목록을 보려면 [서비스 엔드포인트 및 할당량](#)을 참조하십시오.

서비스 클라이언트 생성

프로그래밍 방식으로 에 액세스하려면 각 에 대해 클라이언트 클래스/객체를 AWS 서비스 SDKs 사용합니다 AWS 서비스. EC2 예를 들어 애플리케이션이 Amazon 에 액세스해야 하는 경우 애플리케이션은 해당 서비스와 인터페이스할 Amazon EC2 클라이언트 객체를 생성합니다.

코드 자체에서 클라이언트에 대해 명시적으로 지정된 리전이 없는 경우 클라이언트는 다음 region 설정을 통해 설정된 리전을 기본적으로 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 클라이언트의 인스턴스화 중에 에 따라 지정됩니다 SDK(특정 SDK 가이드 또는 SDK코드 기반 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 AWS 리전 사용할 기본값을 지정합니다. 이 리전은 사용할 특정 리전과 함께 제공되지 않은 SDK 서비스 요청에 사용됩니다.

기본값: 없음. 이 값을 명시적으로 지정해야 합니다.

유효값:

- AWS 일반 참조의 [AWS 서비스 엔드포인트](#)에 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 서비스의 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
```

대부분 애플리케이션 코드 내에서 기본 리전을 설정하는 데 사용할 수 있는 '구성' 객체가 SDKs 있습니다. 자세한 내용은 특정 AWS SDK 개발자 안내서를 참조하세요.

와의 호환성 AWS SDKs

다음은 이 주제에 설명된 기능 및 설정을 SDKs 지원합니다. 모든 일부 예외가 기록됩니다. 모든 JVM 시스템 속성 설정은 AWS SDK for Java 및 AWS SDK for Kotlin 에서만 지원됩니다.

SDK	지참고 또는 추가 정보 운
AWS CLI v2	예 AWS CLI v2는 의 값을 의 값AWS_REGION 보다 먼저 사용합니 다AWS_DEFAULT_REGION (두 변수 모두 선택됨).
AWS CLI v1	예 AWS CLI v1은 이 용도로 AWS_DEFAULT_REGION 이름이 지정된 환경 변수를 사용합니다.
SDK C++용	예
SDK Go V2용(1.x)	예
SDK Go 1.x(V1)용	예 공유 config 파일 설정을 사용하려면 구성 파일에서 로드를 켜야 합니다. 세션 을 참조하십시오.
SDK Java 2.x용	예
SDK Java 1.x용	예
SDK JavaScript 3.x용	예
SDK JavaScript 2.x용	예
SDK Kotlin용	예
SDK 용.NET 3.x	예
SDK 3.xPHP용	예
SDK Python용(Boto3)	예 이를 AWS_DEFAULT_REGION 위해 라는 환경 변수를 SDK 사용합니다.
SDK Ruby 3.x용	예
SDK Rust용	예
SDK Swift용	예
용 도구 PowerShell	예

AWS STS 리전 엔드포인트

AWS Security Token Service (AWS STS)는 글로벌 및 리전 서비스로 모두 사용할 수 있습니다. 일부 SDKs 및 AWS 는 기본적으로 글로벌 서비스 엔드포인트(<https://sts.amazonaws.com>)를 CLIs 사용하는 반면, 일부는 리전 서비스 엔드포인트()를 사용합니다. https://sts.{region_identifier}.{partition_domain}. 글로벌 요청은 미국 동부(버지니아 북부) 리전에 매핑됩니다. AWS STS 엔드포인트에 대한 자세한 내용은 AWS Security Token Service API 참조의 [엔드포인트](#)를 참조하세요. 또는 AWS Identity and Access Management 사용 설명서의 [AWS STS 에서 관리 AWS 리전](#) 방법을 알아봅니다.

가능하면 리전 엔드포인트를 사용하고 를 구성하는 것이 AWS 좋습니다. [AWS 리전](#). 상용이 아닌 [파티션](#)의 고객은 리전 엔드포인트를 사용해야 합니다. 모든 SDKs 및 도구가 이 설정을 지원하는 것은 아니지만 모든 도구에는 글로벌 및 리전 엔드포인트에 대해 정의된 동작이 있습니다. 자세한 내용은 다음 단원을 참조하십시오.

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 대화하는 데 사용하는 엔드포인트를 결정하는 AWS 서비스 방법을 지정합니다. AWS STS.

기본값: legacy

Note

2022년 7월 이후에 릴리스되는 모든 새 SDK 메이저 버전은 기본적으로 로 설정됩니다. 새 SDK 메이저 버전은 이 설정을 제거하고 regional 동작을 사용할 수 있습니다. 이 변경으로 인한 향후 영향을 줄이려면 가능하면 사용자 애플리케이션에서 regional을 사용하여 시작하는 것이 좋습니다.

유효한 값:, (권장 값: regional)

- **legacy** - 글로벌 AWS STS 엔드포인트인 를 사용합니다. sts.amazonaws.com.
- **regional** - SDK 또는 도구는 현재 구성된 리전에 대해 항상 AWS STS 엔드포인트를 사용합니다. 예를 들어 클라이언트가 를 사용하도록 구성된 경우 us-west-2에 대한 모든 호출 AWS STS 은 글로벌 엔드포인트 sts.us-west-2.amazonaws.com 대신 리전 sts.amazonaws.com 엔

드포인트 에 수행됩니다. 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
```

와의 호환성 AWS SDKs

Note

가능하면 리전 엔드포인트를 사용하고 를 구성하는 것이 AWS 좋습니다 [AWS 리전](#).

다음 표에는 SDK 또는 도구에 대한 요약이 나와 있습니다.

- **설정 지원** : 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** : 변경 설정을 사용할 수 없는 경우에도 클라이언트가 사용하는 기본 엔드포인트입니다.
- **서비스 클라이언트 대체 동작** : 리전 엔드포인트를 사용해야 하지만 리전이 구성되지 않은 경우 가 SDK 수행하는 작업입니다. 이는 기본값으로 인해 리전 엔드포인트를 사용하고 있는지 또는 설정에 의해 `regional`가 선택되어 있는지 여부에 관계없이 동작입니다.

또한 이 테이블은 다음 값도 사용합니다.

- **글로벌 엔드포인트**: `https://sts.amazonaws.com`.

- 리전 엔드포인트 : 애플리케이션에서 [AWS 리전](#) 사용하는 구성된 를 기반으로 합니다.
- **us-east-1** (리전): us-east-1 리전 엔드포인트를 사용하지만 일반적인 글로벌 요청보다 세션 토큰이 더 길니다.

SDK	기본 설정 값	기본 서비스 클라이언트 대상 STS 엔드포인트	서비스 클라이언트 대체 동작	참고 또는 추가 정보
AWS CLI v2	이 N/A L D	리전 엔드포인트	글로벌 엔드포인트	
AWS CLI v1	0: legacy	글로벌 엔드포인트	글로벌 엔드포인트	
SDK C++용	이 N/A L D	리전 엔드포인트	us-east-1 (리전)	
SDK Go V2 용(1.x)	이 N/A L D	리전 엔드포인트	요청 실패	
SDK Go 1.x(V1)용	0: legacy	글로벌 엔드포인트	글로벌 엔드포인트	공유 config 파일 설정을 사용하려면 구성 파일에서 로드를 켜야 합니다. 세션 을 참조하십시오.
SDK Java 2.x 용	이 N/A L D	리전 엔드포인트	요청 실패	리전이 구성되지 않은 경우 AssumeRole 및 AssumeRoleWithWebIdentity 는 글로벌 STS 엔드포인트를 사용합니다.
SDK Java 1.x 용	0: legacy	글로벌 엔드포인트	글로벌 엔드포인트	

SDK	기본 설정 값	기본 서비스 클라이언트 대상 STS 엔드포인트	서비스 클라이언트 대체 동작	참고 또는 추가 정보
SDK JavaScript 3.x용	이 리전 엔드포인트	리전 엔드포인트	요청 실패	
SDK JavaScript 2.x용	0: legacy	글로벌 엔드포인트	글로벌 엔드포인트	
SDK Kotlin용	이 리전 엔드포인트	리전 엔드포인트	글로벌 엔드포인트	
SDK 용.NET 3.x	0: legacy	글로벌 엔드포인트	글로벌 엔드포인트	
SDK 3.xPHP 용	0: legacy	글로벌 엔드포인트	요청 실패	
SDK Python 용(Boto3)	0: legacy	글로벌 엔드포인트	글로벌 엔드포인트	
SDK Ruby 3.x용	0: regional	리전 엔드포인트	요청 실패	
SDK Rust용	이 리전 엔드포인트	리전 엔드포인트	요청 실패	
SDK Swift용	이 리전 엔드포인트	리전 엔드포인트	요청 실패	
용 도구 PowerShell	0: legacy	글로벌 엔드포인트	글로벌 엔드포인트	

듀얼 스택 및 엔드포인트 FIPS

다음을 사용하여 이 기능을 구성하십시오.

use_dualstack_endpoint- 공유 AWS `config` 파일 설정, **AWS_USE_DUALSTACK_ENDPOINT** - 환경 변수, **aws.useDualstackEndpoint**- JVM 시스템 속성: 자바/코틀린 전용

이중 스택 엔드포인트로 요청을 SDK 보낼지 여부를 설정합니다. IPv6 트래픽을 모두 IPv4 지원하는 이중 스택 엔드포인트에 대한 자세한 내용은 Amazon Simple Storage Service 사용 설명서의 [Amazon S3 이중 스택 엔드포인트](#) 사용을 참조하십시오. 이중 스택 엔드포인트는 일부 리전에 사용할 수 있는 서비스입니다.

기본값: `false`

유효한 값:

- **true**— SDK or 도구는 이중 스택 엔드포인트를 사용하여 네트워크 요청을 시도합니다. 서비스에 대한 이중 스택 엔드포인트가 없는 경우 및/또는 AWS 리전요청이 실패합니다.
- **false**— SDK or 도구는 이중 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: 자바/코틀린 전용

SDK 또는 도구가 호환 엔드포인트로 요청을 보낼지 여부를 설정합니다. FIPS 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. 표준과 다릅니다. AWS 엔드포인트는 FIPS 140-2를 준수하는 TLS 소프트웨어 라이브러리를 사용합니다. FIPS 이 설정이 활성화되어 있는데 해당 서비스에 대한 FIPS 엔드포인트가 없는 경우 AWS 리전, AWS 통화가 실패할 수 있습니다. [서비스별 엔드포인트](#) 그리고 `--endpoint-url` 옵션 AWS Command Line Interface 이 설정을 재정의하십시오.

FIPS 엔드포인트를 지정하는 다른 방법에 대해 자세히 알아보려면 AWS 리전 [서비스별 FIPS 엔드포인트](#)를 참조하십시오. Amazon Elastic Compute Cloud 서비스 엔드포인트에 대한 자세한 내용은 Amazon [레퍼런스의 이중 스택 \(IPv4 및 IPv6\) 엔드포인트](#)를 참조하십시오. EC2 API

기본값: `false`

유효한 값:

- **true**— SDK or 도구는 규정을 준수하는 엔드포인트로 요청을 보냅니다. FIPS
- **false**— SDK or 도구는 호환 엔드포인트에 FIPS 요청을 보내지 않습니다.

다음과 호환됩니다. AWS SDKs

다음은 이 항목에 설명된 기능 및 설정을 SDKs 지원합니다. 모든 일부 예외가 기록됩니다. 모든 JVM 시스템 속성 설정은 에서 지원됩니다. AWS SDK for Java 그리고 AWS SDK for Kotlin 오직.

SDK	지	참고 또는 추가 정보
AWS CLI v2	예	
SDKC++의 경우	예	
SDK고 V2 (1.x) 용	예	
SDKGo 1.x (V1) 용	예	공유 config 파일 설정을 사용하려면 구성 파일에서 로드를 켜야 합니다. 세션 을 참조하십시오.
SDK자바 2.x의 경우	예	
SDK자바 1.x의 경우	아 니 요	
SDK3.x용 JavaScript	예	
SDK2.x의 경우 JavaScript	예	
SDK코틀린의 경우	예	
SDK왜냐하면. NET3.x	예	
SDK3.x의 경우 PHP	예	
SDK파이썬용 (보토3)	예	
SDK루비 3.x의 경우	예	
SDK러스트용	예	
SDK스위프트용	예	

SDK	지 참고 또는 추가 정보 온
를 위한 도구 PowerShell	예

엔드포인트 검색

SDKs엔드포인트 검색을 사용하여 서비스 엔드포인트에 URLs 액세스하고 (다양한 리소스에 액세스할 수 있음) 다음과 같은 유연성을 유지합니다. AWS URLs필요에 따라 변경할 수 있습니다. 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 SDK 는 사용 가능한 엔드포인트를 검색하도록 DescribeEndpoints 요청한 다음 해당 엔드포인트를 사용하여 또는 같은 특정 작업을 완료합니다. CreateDatabase CreateTable

다음을 사용하여 이 기능을 구성하십시오.

endpoint_discovery_enabled- 공유 AWS **config**파일 설정,

AWS_ENABLE_ENDPOINT_DISCOVERY - 환경 변수, **aws.endpointDiscoveryEnabled**- JVM 시스템 속성: 자바/코틀린 전용, 코드에서 값을 직접 구성하려면 해당 값을 직접 참조하십시오. SDK

DynamoDB에 대한 엔드포인트 검색을 켜거나 끕니다.

타임스트림에서는 엔드포인트 검색이 필요하고 Amazon DynamoDB에서는 선택 사항입니다. 이 설정의 기본값은 서비스에 엔드포인트 검색이 필요한지 여부에 false 따라 둘 중 하나 true 또는 두 가지로 설정됩니다. 타임스트림 요청의 기본값은 이고 Amazon DynamoDB 요청은 기본적으로입니다. true false

유효한 값:

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

와의 호환성 AWS SDKs

다음은 이 항목에 설명된 기능 및 설정을 SDKs 지원합니다. 모든 일부 예외가 기록됩니다. 모든 JVM 시스템 속성 설정은 에서 지원됩니다. AWS SDK for Java 그리고 AWS SDK for Kotlin 오직.

SDK	지	참고 또는 추가 정보
AWS CLI v2	예	
SDKC++의 경우	예	
SDK고 V2 (1.x) 용	예	
SDKGo 1.x (V1) 용	예	공유 config 파일 설정을 사용하려면 구성 파일에서 로드를 켜야 합니다. 세션 을 참조하십시오.
SDK자바 2.x의 경우	예	Java 2.x의 AWS_ENDPOINT_DISCOVERY_ENABLED 경우 환경 변수 이름으로 사용합니다. SDK
SDK자바 1.x의 경우	부 분	JVM시스템 속성은 지원되지 않습니다.
SDK JavaScript 3.x의 경우	예	
SDK2.x의 경우 JavaScript	예	
SDK코틀린의 경우	예	
SDK왜냐하면. NET3.x	예	
SDK3.x의 경우 PHP	예	
SDK파이썬용 (보토3)	예	
SDK루비 3.x의 경우	예	
SDK러스트용	부 분	타임스트림에만 지원됩니다.

SDK	지참고 또는 추가 정보 온
SDK스위프트용	아 니 요
를 위한 도구 PowerShell	예

일반 구성 설정

SDKs전체 SDK 동작을 구성하는 몇 가지 일반 설정을 지원합니다.

다음을 사용하여 이 기능을 구성하십시오.

api_versions- 공유 AWS config파일 설정

약간 AWS 서비스는 API 이전 버전과의 호환성을 지원하기 위해 여러 버전을 유지 관리합니다. 기본적으로, SDK AWS CLI 작업에는 사용 가능한 최신 API 버전이 사용됩니다. 요청에 사용할 특정 API 버전을 요구하려면 프로필에 api_versions 설정을 포함하세요.

기본값: 없음. (API에서는 최신 버전을 사용합니다SDK.)

유효한 값: 이 설정은 각각 하나를 식별하는 하나 이상의 들여쓰기된 줄이 뒤에 오는 중첩 설정입니다. AWS 사용할 서비스 및 API 버전에 대한 설명서를 참조하십시오. AWS 어떤 API 버전을 사용할 수 있는지 알아보려면 서비스를 이용하십시오.

이 예제에서는 두 API 버전의 특정 버전을 설정합니다. AWS config파일 내 서비스. 이러한 API 버전은 이러한 설정이 포함된 프로필에서 실행되는 명령에만 사용됩니다. 다른 서비스의 명령은 해당 서비스의 최신 버전을 사용합니다API.

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

ca_bundle- 공유 AWS config파일 설정, AWS_CA_BUNDLE - 환경 변수

SSL/TLS연결을 설정할 때 사용할 사용자 지정 인증서 번들 (.pem확장명이 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 체제의 경로 처리 방식과 경로 문자 이스케이프 처리 방식의 차이로 인해 Windows의 config 파일에서 이 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

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

기본값: json

유효한 값:

- **json**— 출력 형식은 [JSON](#) 문자열 형식입니다.
- **yaml**— 출력 형식은 문자열 형식입니다. [YAML](#)
- **yaml-stream**— 출력은 스트리밍되고 문자열 형식이 지정됩니다. [YAML](#) 스트리밍을 통해 대용량 데이터 유형을 빠르게 처리할 수 있습니다.
- **text** - 출력은 여러 줄의 탭으로 구분된 문자열 값으로 형식이 지정됩니다. 출력을 grep, sed 또는 awk와 같은 텍스트 프로세서로 전달하는 데 사용할 수 있습니다.

- **table** - 출력은 셀 테두리를 형성하기 위해 +- 문자를 사용하여 표로 형식이 지정됩니다. 일반적으로 읽기는 쉽지만 프로그래밍 방식으로는 유용하지 않은 '인간 친화적' 형식으로 정보를 표시합니다.

parameter_validation- 공유 AWS config 파일 설정

SDK 또는 도구가 명령줄 매개 변수를 다음으로 보내기 전에 유효성 검사를 시도할지 여부를 지정합니다. AWS 서비스 엔드포인트.

기본값: true

유효값:

- **true** - 기본값입니다. SDK or 도구는 명령줄 매개 변수의 클라이언트 측 검증을 수행합니다. 이를 통해 SDK or 도구는 매개 변수가 유효한지 확인하고 일부 오류를 포착할 수 있습니다. SDK or 도구는 요청을 보내기 전에 유효하지 않은 요청을 거부할 수 있습니다. AWS 서비스 엔드포인트.
- **false**— SDK or 도구는 명령줄 매개 변수를 로 보내기 전에 해당 매개 변수의 유효성을 검사하지 않습니다. AWS 서비스 엔드포인트. The AWS 서비스 엔드포인트는 모든 요청의 유효성을 검사하고 유효하지 않은 요청은 거부할 책임이 있습니다.

와의 호환성 AWS SDKs

다음은 이 항목에 설명된 기능 및 설정을 SDKs 지원합니다. 모든 일부 예외가 기록됩니다. 모든 JVM 시스템 속성 설정은 에서 지원됩니다. AWS SDK for Java 그리고 AWS SDK for Kotlin 오직.

SDK	지 운	참고 또는 추가 정보
AWS CLI v2	부 분	api_versions 이 지원되지 않음.
SDKC++의 경우	예	
SDKGo V2 (1.x) 용	부 분	api_versions 및 parameter_validation 이 지원되지 않음.
SDKGo 1.x (V1) 용	부 분	api_versions 및 parameter_validation 이 지원되지 않음. 공유 config 파일 설정을 사용하려면 구성 파일에서 로드를 켜야 합니다. 세션 을 참조하십시오.

SDK	자 운 참고 또는 추가 정보
SDK자바 2.x의 경우	아 니 요
SDK자바 1.x의 경우	아 니 요
SDK3.x용 JavaScript	예
SDK2.x의 경우 JavaScript	예
SDK코틀린의 경우	아 니 요
SDK왜냐하면. NET3.x	아 니 요
SDK3.x의 경우 PHP	예
SDK파이썬용 (보토3)	예
SDK루비 3.x의 경우	예
SDK러스트용	아 니 요
SDK스위프트용	아 니 요

SDK	지 운	참고 또는 추가 정보
를 위한 도구 PowerShell	아 니 요	

IMDS 클라이언트

SDKs 세션 지향 요청을 사용하여 인스턴스 메타데이터 서비스 버전 2(IMDSv2) 클라이언트를 구현합니다. 에 대한 자세한 내용은 Amazon EC2 사용 설명서의 [사용을 IMDSv2](#) IMDSv2참조하세요. IMDS 클라이언트는 SDK 코드 기반에서 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

토큰TTL의 .

기본값: 21,600초(6시간, 최대 할당 시간).

유효한 값: 숫자.

endpoint - 클라이언트 구성 객체 멤버

의 엔드포인트입니다IMDS.

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.

다음 옵션은 대부분의 에서 지원됩니다SDKs. 자세한 내용은 특정 SDK 코드 기반을 참조하세요.

endpoint_mode- 클라이언트 구성 객체 멤버

의 엔드포인트 모드입니다IMDS.

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: 와 같은 유효한 I/O 스트림입니다STDOUT.

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: 에 따라 다릅니다SDK.

유효한 값: 에 따라 다릅니다SDK. 숫자 값이거나 사용자 지정 함수 호출이 될 수 있습니다.

와의 호환성 AWS SDKs

다음은 이 주제에 설명된 기능 및 설정을 SDKs 지원합니다. 모든 일부 예외가 기록됩니다. 모든 JVM 시스템 속성 설정은 AWS SDK for Java 및 AWS SDK for Kotlin 에서만 지원됩니다.

SDK	지 원	참고 또는 추가 정보
AWS CLI v2	예	
SDK C++용	아 니 요	
SDK Go V2용(1.x)	예	
SDK Go 1.x(V1)용	예	
SDK Java 2.x용	예	
SDK Java 1.x용	예	
SDK JavaScript 3.x용	예	
SDK JavaScript 2.x용	예	
SDK Kotlin용	아 니 요	
SDK 용.NET 3.x	예	
SDK 3.xPHP용	예	
SDK Python용(Boto3)	예	
SDK Ruby 3.x용	예	
SDK Rust용	예	
SDK Swift용	예	

SDK	지참고 또는 추가 정보 온
용 도구 PowerShell	예

재시도 동작

재시도 동작에는 에 대한 요청으로 인한 장애 복구 SDKs 시도 방법에 대한 설정이 포함됩니다. AWS 서비스.

다음을 사용하여 이 기능을 구성하십시오.

retry_mode- 공유 AWS **config**파일 설정, **AWS_RETRY_MODE** - 환경 변수, **aws.retryMode**- JVM 시스템 속성: 자바/코틀린 전용

SDK또는 개발자 도구가 재시도를 시도하는 방법을 지정합니다.

기본값: 이 값은 사용자 SDK 고유의 값입니다. 특정 SDK 가이드 또는 사용 중인 코드베이스에서 기본값을 확인하세요 `retry_mode`. SDK

유효한 값:

- **standard**— (권장) 전체에 걸쳐 권장되는 재시도 규칙 세트 AWS SDKs. 이 모드에는 재시도 되는 표준 오류 집합이 포함되며 가용성과 안정성을 극대화하기 위해 재시도 횟수를 자동으로 조정합니다. 이 모드는 멀티테넌트 애플리케이션에서 사용하기에 안전합니다. 명시적으로 `max_attempts`으로 구성되어 있지 않으면 이 모드의 기본 최대 시도 횟수는 3회입니다.
- **adaptive**— 표준 모드의 기능과 클라이언트측 자동 속도 제한을 포함하는 재시도 모드로서, 특수한 사용 사례에만 적합합니다. 응용 프로그램 테넌트를 분리하는 데 주의를 기울이지 않는 한 다중 테넌트 응용 프로그램에는 이 재시도 모드를 사용하지 않는 것이 좋습니다. 자세한 내용은 [모드 선택 standard 및 adaptive 재시도 모드](#) 섹션을 참조하세요. 이 모드는 실험용이며 향후 동작이 변경될 수 있습니다.
- **legacy**— (권장되지 않음) 해당 사용자에게만 해당됩니다 SDK (특정 SDK 가이드 또는 사용 중인 코드베이스를 확인하세요). SDK

max_attempts- 공유 AWS **config**파일 설정, **AWS_MAX_ATTEMPTS** - 환경 변수, **aws.maxAttempts**- JVM 시스템 속성: 자바/코틀린 전용

요청에 대한 최대 시도 횟수를 지정합니다.

기본값: 이 값이 지정되지 않은 경우 기본값은 `retry_mode` 설정 값에 따라 달라집니다.

- `retry_mode`가 `legacy` — 사용자 고유의 기본값을 사용합니다 SDK (기본값은 특정 SDK 가이드 또는 사용 중인 코드베이스를 확인하세요). `SDK max_attempts`
- `retry_mode`가 `standard`인 경우 — 세 번 시도합니다.
- `retry_mode`가 `adaptive`인 경우 — 세 번 시도합니다.

유효한 값: 0보다 큰 숫자.

모드 선택 **standard** 및 **adaptive** 재시도 모드

용도에 더 적합하다고 확신할 수 없는 경우가 아니면 `standard` 재시도 모드를 사용하는 것이 좋습니다. `adaptive`

Note

이 `adaptive` 모드에서는 백엔드 서비스가 요청을 제한할 수 있는 범위를 기반으로 클라이언트를 풀링한다고 가정합니다. 이렇게 하지 않으면 두 리소스에 동일한 클라이언트를 사용하는 경우 한 리소스의 병목 현상으로 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.
정전 시 재시도를 방지하기 위해 회로 차단이 SDK 지원됩니다.	정전 시 재시도를 방지하기 위해 회로 차단을 지원합니다. SDK
장애 발생 시 지터링 지수 백오프를 사용합니다.	지연 시간이 증가할 수 있는 대신 동적 백오프 기간을 사용하여 실패한 요청 수를 최소화하려고 합니다.
첫 번째 요청 시도는 절대 지연시키지 말고 재시도만 지연하십시오.	초기 요청 시도를 제한하거나 지연시킬 수 있습니다.

adaptive모드를 사용하는 경우 애플리케이션은 병목 현상이 발생할 수 있는 각 리소스를 중심으로 설계된 클라이언트를 구성해야 합니다. 이 경우 리소스는 단순히 각 리소스만 고려하는 것보다 더 세밀하게 조정됩니다. AWS 서비스. AWS 서비스 요청을 제한하는 데 사용하는 추가 차원이 있을 수 있습니다. Amazon DynamoDB 서비스를 예로 들어 보겠습니다. DynamoDB 사용 AWS 리전 또한 요청을 제한하기 위해 액세스할 수 있는 테이블도 있습니다. 즉, 코드가 액세스하는 테이블 하나가 다른 테이블보다 병목 현상이 더 심할 수 있습니다. 코드가 동일한 클라이언트를 사용하여 모든 테이블에 액세스하고 해당 테이블 중 하나에 대한 요청이 병목 현상을 겪는 경우 적응형 재시도 모드를 사용하면 모든 테이블의 요청 속도가 감소합니다. 코드는 Region-and-table 쌍당 하나의 클라이언트를 포함하도록 설계되어야 합니다. adaptive모드를 사용할 때 예상치 못한 지연이 발생하는 경우 구체적인 내용을 확인하세요. AWS 사용 중인 서비스에 대한 설명서 가이드.

재시도 모드 구현 세부 정보

The AWS SDKs [토큰 버킷](#)을 사용하여 요청을 재시도할지 여부와 (adaptive재시도 모드의 경우) 요청을 얼마나 빨리 전송할지 결정하십시오. 는 재시도 토큰 버킷과 요청 속도 토큰 버킷이라는 두 개의 토큰 버킷을 사용합니다. SDK

- 재시도 토큰 버킷은 정전 시 업스트림 및 다운스트림 서비스를 보호하기 위해 재시도를 일시적으로 비활성화할지 여부를 결정하는 데 사용됩니다. SDK 재시도가 시도되기 전에 버킷에서 토큰을 획득하고, 요청이 성공하면 토큰이 버킷으로 반환됩니다. 재시도 시 버킷이 비어 있는 경우는 요청을 재시도하지 않습니다. SDK
- 요청 속도 토큰 버킷은 adaptive 재시도 모드에서만 요청을 전송할 속도를 결정하는 데 사용됩니다. 토큰은 요청이 전송되기 전에 버킷에서 획득되며, 서비스에서 반환한 제한 응답에 따라 동적으로 결정된 속도로 토큰이 버킷으로 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 재시도 모드에서만 사용됩니다. adaptive 이 단계에서는 요청 속도 토큰 버킷에서 토큰을 획득합니다. 토큰을 사용할 수 없는 경우 토큰을 사용할 수 있을 때까지 기다립니다. 기다리는 대신 요청을 실패시키는 구성 옵션이 SDK 있을 수 있습니다. 버킷의 토큰은 클라이언트가 수신한 제한 응답 수에 따라 동적으로 결정되는 비율로 리필됩니다.

SendHttpRequest:

이 단계에서는 요청을 다음으로 전송합니다. AWS. 대부분 AWS SDKsHTTP요청을 할 때 연결 풀을 사용하는 HTTP 라이브러리를 사용하여 기존 연결을 재사용하십시오. 일반적으로 전송률 조절 오류로 인해 요청이 실패하면 연결이 재사용되지만 일시적 오류로 인해 요청이 실패하는 경우에는 다시 사용되지 않습니다.

RequestBookkeeping:

요청이 성공하면 토큰이 토큰 버킷에 추가됩니다. adaptive재시도 모드의 경우에만 요청률 토큰 버킷의 유효노출률은 수신된 응답 유형에 따라 업데이트됩니다.

Retryable:

이 단계에서는 다음을 기반으로 응답을 재시도할 수 있는지 여부를 결정합니다.

- HTTP상태 코드.
- 서비스에서 반환된 오류 코드입니다.
- 연결 오류로, 서비스로부터 HTTP 응답을 받지 못한 상태에서 수신한 모든 오류로 정의됩니다. SDK

일시적 오류 (HTTP상태 코드 400, 408, 500, 502, 503, 504) 와 제한 오류 (HTTP상태 코드 400, 403, 429, 502, 503, 509) 는 모두 잠재적으로 재시도될 수 있습니다. SDK재시도 동작은 서비스의 오류 코드 또는 기타 데이터와 조합하여 결정됩니다.

MAX_ATTEMPTS:

기본 최대 시도 횟수는 설정으로 재정의되지 않는 한 `retry_mode` 설정에 따라 설정됩니다.

`max_attempts`

HasRetryQuota

이 단계에서는 재시도 토큰 버킷에서 토큰을 획득합니다. 재시도 토큰 버킷이 비어 있는 경우 요청은 재시도되지 않습니다.

ExponentialBackoff

재시도할 수 있는 오류의 경우 잘린 지수 백오프를 사용하여 재시도 지연을 계산합니다. 지터가 있는 잘린 이진 지수 백오프를 SDKs 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: $0 \leq b \leq 1$

$r = 2$

`MAX_BACKOFF = 20 seconds` SDKs 대부분 그렇죠. 확인을 위해 특정 SDK 가이드 또는 소스 코드를 참조하십시오.

다음과 호환됩니다. AWS SDKs

다음은 이 항목에 설명된 기능 및 설정을 SDKs 지원합니다. 모든 일부 예외가 기록됩니다. 모든 JVM 시스템 속성 설정은 에서 지원됩니다. AWS SDK for Java 그리고 AWS SDK for Kotlin 오직.

SDK	지 참고 또는 추가 정보 운
AWS CLI v2	예
SDKC++의 경우	예
SDK고 V2 (1.x) 용	예

SDK	지 운	참고 또는 추가 정보
SDKGo 1.x (V1) 용	아 니 요	
SDK자바 2.x의 경우	예	
SDK자바 1.x의 경우	예	JVM시스템 속성: <code>com.amazonaws.sdk.maxAttempts</code> 대신 사용, <code>com.amazonaws.sdk.retryMode</code> 대신 <code>aws.retryMode</code> 사용 <code>aws.maxAttempts</code> .
SDK JavaScript 3.x용	예	
SDK2.x의 경우 JavaScript	아 니 요	최대 재시도 횟수, 지터가 있는 지수 백오프, 재시도 백오프를 위한 사용자 지정 방법 옵션을 지원합니다.
SDK코틀린의 경우	예	
SDK왜냐하면. NET3.x	예	
SDK3.x의 경우 PHP	예	
SDK파이썬용 (보토3)	예	
SDK루비 3.x의 경우	예	
SDK러스트용	예	
SDK스위프트용	예	
를 위한 도구 PowerShell	예	

요청 압축

Note

설정 페이지의 레이아웃을 이해하거나 호환성을 해석하는 데 도움이 필요합니다. AWS SDKs 다음 표는 을 참조하십시오 [설정 페이지](#).

AWS SDKs 그리고 도구는 요청을 전송할 때 페이로드를 자동으로 압축할 수 있습니다. AWS 서비스 압축된 페이로드 수신을 지원합니다. 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. 압축의 경우 SDK 또는 도구는 서비스와 에서 모두 지원하는 인코딩 알고리즘을 선택합니다. SDK 그러나 현재 가능한 인코딩 목록은 gzip으로만 구성되어 있지만 향후 확장될 수 있습니다.

요청 압축은 애플리케이션에서 [Amazon](#)을 사용하는 경우 특히 유용할 수 CloudWatch 있습니다. CloudWatch 로그, 지표 및 이벤트의 형태로 모니터링 및 운영 데이터를 수집하는 모니터링 및 관찰 서비스입니다. 압축을 지원하는 서비스 작업의 한 예로 CloudWatch 의 [PutMetricDataAPI](#) 메서드를 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: 자바/코틀린 전용

SDK 또는 도구가 요청을 보내기 전에 페이로드를 압축할지 여부를 설정합니다.

기본값: false

유효값:

- **true** - 요청 압축을 해제합니다.
- **false** - 가능하면 요청 압축을 사용합니다.

request_min_compression_size_bytes- 공유 AWS config파일 설정, **AWS_REQUEST_MIN_COMPRESSION_SIZE_BYTES** - 환경 변수,
aws.requestMinCompressionSizeBytes- JVM 시스템 속성: 자바/코틀린 전용

또는 도구가 압축해야 하는 요청 본문의 최소 크기 (바이트) 를 SDK 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.

와의 호환성 AWS SDKs

다음은 이 항목에 설명된 기능 및 설정을 SDKs 지원합니다. 모든 일부 예외가 기록됩니다. 모든 JVM 시스템 속성 설정은 에서 지원됩니다. AWS SDK for Java 그리고 AWS SDK for Kotlin 오직.

SDK	지 운	참고 또는 추가 정보
AWS CLI v2	예	
SDKC++의 경우	예	
SDK고 V2 (1.x) 용	예	
SDKGo 1.x (V1) 용	아 니 요	
SDK자바 2.x의 경우	예	
SDK자바 1.x의 경우	아 니 요	
SDK3.x용 JavaScript	예	

SDK	지 운	참고 또는 추가 정보
SDK2.x의 경우 JavaScript	아 니 요	
SDK코틀린의 경우	예	
SDK왜냐하면. NET3.x	예	
SDK3.x의 경우 PHP	예	
SDK파이썬용 (보토3)	예	
SDK루비 3.x의 경우	예	
SDK러스트용	예	
SDK스위프트용	아 니 요	
를 위한 도구 PowerShell	예	

서비스별 엔드포인트

서비스별 엔드포인트 구성은 API 요청에 대해 선택한 엔드포인트를 사용하고 해당 선택이 지속되도록 하는 옵션을 제공합니다. 이러한 설정은 로컬 엔드포인트, VPC 엔드포인트 및 타사 로컬을 지원할 수 있는 유연성을 제공합니다. AWS 개발 환경, 테스트 환경과 프로덕션 환경에 서로 다른 엔드포인트를 사용할 수 있습니다. 개별 엔드포인트를 URL 지정할 수 있습니다. AWS 서비스.

다음을 사용하여 이 기능을 구성하십시오.

endpoint_url- 공유 AWS config파일 설정, **AWS_ENDPOINT_URL** - 환경 변수,
aws.endpointUrl- JVM 시스템 속성: 자바/코틀린 전용

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 섹션 내에서도 이 설정을 사용할 수 있습니다. `services` AWS config 특정 서비스에 대한 사용자 지정 엔드포인트를 설정하기 위한 파일입니다. `services` 섹션에서 사용할 모든 서비스 식별자 키 목록은 [서비스별 엔드포인트 식별자](#) 섹션을 참조하십시오.

기본값: none

유효한 값: 엔드포인트의 스키마와 호스트를 URL 포함한 A.에는 하나 이상의 경로 세그먼트가 포함된 경로 구성 요소가 선택적으로 포함될 URL 수 있습니다.

AWS_ENDPOINT_URL_<SERVICE> - 환경 변수, **aws.endpointUrl<ServiceName>**- JVM 시스템 속성: 자바/코틀린만 해당

AWS_ENDPOINT_URL_<SERVICE>, 어디에 <SERVICE> AWS 서비스 식별자는 특정 서비스에 대한 사용자 지정 엔드포인트를 설정합니다. 모든 서비스별 환경 변수 목록은 [서비스별 엔드포인트 식별자](#)를 참조하십시오.

이 서비스별 엔드포인트는 **AWS_ENDPOINT_URL**에 설정된 모든 글로벌 엔드포인트보다 우선합니다.

기본값: none

유효한 값: 엔드포인트의 스키마와 호스트를 URL 포함한 A.에는 하나 이상의 경로 세그먼트가 포함된 경로 구성 요소가 선택적으로 포함될 URL 수 있습니다.

ignore_configured_endpoint_urls- 공유 AWS config 파일 설정, **AWS_IGNORE_CONFIGURED_ENDPOINT_URLS** - 환경 변수, **aws.ignoreConfiguredEndpointUrls**- JVM 시스템 속성: 자바/코틀린 전용

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, 다음과 같은 `--endpoint-url` 명령줄 파라미터를 포함합니다. AWS CLI 클라이언트 URL 생성자에 엔드포인트를 전달하거나 명령어는 항상 유효합니다.

기본값: false

유효한 값:

- **true**— SDK or 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 엔드포인트로 URL 라우팅하려면 `AWS_ENDPOINT_URL` 글로벌 환경 변수를 설정합니다.

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

특정 요청에 대한 요청을 라우팅하려면 AWS 서비스 사용자 지정 엔드포인트로 URL `AWS_ENDPOINT_URL_<SERVICE>` 환경 변수를 사용하십시오. Amazon DynamoDB a에는 `serviceId` a가 [DynamoDB](#) 있습니다. 이 서비스의 엔드포인트 URL 환경 변수는 `입니`다 `AWS_ENDPOINT_URL_DYNAMODB`. 이 엔드포인트는 이 서비스에 대해 `AWS_ENDPOINT_URL`에 설정된 글로벌 엔드포인트보다 우선합니다.

```
export AWS_ENDPOINT_URL_DYNAMODB=http://localhost:5678
```

또 다른 예로, AWS Elastic Beanstalk a가 `serviceId` 있습니다 [Elastic Beanstalk](#). 그 AWS 서비스 식별자는 API 모델을 기반으로 모든 공백을 `serviceId` 밑줄로 바꾸고 모든 문자를 대문자로 표시합니다. 이 서비스에 대한 엔드포인트를 설정하려면 해당 환경 변수는 `AWS_ENDPOINT_URL_ELASTIC_BEANSTALK`입니다. 모든 서비스별 환경 변수 목록은 [서비스별 엔드포인트 식별자](#)을 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_url`이 사용됩니다.

- `profile` 내에서 직접 지정된 `endpoint_url`은 해당 엔드포인트를 글로벌 엔드포인트로 만듭니다.
- `services` 섹션 내의 서비스 식별자 키 아래에 중첩된 `endpoint_url`은 해당 엔드포인트가 해당 서비스에 대한 요청에만 적용되도록 만듭니다. 공유 `config` 파일에서 `services` 섹션을 정의하는 방법에 대한 자세한 내용은 [구성 파일 형식](#)를 참조하십시오.

다음 예제는 `services` 정의를 사용하여 Amazon URL S3에 사용할 서비스별 엔드포인트와 다른 모든 서비스에 사용할 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 예제는 Amazon S3의 URLs 서비스별 엔드포인트를 설정하는 방법을 보여줍니다. AWS Elastic Beanstalk 동일한 프로 필에서. AWS Elastic Beanstalk 가 serviceId 있습니다 [Elastic Beanstalk](#). 그 AWS 서비스 식별자는 API 모델을 기반으로 모든 공백을 serviceId 밑줄로 바꾸고 모든 문자를 소문자로 바꿉니다. 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 식별자](#)를 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

역할 수임 기능을 위한 `source_profile` IAM 매개 변수를 통해 구성된 역할 기반 자격 증명에 프로필에 있는 경우는 지정된 프로필에 대한 서비스 SDK 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
```

Bprofile을 사용하고 코드를 EC2 Amazon에 호출하면 엔드포인트는 다음과 같이 `https://profile-b-ec2-endpoint.aws` 확인됩니다. 코드에서 다른 서비스에 요청을 하는 경우 엔드포인트 확인은 사용자 지정 로직을 따르지 않습니다. 엔드포인트는 프로파일 A에 정의된 글로벌 엔드포인트로 확인되지 않습니다. 글로벌 엔드포인트가 프로파일 B에 적용되려면 프로파일 B 내에서 직접 `endpoint_url`을 설정해야 합니다. `source_profile` 설정에 대한 자세한 내용은 [역할 보안 인증 제 공자 수임](#) 단원을 참조하십시오.

설정의 우선 순위

이 기능의 설정은 동시에 사용할 수 있지만 서비스당 하나의 값만 우선합니다. 특정 사용자에게 API 걸려온 통화의 경우 AWS 서비스다음 순서를 사용하여 값을 선택합니다.

1. 코드나 서비스 클라이언트 자체에 설정된 모든 명시적 설정은 다른 모든 설정보다 우선합니다.
 - 의 경우 AWS CLI이 값은 `--endpoint-url` 명령줄 매개 변수에서 제공하는 값입니다. 의 SDK 경우 명시적 할당은 인스턴스화할 때 설정하는 매개 변수의 형태를 취할 수 있습니다. AWS 서비스 클라이언트 또는 구성 객체.
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. 각 URL 엔드포인트의 모든 기본 엔드포인트 AWS 서비스 마지막에 사용됩니다.

와의 호환성 AWS SDKs

다음은 이 항목에 설명된 기능 및 설정을 SDKs 지원합니다. 모든 일부 예외가 기록됩니다. 모든 JVM 시스템 속성 설정은 에서 지원됩니다. AWS SDK for Java 그리고 AWS SDK for Kotlin 오직.

SDK	지 운	참고 또는 추가 정보
AWS CLI v2	예	
SDKC++의 경우	아 니 요	
SDK고 V2 (1.x) 용	예	
SDK고 1.x (V1) 용	아 니 요	
SDK자바 2.x의 경우	예	
SDK자바 1.x의 경우	아 니 요	
SDK3.x용 JavaScript	예	
SDK2.x의 경우 JavaScript	아 니 요	
SDK코틀린의 경우	예	
SDK왜냐하면. NET3.x	예	

SDK	지 운	참고 또는 추가 정보
SDK3.x의 경우 PHP	예	
SDK파이썬용 (보토3)	예	
SDK루비 3.x의 경우	예	
SDK러스트용	아 니 요	
SDK스위프트용	아 니 요	
를 위한 도구 PowerShell	예	

서비스별 엔드포인트 식별자

다음 표의 식별자를 사용하는 방법 및 위치에 대한 자세한 내용은 [서비스별 엔드포인트](#) 섹션을 참조하십시오.

serviceId	공용 서비스별 자격식별자 AWS_ENDPOINT_URL_<SERVICE> 환경 변수
AccessAnalyzer	aws_endpoint_url_accessanalyzer
Account	aws_endpoint_url_account
ACM	aws_endpoint_url_acm
ACM PCA	aws_endpoint_url_acm_pca
Alexa For Business	aws_endpoint_url_alexa_for_business
amp	aws_endpoint_url_amp
Amplify	aws_endpoint_url_amplify
AmplifyBackend	aws_endpoint_url_amplifybackend
AmplifyUIBuilder	aws_endpoint_url_amplifyuibuilder

serviceId	공용 서비스별 자격식별자 AWS_ENDPOINT_URL_<SERVICE> 환경 변수
API Gateway	a) AWS_ENDPOINT_URL_API_GATEWAY a)
ApiGatewayManagementApi	a) AWS_ENDPOINT_URL_APIGATEWAYMANAGEMENTAPI y) n)
ApiGatewayV2	a) AWS_ENDPOINT_URL_APIGATEWAYV2 y)
AppConfig	a) AWS_ENDPOINT_URL_APPCONFIG
AppConfigData	a) AWS_ENDPOINT_URL_APPCONFIGDATA d)
AppFabric	a) AWS_ENDPOINT_URL_APPFABRIC
Appflow	a) AWS_ENDPOINT_URL_APPFLOW
AppIntegrations	a) AWS_ENDPOINT_URL_APPINTEGRATIONS a)

serviceId	공용 서비스별 환경 변수 AWS_ENDPOINT_URL_<SERVICE>
Application Auto Scaling	aws:application-auto-scaling
Application Insights	aws:application-insights
ApplicationCostProfiler	aws:application-cost-profiler
App Mesh	aws:app-mesh
AppRunner	aws:apprunner
AppStream	aws:appstream
AppSync	aws:appsync

serviceId	공용 서비스별 엔드포인트 URL 환경 변수
ARC Zonal Shift	a: AWS_ENDPOINT_URL_ARC_ZONAL_SHIFT
Artifact	a: AWS_ENDPOINT_URL_ARTIFACT
Athena	a: AWS_ENDPOINT_URL_ATHENA
AuditManager	a: AWS_ENDPOINT_URL_AUDITMANAGER
Auto Scaling	a: AWS_ENDPOINT_URL_AUTO_SCALING
Auto Scaling Plans	a: AWS_ENDPOINT_URL_AUTO_SCALING_PLANS
b2bi	b: AWS_ENDPOINT_URL_B2BI
Backup	b: AWS_ENDPOINT_URL_BACKUP
Backup Gateway	b: AWS_ENDPOINT_URL_BACKUP_GATEWAY

serviceId	공 유 용 서 비 스 스 별 자 카 시 C 파 일	환경 변수
BackupStorage	b:	AWS_ENDPOINT_URL_BACKUPSTORAGE
Batch	b:	AWS_ENDPOINT_URL_BATCH
BCM Data Exports	b: e:	AWS_ENDPOINT_URL_BCM_DATA_EXPORTS
Bedrock	b:	AWS_ENDPOINT_URL_BEDROCK
Bedrock Agent	b: g:	AWS_ENDPOINT_URL_BEDROCK_AGENT
Bedrock Agent Runtime	b: g: ir	AWS_ENDPOINT_URL_BEDROCK_AGENT_RUNTIME
Bedrock Runtime	b: ur	AWS_ENDPOINT_URL_BEDROCK_RUNTIME
billingconductor	b: n:	AWS_ENDPOINT_URL_BILLINGCONDUCTOR

serviceId	공용 서비스별 환경 변수 AWS_ENDPOINT_URL_<SERVICE>
Braket	b: AWS_ENDPOINT_URL_BRAKET
Budgets	b: AWS_ENDPOINT_URL_BUDGETS
Cost Explorer	c: AWS_ENDPOINT_URL_COST_EXPLORER
chatbot	c: AWS_ENDPOINT_URL_CHATBOT
Chime	c: AWS_ENDPOINT_URL_CHIME
Chime SDK Identity	c: AWS_ENDPOINT_URL_CHIME_SDK_IDENTITY
Chime SDK Media Pipelines	c: AWS_ENDPOINT_URL_CHIME_SDK_MEDIA_PIPELINES
Chime SDK Meetings	c: AWS_ENDPOINT_URL_CHIME_SDK_MEETINGS

serviceId	공용 서비스별 엔드포인트 URL 환경 변수
Chime SDK Messaging	c: AWS_ENDPOINT_URL_CHIME_SDK_MESSAGING
Chime SDK Voice	c: AWS_ENDPOINT_URL_CHIME_SDK_VOICE
CleanRooms	c: AWS_ENDPOINT_URL_CLEANROOMS
CleanRoomsML	c: AWS_ENDPOINT_URL_CLEANROOMSML
Cloud9	c: AWS_ENDPOINT_URL_CLOUD9
CloudControl	c: AWS_ENDPOINT_URL_CLOUDCONTROL
CloudDirectory	c: AWS_ENDPOINT_URL_CLOUDDIRECTORY
CloudFormation	c: AWS_ENDPOINT_URL_CLOUDFORMATION

serviceId	공용 서비스별 엔드포인트 환경 변수
CloudFront	c: AWS_ENDPOINT_URL_CLOUDFRONT
CloudFront KeyValueStore	c: AWS_ENDPOINT_URL_CLOUDFRONT_KEYVALUESTORE
CloudHSM	c: AWS_ENDPOINT_URL_CLOUDHSM
CloudHSM V2	c: AWS_ENDPOINT_URL_CLOUDHSM_V2
CloudSearch	c: AWS_ENDPOINT_URL_CLOUDSEARCH
CloudSearch Domain	c: AWS_ENDPOINT_URL_CLOUDSEARCH_DOMAIN
CloudTrail	c: AWS_ENDPOINT_URL_CLOUDTRAIL

serviceId	공용 서비스별 자카파일 AWS_ENDPOINT_URL_<SERVICE> 환경 변수
CloudTrail Data	c: AWS_ENDPOINT_URL_CLOUDTRAIL_DATA l_
CloudWatch	c: AWS_ENDPOINT_URL_CLOUDWATCH h
codeartifact	c: AWS_ENDPOINT_URL_CODEARTIFACT a:
CodeBuild	c: AWS_ENDPOINT_URL_CODEBUILD
CodeCatalyst	c: AWS_ENDPOINT_URL_CODECATALYST y:
CodeCommit	c: AWS_ENDPOINT_URL_CODECOMMIT t
CodeDeploy	c: AWS_ENDPOINT_URL_CODEDEPLOY y
CodeGuru Reviewer	c: AWS_ENDPOINT_URL_CODEGURU_REVIEWER r:

serviceId	공용 서비스별 환경 변수
CodeGuru Security	<code>AWS_ENDPOINT_URL_CODEGURU_SECURITY</code>
CodeGuruProfiler	<code>AWS_ENDPOINT_URL_CODEGURUPROFILER</code>
CodePipeline	<code>AWS_ENDPOINT_URL_CODEPIPELINE</code>
CodeStar	<code>AWS_ENDPOINT_URL_CODESTAR</code>
CodeStar connections	<code>AWS_ENDPOINT_URL_CODESTAR_CONNECTIONS</code>
codestar notifications	<code>AWS_ENDPOINT_URL_CODESTAR_NOTIFICATIONS</code>
Cognito Identity	<code>AWS_ENDPOINT_URL_COGNITO_IDENTITY</code>

serviceId	공용 서비스별 환경 변수 AWS_ENDPOINT_URL_<SERVICE>
Cognito Identity Provider	AWS_ENDPOINT_URL_COGNITO_IDENTITY_PROVIDER
Cognito Sync	AWS_ENDPOINT_URL_COGNITO_SYNC
Comprehend	AWS_ENDPOINT_URL_COMPREHEND
ComprehendMedical	AWS_ENDPOINT_URL_COMPREHENDMEDICAL
Compute Optimizer	AWS_ENDPOINT_URL_COMPUTE_OPTIMIZER
Config Service	AWS_ENDPOINT_URL_CONFIG_SERVICE
Connect	AWS_ENDPOINT_URL_CONNECT

serviceId	공용 서비스별 엔드포인트 환경 변수
Connect Contact Lens	AWS_ENDPOINT_URL_CONNECT_CONTACT_LENS
ConnectCampaigns	AWS_ENDPOINT_URL_CONNECTCAMPAIGNS
ConnectCases	AWS_ENDPOINT_URL_CONNECTCASES
ConnectParticipant	AWS_ENDPOINT_URL_CONNECTPARTICIPANT
ControlTower	AWS_ENDPOINT_URL_CONTROLTOWER
Cost Optimization Hub	AWS_ENDPOINT_URL_COST_OPTIMIZATION_HUB

serviceId	공용 서비스별 환경 변수 AWS_ENDPOINT_URL_<SERVICE>
Cost and Usage Report Service	c: AWS_ENDPOINT_URL_COST_AND_USAGE_REPO u: RT_SERVICE o: c:
Customer Profiles	c: AWS_ENDPOINT_URL_CUSTOMER_PROFILES p:
DataBrew	d: AWS_ENDPOINT_URL_DATABREW
DataExchange	d: AWS_ENDPOINT_URL_DATAEXCHANGE n:
Data Pipeline	d: AWS_ENDPOINT_URL_DATA_PIPELINE l:
DataSync	d: AWS_ENDPOINT_URL_DATASYNC
DataZone	d: AWS_ENDPOINT_URL_DATAZONE
DAX	d: AWS_ENDPOINT_URL_DAX

serviceId	공용 서비스별 환경 변수
Detective	d: AWS_ENDPOINT_URL_DETECTIVE
Device Farm	d: AWS_ENDPOINT_URL_DEVICE_FARM
DevOps Guru	d: AWS_ENDPOINT_URL_DEVOPS_GURU
Direct Connect	d: AWS_ENDPOINT_URL_DIRECT_CONNECT
Application Discovery Service	a: AWS_ENDPOINT_URL_APPLICATION_DISCOVERY_SERVICE
DLM	d: AWS_ENDPOINT_URL_DLM
Database Migration Service	d: AWS_ENDPOINT_URL_DATABASE_MIGRATION_SERVICE

serviceId	공 유 용 서 버 스 스 별 자 크 리 프 파 일	환경 변수
DocDB	d:	AWS_ENDPOINT_URL_DOCDB
DocDB Elastic	d:	AWS_ENDPOINT_URL_DOCDB_ELASTIC
drs	d:	AWS_ENDPOINT_URL_DRS
Directory Service	d:	AWS_ENDPOINT_URL_DIRECTORY_SERVICE
DynamoDB	d:	AWS_ENDPOINT_URL_DYNAMODB
DynamoDB Streams	d:	AWS_ENDPOINT_URL_DYNAMODB_STREAMS
EBS	e:	AWS_ENDPOINT_URL_EBS
EC2	e:	AWS_ENDPOINT_URL_EC2
EC2 Instance Connect	e:	AWS_ENDPOINT_URL_EC2_INSTANCE_CONNECT
ECR	e:	AWS_ENDPOINT_URL_ECR

serviceId	공용 서비스별 엔드포인트 URL 환경 변수
ECR PUBLIC	e: AWS_ENDPOINT_URL_ECR_PUBLIC
ECS	e: AWS_ENDPOINT_URL_ECS
EFS	e: AWS_ENDPOINT_URL_EFS
EKS	e: AWS_ENDPOINT_URL_EKS
EKS Auth	e: AWS_ENDPOINT_URL_EKS_AUTH
Elastic Inference	e: AWS_ENDPOINT_URL_ELASTIC_INFERENCE
ElastiCache	e: AWS_ENDPOINT_URL_ELASTICACHE
Elastic Beanstalk	e: AWS_ENDPOINT_URL_ELASTIC_BEANSTALK
Elastic Transcoder	e: AWS_ENDPOINT_URL_ELASTIC_TRANSCODER

serviceId	공용 서비스별 엔드포인트 URL 환경 변수
Elastic Load Balancing	e: AWS_ENDPOINT_URL_ELASTIC_LOAD_BALANCING o: c:
Elastic Load Balancing v2	e: AWS_ENDPOINT_URL_ELASTIC_LOAD_BALANCING_V2 o: c:
EMR	er AWS_ENDPOINT_URL_EMR
EMR containers	er AWS_ENDPOINT_URL_EMR_CONTAINERS i:
EMR Serverless	er AWS_ENDPOINT_URL_EMR_SERVERLESS r:
EntityResolution	er AWS_ENDPOINT_URL_ENTITYRESOLUTION o:
Elasticsearch Service	e: AWS_ENDPOINT_URL_ELASTICSEARCH_SERVICE a: i:

serviceId	공 유 용 서 비 스 스 별 자 크 리 드 파 일	환경 변수
EventBridge	e	AWS_ENDPOINT_URL_EVENTBRIDGE
Evidently	e	AWS_ENDPOINT_URL_EVIDENTLY
finspace	f	AWS_ENDPOINT_URL_FINSPLACE
finspace data	f	AWS_ENDPOINT_URL_FINSPLACE_DATA
Firehose	f	AWS_ENDPOINT_URL_FIREHOSE
fis	f	AWS_ENDPOINT_URL_FIS
FMS	f	AWS_ENDPOINT_URL_FMS
forecast	f	AWS_ENDPOINT_URL_FORECAST
forecastquery	f	AWS_ENDPOINT_URL_FORECASTQUERY
FraudDetector	f	AWS_ENDPOINT_URL_FRAUDETECTOR

serviceId	공용 서비스별 환경 변수 AWS_ENDPOINT_URL_<SERVICE>
FreeTier	f: AWS_ENDPOINT_URL_FREETIER
FSx	f: AWS_ENDPOINT_URL_FSX
GameLift	g: AWS_ENDPOINT_URL_GAMELIFT
Glacier	g: AWS_ENDPOINT_URL_GLACIER
Global Accelerator	g: AWS_ENDPOINT_URL_GLOBAL_ACCELERATOR c:
Glue	g: AWS_ENDPOINT_URL_GLUE
grafana	g: AWS_ENDPOINT_URL_GRAFANA
Greengrass	g: AWS_ENDPOINT_URL_GREENGRASS s
GreengrassV2	g: AWS_ENDPOINT_URL_GREENGRASSV2 s\
GroundStation	g: AWS_ENDPOINT_URL_GROUNDSTATION t:

serviceId	공용 서비스별 엔드포인트	환경 변수
GuardDuty	gd	AWS_ENDPOINT_URL_GUARDDUTY
Health	hc	AWS_ENDPOINT_URL_HEALTH
HealthLake	hl	AWS_ENDPOINT_URL_HEALTHLAKE
Honeycode	hc	AWS_ENDPOINT_URL_HONEYCODE
IAM	ia	AWS_ENDPOINT_URL_IAM
identitystore	is	AWS_ENDPOINT_URL_IDENTITYSTORE
imagebuilder	ib	AWS_ENDPOINT_URL_IMAGEBUILDER
ImportExport	ie	AWS_ENDPOINT_URL_IMPORTEXPORT

serviceId	공용 서비스별 엔드포인트 URL 환경 변수
Inspector	i: AWS_ENDPOINT_URL_INSPECTOR
Inspector Scan	i: AWS_ENDPOINT_URL_INSPECTOR_SCAN _s
Inspector2	i: AWS_ENDPOINT_URL_INSPECTOR2 2
InternetMonitor	i: AWS_ENDPOINT_URL_INTERNETMONITOR or
IoT	i: AWS_ENDPOINT_URL_IOT
IoT Data Plane	i: AWS_ENDPOINT_URL_IOT_DATA_PLANE p:
IoT Jobs Data Plane	i: AWS_ENDPOINT_URL_IOT_JOBS_DATA_PLANE d: e

serviceId	공용 서비스별 환경 변수
IoT 1Click Devices Service	i{ AWS_ENDPOINT_URL_IOT_1CLICK_DEVICES_ k_ SERVICE _!
IoT 1Click Projects	i{ AWS_ENDPOINT_URL_IOT_1CLICK_PROJECTS k_ s
IoTAnalytics	i{ AWS_ENDPOINT_URL_IOTANALYTICS i{
IotDeviceAdvisor	i{ AWS_ENDPOINT_URL_IOTDEVICEADVISOR a{
IoT Events	i{ AWS_ENDPOINT_URL_IOT_EVENTS s
IoT Events Data	i{ AWS_ENDPOINT_URL_IOT_EVENTS_DATA s_
IoTFleetHub	i{ AWS_ENDPOINT_URL_IOTFLEETHUB ul

serviceId	공 유 용 서 비 스 스 별 자 크 리 프 파 일	환경 변수
IoTFleetWise	i o	AWS_ENDPOINT_URL_IOTFLEETWISE
IoTSecureTunneling	i o	AWS_ENDPOINT_URL_IOTSECURETUNNELING
IoTSiteWise	i o	AWS_ENDPOINT_URL_IOTSITWISE
IoTThingsGraph	i o	AWS_ENDPOINT_URL_IOTTHINGSGRAPH
IoTTwinMaker	i o	AWS_ENDPOINT_URL_IOTTWINMAKER
IoT Wireless	i o	AWS_ENDPOINT_URL_IOT_WIRELESS
ivs	i v	AWS_ENDPOINT_URL_IVS
IVS RealTime	i v	AWS_ENDPOINT_URL_IVS_REALTIME

serviceId	공용 서비스별 엔드포인트 URL 환경 변수
ivschat	i: AWS_ENDPOINT_URL_IVSCHAT
Kafka	k: AWS_ENDPOINT_URL_KAFKA
KafkaConnect	k: AWS_ENDPOINT_URL_KAFKACONNECT e:
kendra	k: AWS_ENDPOINT_URL_KENDRA
Kendra Ranking	k: AWS_ENDPOINT_URL_KENDRA_RANKING n:
Keyspaces	k: AWS_ENDPOINT_URL_KEYSPACES
Kinesis	k: AWS_ENDPOINT_URL_KINESIS
Kinesis Video Archived Media	k: AWS_ENDPOINT_URL_KINESIS_VIDEO_ARCHI i: VED_MEDIA i: a

serviceId	공용 서비스별 환경 변수 AWS_ENDPOINT_URL_<SERVICE>
Kinesis Video Media	k: AWS_ENDPOINT_URL_KINESIS_VIDEO_MEDIA i: a
Kinesis Video Signaling	k: AWS_ENDPOINT_URL_KINESIS_VIDEO_SIGNALING i: a:
Kinesis Video WebRTC Storage	k: AWS_ENDPOINT_URL_KINESIS_VIDEO_WEBRTC_STORAGE i: C_STORAGE
Kinesis Analytics	k: AWS_ENDPOINT_URL_KINESIS_ANALYTICS n:
Kinesis Analytics V2	k: AWS_ENDPOINT_URL_KINESIS_ANALYTICS_V2 n: v:
Kinesis Video	k: AWS_ENDPOINT_URL_KINESIS_VIDEO i:

serviceId	공용 서비스별 환경 변수 AWS_ENDPOINT_URL_<SERVICE>
KMS	kr AWS_ENDPOINT_URL_KMS
LakeFormation	l AWS_ENDPOINT_URL_LAKEFORMATION t:
Lambda	l AWS_ENDPOINT_URL_LAMBDA
Launch Wizard	l AWS_ENDPOINT_URL_LAUNCH_WIZARD z:
Lex Model Building Service	l AWS_ENDPOINT_URL_LEX_MODEL_BUILDING_ _ SERVICE _:
Lex Runtime Service	l AWS_ENDPOINT_URL_LEX_RUNTIME_SERVICE m e
Lex Models V2	l AWS_ENDPOINT_URL_LEX_MODELS_V2 s.
Lex Runtime V2	l AWS_ENDPOINT_URL_LEX_RUNTIME_V2 m

serviceId	공용 서비스별 환경 변수 AWS_ENDPOINT_URL_<SERVICE>
License Manager	AWS_ENDPOINT_URL_LICENSE_MANAGER
License Manager Linux Subscriptions	AWS_ENDPOINT_URL_LICENSE_MANAGER_LINUX_SUBSCRIPTIONS
License Manager User Subscriptions	AWS_ENDPOINT_URL_LICENSE_MANAGER_USER_SUBSCRIPTIONS
Lightsail	AWS_ENDPOINT_URL_LIGHTSAIL
Location	AWS_ENDPOINT_URL_LOCATION
CloudWatch Logs	AWS_ENDPOINT_URL_CLOUDWATCH_LOGS
CloudWatch Logs	AWS_ENDPOINT_URL_CLOUDWATCH_LOGS

serviceId	공용 서비스별 환경 변수
LookoutEquipment	LookoutEquipment: AWS_ENDPOINT_URL_LOOKOUTEQUIPMENT
LookoutMetrics	LookoutMetrics: AWS_ENDPOINT_URL_LOOKOUTMETRICS
LookoutVision	LookoutVision: AWS_ENDPOINT_URL_LOOKOUTVISION
m2	m2: AWS_ENDPOINT_URL_M2
Machine Learning	Machine Learning: AWS_ENDPOINT_URL_MACHINE_LEARNING
Macie2	Macie2: AWS_ENDPOINT_URL_MACIE2
ManagedBlockchain	ManagedBlockchain: AWS_ENDPOINT_URL_MANAGEDBLOCKCHAIN
ManagedBlockchain Query	ManagedBlockchain Query: AWS_ENDPOINT_URL_MANAGEDBLOCKCHAIN_QUERY

serviceId	공용 서비스별 엔드포인트 환경 변수
Marketplace Agreement	m: AWS_ENDPOINT_URL_MARKETPLACE_AGREEMENT c: e
Marketplace Catalog	m: AWS_ENDPOINT_URL_MARKETPLACE_CATALOG c: g
Marketplace Deployment	m: AWS_ENDPOINT_URL_MARKETPLACE_DEPLOYMENT c: m
Marketplace Entitlement Service	m: AWS_ENDPOINT_URL_MARKETPLACE_ENTITLEMENT_SERVICE c: v:
Marketplace Commerce Analytics	m: AWS_ENDPOINT_URL_MARKETPLACE_COMMERCE_ANALYTICS c: i

serviceId	공용 서비스별 자격식별자 AWS_ENDPOINT_URL_<SERVICE> 환경 변수
MediaConnect	m AWS_ENDPOINT_URL_MEDIACONNECT e
MediaConvert	m AWS_ENDPOINT_URL_MEDIACONVERT e:
MediaLive	m AWS_ENDPOINT_URL_MEDIALIVE
MediaPackage	m AWS_ENDPOINT_URL_MEDIAPACKAGE a
MediaPackage Vod	m AWS_ENDPOINT_URL_MEDIAPACKAGE_VOD a
MediaPackageV2	m AWS_ENDPOINT_URL_MEDIAPACKAGEV2 a
MediaStore	m AWS_ENDPOINT_URL_MEDIASTORE e
MediaStore Data	m AWS_ENDPOINT_URL_MEDIASTORE_DATA e.

serviceId	공용 서비스별 엔드포인트 URL 환경 변수
MediaTailor	m: AWS_ENDPOINT_URL_MEDIATAILOR o:
Medical Imaging	m: AWS_ENDPOINT_URL_MEDICAL_IMAGING m:
MemoryDB	m: AWS_ENDPOINT_URL_MEMORYDB
Marketplace Metering	m: AWS_ENDPOINT_URL_MARKETPLACE_METERING c: n:
Migration Hub	m: AWS_ENDPOINT_URL_MIGRATION_HUB _l
mgn	m: AWS_ENDPOINT_URL_MGN
Migration Hub Refactor Spaces	m: AWS_ENDPOINT_URL_MIGRATION_HUB_REFACTOR_SPACES c: e:

serviceId	공용 서비스별 엔드포인트 URL 환경 변수
MigrationHub Config	m: AWS_ENDPOINT_URL_MIGRATIONHUB_CONFIG
MigrationHubOrchestrator	m: AWS_ENDPOINT_URL_MIGRATIONHUBORCHESTRATOR
MigrationHubStrategy	m: AWS_ENDPOINT_URL_MIGRATIONHUBSTRATEGY
Mobile	m: AWS_ENDPOINT_URL_MOBILE
mq	m: AWS_ENDPOINT_URL_MQ
MTurk	m: AWS_ENDPOINT_URL_MTURK
MWA	m: AWS_ENDPOINT_URL_MWA
Neptune	n: AWS_ENDPOINT_URL_NEPTUNE

serviceId	공용 서비스별 엔드포인트 환경 변수
Neptune Graph	n: AWS_ENDPOINT_URL_NEPTUNE_GRAPH i:
neptunedata	n: AWS_ENDPOINT_URL_NEPTUNEDATA t:
Network Firewall	n: AWS_ENDPOINT_URL_NETWORK_FIREWALL i:
NetworkManager	n: AWS_ENDPOINT_URL_NETWORKMANAGER n:
NetworkMonitor	n: AWS_ENDPOINT_URL_NETWORKMONITOR n:
nimble	n: AWS_ENDPOINT_URL_NIMBLE
OAM	o: AWS_ENDPOINT_URL_OAM
Omics	or: AWS_ENDPOINT_URL_OMICS
OpenSearch	o: AWS_ENDPOINT_URL_OPENSEARCH h

serviceId	공용 서비스별 환경 변수 AWS_ENDPOINT_URL_<SERVICE>
OpenSearchServerless	o: AWS_ENDPOINT_URL_OPENSEARCHSERVERLESS
OpsWorks	o: AWS_ENDPOINT_URL_OPSWORKS
OpsWorksCM	o: AWS_ENDPOINT_URL_OPSWORKSCM
Organizations	o: AWS_ENDPOINT_URL_ORGANIZATIONS
OSIS	o: AWS_ENDPOINT_URL_OSIS
Outposts	o: AWS_ENDPOINT_URL_OUTPOSTS
p8data	p: AWS_ENDPOINT_URL_P8DATA
p8data	p: AWS_ENDPOINT_URL_P8DATA
Panorama	p: AWS_ENDPOINT_URL_PANORAMA

serviceId	공용 서비스별 엔드포인트 URL 환경 변수
Payment Cryptography	p: AWS_ENDPOINT_URL_PAYMENT_CRYPTOGRAPHY
Payment Cryptography Data	p: AWS_ENDPOINT_URL_PAYMENT_CRYPTOGRAPHY_DATA
Pca Connector Ad	p: AWS_ENDPOINT_URL_PCA_CONNECTOR_AD
Personalize	p: AWS_ENDPOINT_URL_PERSONALIZE
Personalize Events	p: AWS_ENDPOINT_URL_PERSONALIZE_EVENTS
Personalize Runtime	p: AWS_ENDPOINT_URL_PERSONALIZE_RUNTIME
PI	p: AWS_ENDPOINT_URL_PI

serviceId	공용 서비스별 엔드포인트 URL 환경 변수
Pinpoint	p: AWS_ENDPOINT_URL_PINPOINT
Pinpoint Email	p: AWS_ENDPOINT_URL_PINPOINT_EMAIL
Pinpoint SMS Voice	p: AWS_ENDPOINT_URL_PINPOINT_SMS_VOICE
Pinpoint SMS Voice V2	p: AWS_ENDPOINT_URL_PINPOINT_SMS_VOICE_V2
Pipes	p: AWS_ENDPOINT_URL_PIPES
Polly	p: AWS_ENDPOINT_URL_POLLY
Pricing	p: AWS_ENDPOINT_URL_PRICING
PrivateNetworks	p: AWS_ENDPOINT_URL_PRIVATENETWORKS
Proton	p: AWS_ENDPOINT_URL_PROTON

serviceId	공용 서비스별 자격식별자	AWS_ENDPOINT_URL_<SERVICE> 환경 변수
QBusiness	qbusiness	AWS_ENDPOINT_URL_QBUSINESS
QConnect	qconnect	AWS_ENDPOINT_URL_QCONNECT
QLDB	qldb	AWS_ENDPOINT_URL_QLDB
QLDB Session	qldb-session	AWS_ENDPOINT_URL_QLDB_SESSION
QuickSight	quicksight	AWS_ENDPOINT_URL_QUICKSIGHT
RAM	ram	AWS_ENDPOINT_URL_RAM
rbin	rbin	AWS_ENDPOINT_URL_RBIN
RDS	rds	AWS_ENDPOINT_URL_RDS
RDS Data	rds-data	AWS_ENDPOINT_URL_RDS_DATA
Redshift	redshift	AWS_ENDPOINT_URL_REDSHIFT

serviceId	공용 서비스별 환경 변수 AWS_ENDPOINT_URL_<SERVICE>
Redshift Data	AWS_ENDPOINT_URL_REDSHIFT_DATA
Redshift Serverless	AWS_ENDPOINT_URL_REDSHIFT_SERVERLESS
Rekognition	AWS_ENDPOINT_URL_REKOGNITION
repostspace	AWS_ENDPOINT_URL_REPOSTSPACE
resiliencehub	AWS_ENDPOINT_URL_RESILIENCEHUB
Resource Explorer 2	AWS_ENDPOINT_URL_RESOURCE_EXPLORER_2
Resource Groups	AWS_ENDPOINT_URL_RESOURCE_GROUPS

serviceId	공용 서비스별 환경 변수
Resource Groups Tagging API	<pre> AWS_ENDPOINT_URL_RESOURCE_GROUPS_TAGGING_API </pre>
RoboMaker	<pre> AWS_ENDPOINT_URL_ROBOMAKER </pre>
RolesAnywhere	<pre> AWS_ENDPOINT_URL_ROLESEANYWHERE </pre>
Route 53	<pre> AWS_ENDPOINT_URL_ROUTE_53 </pre>
Route53 Recovery Cluster	<pre> AWS_ENDPOINT_URL_ROUTE53_RECOVERY_CLUSTER </pre>
Route53 Recovery Control Config	<pre> AWS_ENDPOINT_URL_ROUTE53_RECOVERY_CONTROL_CONFIG </pre>

serviceId	공용 서비스별 자격식별자 AWS_ENDPOINT_URL_<SERVICE> 환경 변수
Route53 Recovery Readiness	i: AWS_ENDPOINT_URL_ROUTE53_RECOVERY_READINESS
Route 53 Domains	i: AWS_ENDPOINT_URL_ROUTE_53_DOMAINS
Route53Resolver	i: AWS_ENDPOINT_URL_ROUTE53RESOLVER
RUM	i: AWS_ENDPOINT_URL_RUM
S3	s: AWS_ENDPOINT_URL_S3
S3 Control	s: AWS_ENDPOINT_URL_S3_CONTROL
S3Outposts	s: AWS_ENDPOINT_URL_S3OUTPOSTS
SageMaker	s: AWS_ENDPOINT_URL_SAGEMAKER

serviceId	공용 서비스별 환경 변수 AWS_ENDPOINT_URL_<SERVICE>
SageMaker A2I Runtime	s: AWS_ENDPOINT_URL_SAGEMAKER_A2I_RUNTIME
Sagemaker Edge	s: AWS_ENDPOINT_URL_SAGEMAKER_EDGE
SageMaker FeatureStore Runtime	s: AWS_ENDPOINT_URL_SAGEMAKER_FEATURESTORE_RUNTIME
SageMaker Geospatial	s: AWS_ENDPOINT_URL_SAGEMAKER_GEOSPATIAL
SageMaker Metrics	s: AWS_ENDPOINT_URL_SAGEMAKER_METRICS
SageMaker Runtime	s: AWS_ENDPOINT_URL_SAGEMAKER_RUNTIME

serviceId	공용 서비스별 환경 변수 AWS_ENDPOINT_URL_<SERVICE>
savingsplans	savingsplans: AWS_ENDPOINT_URL_SAVINGSPLANS
Scheduler	scheduler: AWS_ENDPOINT_URL_SCHEDULER
schemas	schemas: AWS_ENDPOINT_URL_SCHEMAS
SimpleDB	sdb: AWS_ENDPOINT_URL_SIMPLEDB
Secrets Manager	secretsmanager: AWS_ENDPOINT_URL_SECRETS_MANAGER
SecurityHub	securityhub: AWS_ENDPOINT_URL_SECURITYHUB
SecurityLake	securitylake: AWS_ENDPOINT_URL_SECURITYLAKE
ServerlessApplicationRepository	serverlessapplicationrepository: AWS_ENDPOINT_URL_SERVERLESSAPPLICATIONREPOSITORY

serviceId	공용 서비스별 자격식별자 파일 AWS_ENDPOINT_URL_<SERVICE> 환경 변수
Service Quotas	s: AWS_ENDPOINT_URL_SERVICE_QUOTAS u:
Service Catalog	s: AWS_ENDPOINT_URL_SERVICE_CATALOG a:
Service Catalog AppRegistry	s: AWS_ENDPOINT_URL_SERVICE_CATALOG_APP a: REGISTRY p:
ServiceDiscovery	s: AWS_ENDPOINT_URL_SERVICEDISCOVERY s:
SES	s: AWS_ENDPOINT_URL_SES
SESV2	s: AWS_ENDPOINT_URL_SESV2
Shield	s: AWS_ENDPOINT_URL_SHIELD
signer	s: AWS_ENDPOINT_URL_SIGNER

serviceId	공 유 용 서 버 스 스 발 자 크 리 프 파 일	환경 변수
SimSpaceWeaver	s:	AWS_ENDPOINT_URL_SIMSPACEWEAVER
SMS	sr	AWS_ENDPOINT_URL_SMS
Snow Device Management	si c: m:	AWS_ENDPOINT_URL_SNOW_DEVICE_MANAGEMENT
Snowball	si	AWS_ENDPOINT_URL_SNOWBALL
SNS	si	AWS_ENDPOINT_URL_SNS
SQS	sc	AWS_ENDPOINT_URL_SQS
SSM	s:	AWS_ENDPOINT_URL_SSM
SSM Contacts	s: c:	AWS_ENDPOINT_URL_SSM_CONTACTS
SSM Incidents	s: e:	AWS_ENDPOINT_URL_SSM_INCIDENTS
Ssm Sap	s:	AWS_ENDPOINT_URL_SSM_SAP

serviceId	공용 서비스별 환경 변수 AWS_ENDPOINT_URL_<SERVICE>
SSO	s: AWS_ENDPOINT_URL_SSO
SSO Admin	s: AWS_ENDPOINT_URL_SSO_ADMIN
SSO OIDC	s: AWS_ENDPOINT_URL_SSO_OIDC
SFN	s: AWS_ENDPOINT_URL_SFN
Storage Gateway	s: AWS_ENDPOINT_URL_STORAGE_GATEWAY
STS	s: AWS_ENDPOINT_URL_STS
SupplyChain	s: AWS_ENDPOINT_URL_SUPPLYCHAIN
Support	s: AWS_ENDPOINT_URL_SUPPORT
Support App	s: AWS_ENDPOINT_URL_SUPPORT_APP
SWF	s: AWS_ENDPOINT_URL_SWF

serviceId	공용 서비스별 환경 변수 AWS_ENDPOINT_URL_<SERVICE>
synthetics	synthetics AWS_ENDPOINT_URL_SYNTHETICS
Textract	textract AWS_ENDPOINT_URL_TEXTRACT
Timestream InfluxDB	timestream_influxdb AWS_ENDPOINT_URL_TIMESTREAM_INFLUXDB
Timestream Query	timestream_query AWS_ENDPOINT_URL_TIMESTREAM_QUERY
Timestream Write	timestream_write AWS_ENDPOINT_URL_TIMESTREAM_WRITE
tnb	tnb AWS_ENDPOINT_URL_TNB
Transcribe	transcribe AWS_ENDPOINT_URL_TRANSCRIBE
Transfer	transfer AWS_ENDPOINT_URL_TRANSFER

serviceId	공용 서비스별 환경 변수 AWS_ENDPOINT_URL_<SERVICE>
Translate	t: AWS_ENDPOINT_URL_TRANSLATE
TrustedAdvisor	t: AWS_ENDPOINT_URL_TRUSTEDADVISOR v:
VerifiedPermissions	v: AWS_ENDPOINT_URL_VERIFIEDPERMISSIONS e: s
Voice ID	v: AWS_ENDPOINT_URL_VOICE_ID
VPC Lattice	v: AWS_ENDPOINT_URL_VPC_LATTICE c:
WAF	w: AWS_ENDPOINT_URL_WAF
WAF Regional	w: AWS_ENDPOINT_URL_WAF_REGIONAL n:
WAFV2	w: AWS_ENDPOINT_URL_WAFV2

serviceId	공용 서비스별 엔드포인트 환경 변수
WellArchitected	w: AWS_ENDPOINT_URL_WELLARCHITECTED
Wisdom	w: AWS_ENDPOINT_URL_WISDOM
WorkDocs	w: AWS_ENDPOINT_URL_WORKDOCS
WorkLink	w: AWS_ENDPOINT_URL_WORKLINK
WorkMail	w: AWS_ENDPOINT_URL_WORKMAIL
WorkMailMessageFlow	w: AWS_ENDPOINT_URL_WORKMAILMESSAGEFLOW
WorkSpaces	w: AWS_ENDPOINT_URL_WORKSPACES
WorkSpaces Thin Client	w: AWS_ENDPOINT_URL_WORKSPACES_THIN_CLIENT

serviceId	공 유 용 서 비 스 스 발 자 크 A C 파 일	환경 변수
WorkSpaces Web	w s_	AWS_ENDPOINT_URL_WORKSPACES_WEB
XRay	x:	AWS_ENDPOINT_URL_XRAY

스마트 구성 기본값

스마트 구성 기본값 기능을 사용하면 AWS SDKs 다른 구성 설정에 대해 미리 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: 자바/코틀린 전용

이 설정을 사용하면 사용자 애플리케이션 아키텍처에 맞는 모드를 선택하여 애플리케이션에 최적화된 기본값을 제공할 수 있습니다. 만약 AWS SDK 설정에 명시적으로 설정된 값이 있으면 해당 값이 항상 우선합니다. 다음과 같은 경우 AWS SDK 설정에 명시적으로 설정된 값이 없고 레거시와 `defaults_mode` 같지 않은 경우 이 기능은 애플리케이션에 최적화된 다양한 설정에 대해 다양한 기본값을 제공할 수 있습니다. 설정에는 HTTP 통신 설정, 재시도 동작, 서비스 지역별 엔드포인트 설정 및 가능한 모든 SDK 관련 구성이 포함될 수 있습니다. 이 기능을 사용하는 고객은 일반 사용 시나리오에 맞게 조정된 새 구성 기본값을 얻을 수 있습니다. 과 같지 않은 경우 모범 사례가 발전

함에 따라 제공된 기본값이 변경될 수 있으므로 업그레이드 시 애플리케이션 테스트를 수행하는 `defaults_mode` 것이 좋습니다. legacy SDK

기본값: legacy

참고: 의 새 메이저 버전은 기본적으로 으로 SDKs 설정됩니다. standard

유효한 값:

- `legacy`— 설정 전에 SDK 다양하고 존재했던 기본 설정을 제공합니다 `defaults_mode`.
- `standard`— 대부분의 시나리오에서 안전하게 실행할 수 있는 최신 권장 기본값을 제공합니다.
- `in-region`— 표준 모드를 기반으로 하며 다음을 호출하는 애플리케이션에 맞게 조정된 최적화를 포함합니다. AWS 서비스 동일한 환경 내에서 AWS 리전.
- `cross-region`— 표준 모드를 기반으로 하며 다음을 호출하는 애플리케이션에 맞게 조정된 최적화를 포함합니다. AWS 서비스 다른 지역에서.
- `mobile`— 표준 모드를 기반으로 구축하며 모바일 애플리케이션에 맞게 조정된 최적화를 포함합니다.
- `auto`— 표준 모드를 기반으로 구축하며 실험적 기능을 포함합니다. 런타임 환경을 검색하여 적절한 설정을 자동으로 SDK 결정하려고 시도합니다. 자동 감지는 휴리스틱 기반이며 정확도가 100%는 아닙니다. 런타임 환경을 확인할 수 없는 경우 `standard` 모드가 사용됩니다. 자동 탐지는 [인스턴스 메타데이터](#)를 쿼리할 수 있으며, 이로 인해 지연 시간이 발생할 수 있습니다. 시작 지연 시간이 애플리케이션에 중요한 경우에는 명시적 지연 시간을 `defaults_mode`을 대신 선택하는 것이 좋습니다.

config 파일에서 이 값을 설정하는 예:

```
[default]
defaults_mode = standard
```

다음 파라미터는 `defaults_mode`의 선택에 따라 최적화될 수 있습니다.

- `retryMode`— SDK 시도가 재시도되는 방식을 지정합니다. [재시도 동작](#)을 참조하세요.
- `stsRegionalEndpoints`— SDK 결정 방법을 지정합니다. AWS 서비스 엔드포인트와 통신하는 데 사용하는 엔드포인트 AWS Security Token Service (AWS STS). 보세요 [AWS STS 리전 엔드포인트](#).
- `s3UsEast1RegionalEndpoints`— SDK 결정 방법을 지정합니다. AWS 해당 `us-east-1` 지역의 Amazon S3와 통신하는 데 사용하는 서비스 엔드포인트입니다.

- `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_mode`에 따라 달라집니다. 이러한 값은 현재 다음과 같이 설정되어 있습니다(변경될 수 있음).

파라미터	standard 모드	in-region 모드	cross-region 모드	mobile 모드
<code>retryMode</code>	standard	standard	standard	standard
<code>stsRegionalEndpoints</code>	regional	regional	regional	regional
<code>s3UsEast1RegionalEndpoints</code>	regional	regional	regional	regional
<code>connectTimeoutInMillis</code>	3100	1100	3100	30000
<code>tlsNegotiationTimeoutInMillis</code>	3100	1100	3100	30000

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).

다음과 호환됩니다. AWS SDKs

다음은 이 항목에 설명된 기능 및 설정을 SDKs 지원합니다. 모든 일부 예외가 기록됩니다. 모든 JVM 시스템 속성 설정은 에서 지원됩니다. AWS SDK for Java 그리고 AWS SDK for Kotlin 오직.

SDK	지원	참고 또는 추가 정보
AWS CLI v2	아니요	
SDKC++의 경우	예	최적화되지 않은 매개변수:stsRegionalEndpoints ,s3UsEast1RegionalEndpoints ,tlsNegotiationTimeoutInMillis .
SDK고 V2 (1.x) 용	예	최적화되지 않은 매개변수:retryMode ,stsRegionalEndpoints ,s3UsEast1RegionalEndpoints .
SDKGo 1.x (V1) 용	아니요	
SDK자바 2.x의 경우	예	최적화되지 않은 매개변수:stsRegionalEndpoints .
SDK자바 1.x의 경우	아니요	
SDK3.x용 JavaScript	예	최적화되지 않은 매개변수:stsRegionalEndpoints ,s3UsEast1RegionalEndpoints ,tlsNegotiationTimeoutInMillis . connectTimeoutInMillis 가 connectionTimeout 호출됩니다.

SDK	지원	참고 또는 추가 정보
SDK2.x의 경우 JavaScript	아니요	
SDK코틀린의 경우	아니요	
SDK왜냐하면. NET3.x	예	최적화되지 않은 매개변수:connectTimeoutInMillis ,tlsNegotiationTimeoutInMillis .
SDK3.x의 경우 PHP	예	최적화되지 않은 매개변수:tlsNegotiationTimeoutInMillis .
SDK파이썬용 (보통3)	예	최적화되지 않은 매개변수:tlsNegotiationTimeoutInMillis .
SDK루비 3.x의 경우	예	
SDK러스트용	아니요	
SDK스위프트용	아니요	
를 위한 도구 PowerShell	예	최적화되지 않은 매개변수:connectTimeoutInMillis ,tlsNegotiationTimeoutInMillis .

AWS 공통 런타임(CRT) 라이브러리

AWS 공통 런타임(CRT) 라이브러리는 의 기본 라이브러리입니다 SDKs. CRT 는 C로 작성된 독립 패키지의 모듈식 제품군입니다. 각 패키지는 다양한 필수 기능에 대해 우수한 성능과 최소한의 공간을 제공합니다. 이러한 기능은 공통적이며 모든 기능에서 공유 SDKs되므로 코드 재사용, 최적화 및 정확도가 향상됩니다. 패키지는 다음과 같습니다.

- [awslabs/aws-c-auth](#): AWS 클라이언트 측 인증(표준 자격 증명 공급자 및 서명(sigv4))
- [awslabs/aws-c-cal](#): 암호화 기본 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, 프리루드, 페이로드, crc/트레일러), 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](#): AWS IoT 클라우드 서비스와 디바이스 통합의 C99 구현
- [awslabs/aws-c-mqtt](#): 사물 인터넷 (IoT)을 위한 간단한 표준 메시징 프로토콜
- [awslabs/aws-c-s3](#): Amazon S3 서비스와 통신하기 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](#): 플랫폼 간 하드웨어 가속화 CRC32c 및 효율적인 소프트웨어 구현으로 CRC32 대체
- [awslabs/aws-1c](#): Google BoringSSL 프로젝트 및 OpenSSL 프로젝트의 코드를 기반으로 AWS 및 해당 고객을 위해 AWS 암호화 팀이 유지 관리하는 범용 암호화 라이브러리
- [awslabs/s2n](#): 보안을 우선으로 하여 작고 빠르게 설계된 TLS/SSL 프로토콜의 C99 구현

CRT 는 Go 및 Rust를 SDKs 제외한 모든 를 통해 사용할 수 있습니다.

CRT 종속성

CRT 라이브러리는 관계 및 종속성의 복잡한 네트워크를 형성합니다. 이러한 관계를 알고 있으면 소스에서 CRT 직접 를 빌드해야 하는 경우 유용합니다. 그러나 대부분의 사용자는 언어SDK(예 AWS SDK:

C++ 또는 AWS SDK Java용) 또는 언어 IoT 디바이스SDK(예: C++SDK용 AWS IoT 또는 JavaSDK용 AWS IoT)를 통해 CRT 기능에 액세스합니다. 다음 다이어그램에서 언어 CRT 바인딩 상자는 특정 언어에 대한 CRT 라이브러리를 래핑하는 패키지를 나타냅니다. 이는 양식의 패키지 모음으로 `aws-crt-*`, 여기서 '*'는 SDK 언어(예: [aws-crt-cpp](#) 또는)입니다 [aws-crt-java](#).

다음은 CRT 라이브러리의 계층적 종속성을 보여주는 그림입니다.

AWS SDK 및 도구 유지 관리 정책

개요

이 문서에서는 모바일 및 IoT SDK를 포함한 AWS 소프트웨어 개발 키트 (SDK) 및 도구의 유지 관리 정책과 기본 종속성을 간략하게 설명합니다. AWS 신규 또는 업데이트된 AWS API, 새로운 기능, 개선 사항, 버그 수정, 보안 패치 또는 설명서 업데이트에 대한 지원이 포함될 수 있는 업데이트를 AWS SDK 및 도구에 정기적으로 제공합니다. 업데이트는 종속성, 언어 런타임 및 운영 체제의 변경 사항을 해결할 수도 있습니다. AWS SDK 릴리스는 패키지 관리자 (예: Maven, NuGet PyPI) 에 게시되며 소스 코드로 제공됩니다. GitHub

사용자는 SDK 릴리스를 계속 up-to-date 사용하여 최신 기능, 보안 업데이트 및 기본 종속성을 따라갈 것을 권장합니다. 지원되지 않는 SDK 버전을 계속 사용하는 것은 권장되지 않으며 그에 따른 책임은 사용자에게 있습니다.

버전 관리

AWS SDK 릴리스 버전은 X.Y.Z 형식입니다. 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 라이프사이클

주요 SDK 및 도구 버전의 라이프사이클은 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. AWS 릴리스가 안정적인 제품인 것으로 확인되면 해당 릴리스를 릴리스 후보로 표시할 수 있습니다. 릴리스 후보는 중대한 버그가 나타나지 않는 한 GA 릴리스가 될 준비가 되어 있으며 완전한 AWS 지원을 받게 됩니다.
- 일반 공급 (GA) (1단계) - 이 단계에서는 SDK가 완전히 지원됩니다. AWS 새 서비스 지원, 기존 서비스에 대한 API 업데이트, 버그 및 보안 수정이 포함된 정기 SDK 릴리스를 제공합니다. Tools의 경우

새로운 기능 업데이트 및 버그 수정이 포함된 정기 릴리스를 제공할 AWS 예정입니다. AWS SDK의 GA 버전을 최소 24개월 동안 지원합니다.

- 유지 관리 발표 (2단계) - AWS SDK가 유지 관리 모드로 전환되기 최소 6개월 전에 공개적으로 발표합니다. 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 계속 유효합니다. GitHub 리포지토리는 보관될 수 있습니다. 도달한 SDK는 사용자의 end-of-support 재량에 따라 사용할 수 있습니다. 새 메이저 버전으로 업그레이드할 것을 당사는 권장합니다.

다음은 SDK 메이저 버전 라이프사이클을 시각적으로 보여줍니다. 아래 표시된 타임라인은 예시이며 구속력이 없다는 점에 유의하십시오.

종속성 라이프사이클

대부분의 AWS SDK에는 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 제품에 대한 자체 end-of-support 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 메이저 SDK 버전을 늘리지 않고도 기본 종속성에 대한 지원을 중단할 권리가 있습니다.

통신 메서드

유지 관리 공고는 여러 방법으로 전달됩니다.

- 영향 받는 계정에는 특정 SDK 버전에 대한 지원 종료 계획을 알리는 이메일 공지가 발송됩니다. 이 메일은 경로를 설명하고 end-of-support, 캠페인 일정을 지정하고, 업그레이드 지침을 제공합니다.
- AWS API 참조 문서, 사용자 가이드, SDK 제품 마케팅 페이지, GitHub Readme 등의 SDK 설명서가 업데이트되어 캠페인 일정을 나타내고 영향을 받는 애플리케이션 업그레이드에 대한 지침을 제공합니다.
- 캠페인 경로를 설명하고 캠페인 일정을 다시 설명하는 AWS 블로그 게시물이 게시됩니다. end-of-support
- SDK에 지원 중단 경고가 추가되어 SDK 설명서에 대한 경로와 링크를 설명합니다. end-of-support

사용 가능한 AWS SDK 및 도구의 주요 버전 목록과 유지 관리 수명 주기 내 위치를 보려면 [여기](#)를 참조하십시오. [버전 지원](#)

AWS SDKs 및 툴 버전 지원

아래 표에는 사용 가능한 목록이 나와 있습니다. AWS 소프트웨어 개발 키트 (SDK) 메이저 버전 및 주요 버전이 유지 관리 라이프사이클에서 차지하는 위치 및 관련 일정 주요 버전의 수명 주기에 대한 자세한 내용은 AWS SDKs 및 도구 및 기본 종속성을 참조하십시오 [유지 관리 정책](#).

SDK	메이저 버전	현재 단계	일반 출시 날짜	주석
AWS CLI	1.x	정식 출시	2013년 9월 2일	
AWS CLI	2.x	정식 출시	2020년 2월 10일	
SDKC++의 경우	1.x	정식 출시	2015년 9월 2일	
SDK고 V2의 경우	V2 1.x	정식 출시	2021년 1월 19일	
SDK바독용	1.x	유지 관리	2015년 11월 19일	세부 정보 및 날짜는 공지를 참조 하십시오.
SDK자바의 경우	1.x	유지 관리	2010년 3월 25일	세부 정보 및 날짜는 공지를 참조 하십시오.
SDK자바의 경우	2.x	정식 출시	11/20/2018	
SDK에 대한 JavaScript	1.x	지원 종료	2013년 5월 6일	
SDK에 대한 JavaScript	2.x	유지 관리	2014년 6월 19일	세부 정보 및 날짜는 공지를 참조 하십시오.
SDK에 대한 JavaScript	3.x	정식 출시	2020년 12월 15일	
SDK코틀린의 경우	1.x	정식 출시	2023년 11월 27일	

SDK	메이저 버전	현재 단계	일반 출시 날짜	주석
SDK에 대한 .NET	1.x	지원 종료	2009년 11월	
SDK에 대한 .NET	2.x	지원 종료	2013년 11월 8일	
SDK에 대한 .NET	3.x	정식 출시	2015년 7월 28일	
SDK에 대한 PHP	2.x	지원 종료	2012년 11월 2일	
SDK에 대한 PHP	3.x	정식 출시	2015년 5월 27일	
SDK파이썬용 (보토2)	1.x	지원 종료	2011년 7월 13일	
SDK파이썬용 (보토3)	1.x	정식 출시	2015년 6월 22일	
SDK파이썬용 (보토코어)	1.x	정식 출시	2015년 6월 22일	
SDK루비의 경우	1.x	지원 종료	2011년 7월 14일	
SDK루비를 위해	2.x	지원 종료	2015년 2월 15일	
SDK루비를 위해	3.x	정식 출시	2017년 8월 29일	
SDK러스트를 위해	1.x	정식 출시	2023년 11월 27일	
SDK스위프트의 경우	1.x	정식 출시	2024년 9월 17일	
다음에 위한 도구 PowerShell	2.x	지원 종료	2013년 11월 8일	

SDK	메이저 버전	현재 단계	일반 출시 날짜	주석
다음에 위한 도구 PowerShell	3.x	지원 종료	2015년 7월 29일	
다음에 위한 도구 PowerShell	4.x	정식 출시	2019년 11월 21일	

언급되지 않은 SDK OR 도구를 찾고 계신가요? 예를 들어 암호화 SDKs, IoT 디바이스 SDKs, 모바일은 이 가이드에 포함되어 있지 않습니다. 이러한 다른 도구에 대한 설명서를 찾으려면 [기본 도구를 참조하십시오. AWS.](#)

에 대한 문서 기록 AWS SDKs 및 도구 참조 가이드

다음 표에는 주요 추가 사항 및 업데이트에 대한 설명이 나와 있습니다. AWS SDKs 및 도구 참조 안내서. 이 설명서의 업데이트에 대한 알림을 받으려면 RSS 피드를 구독할 수 있습니다.

변경 사항	설명	날짜
설정 SDK 참조에 Swift 추가	모든 설정 참조 호환성에 Swift SDK 지원 추가 AWS SDKs 표.	2024년 9월 17일
SDK자바 1.x 시스템 속성의 경우	지원되는 JVM 시스템 구성 설정에 대한 세부 정보를 다음과 같이 추가합니다. AWS SDK for Java 1.x.	2024년 5월 30일
설정 업데이트	JVM 시스템 구성 설정을 추가합니다.	2024년 3월 27일
호환성 표 업데이트	SDK 지원을 위한 호환성 업데이트, IAM Identity Center 절차 업데이트	2024년 2월 20일
컨테이너 보안 인증 업데이트. IMDS 업데이트.	Amazon에 대한 지원 추가 EKS. IMDSv1 폴백을 비활성화하는 설정 추가	2023년 12월 29일
요청 압축	요청 압축 기능에 대한 설정 추가.	2023년 12월 27일
호환성 테이블	Kotlin, Rust, SDK SDK 및 에 SDK 대한 호환성 표 및 도구 기능을 포함하도록 업데이트되었습니다. AWS Tools for PowerShell.	2023년 12월 10일
인증 업데이트	지원되는 인증 방법 SDKs 및 도구 업데이트.	2023년 7월 1일

IAM베스트 프랙티스 업데이트	가이드가 IAM 모범 사례에 맞게 업데이트되었습니다. 자세한 내용은 의 보안 모범 사례를 참조하십시오. IAM	2023년 2월 27일
SSO업데이트	새 SSO 토큰 구성의 SSO 자격 증명 업데이트.	2022년 11월 19일
설정 업데이트	일반 구성 및 Amazon S3 다중 리전 액세스 포인트 지원 표 업데이트.	2022년 11월 17일
설정 업데이트	IMDS클라이언트 및 IMDS 자격 증명의 명확성을 업데이트했습니다. 환경 변수 업데이트.	2022년 11월 4일
시작 페이지 업데이트	아마존 CodeWhisperer 발표.	2022년 9월 22일
싱글 사인은 서비스 이름 변경	이를 반영한 업데이트 AWS SSO지금은 다음과 같이 불립니다. 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 일 반에 공개되었습니다.	2020년 3월 13일

AWS 용어집

최신 AWS 용어는 AWS 용어집 참조서의 [AWS 용어집](#)을 참조하세요.

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