



참조 안내서

AWS SDK 및 도구



AWS SDK 및 도구: 참조 안내서

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

Amazon의 상표 및 브랜드 디자인은 Amazon 외 제품 또는 서비스와 함께, 브랜드 이미지를 떨어뜨리거나 고객에게 혼동을 일으킬 수 있는 방식으로 사용할 수 없습니다. Amazon이 소유하지 않은 기타 모든 상표는 Amazon 계열사, 관련 업체 또는 Amazon의 지원 업체 여부에 상관없이 해당 소유자의 자산입니다.

Table of Contents

AWS SDK 및 도구 참조 가이드	1
개발자 리소스	2
툴킷 텔레메트리 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
웹 ID 또는 OpenID Connect와 페더레이션	28
AWS 액세스 키	30
단기 보안 인증 정보를 사용합니다	30
장기 보안 인증 정보 사용	30
단기 보안 인증	31
장기 보안 인증	32
Amazon EC2 인스턴스의 IAM 역할	35
IAM 역할 생성	36

Amazon EC2 인스턴스 시작과 IAM 역할 지정	36
EC2 인스턴스에 연결	36
EC2 인스턴스에서 샘플 애플리케이션 실행	37
설정 참조	38
서비스 클라이언트 생성	38
설정의 우선 순위	38
Config 파일 설정 목록	39
Credentials 파일 설정 목록	43
환경 변수 목록	43
JVM 시스템 속성 목록	47
표준화된 보안 인증 공급자	50
보안 인증 공급자 체인	50
AWS 액세스 키	52
역할 공급자 수입	54
컨테이너 제공업체	60
IAM Identity Center 공급자	63
IMDS 공급사	69
프로세스 공급자	73
표준화된 기능	77
애플리케이션 ID	78
Amazon EC2 인스턴스 메타데이터	80
Amazon S3 액세스 포인트	83
Amazon S3 다중 리전 액세스 포인트	85
AWS 리전	87
AWS STS 지역화된 엔드포인트	90
이중 스택 엔드포인트 및 FIPS 엔드포인트	93
엔드포인트 검색	95
일반 구성	96
IMDS 클라이언트	100
재시도 동작	103
요청 압축	108
서비스별 엔드포인트	111
스마트 구성 기본값	165
공통 런타임	170
CRT 종속성	171
유지 관리 정책	172

개요	172
버전 관리	172
SDK 메이저 버전 라이프사이클	172
종속성 라이프사이클	173
통신 메서드	174
버전 지원 매트릭스	175
설명서 기록	178
AWS 용어집	180
.....	clxxxi

AWS SDK 및 도구 참조 가이드

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

이 안내서에는 다음과 관련된 정보가 포함되어 있습니다.

- [구성](#)— 공유 config 및 credentials 파일 또는 환경 변수를 사용하여 AWS SDK와 도구를 구성하는 방법.
- [인증 및 액세스](#)— 를 사용하여 개발할 AWS 때 코드나 도구가 인증되는 방식을 설정합니다. AWS 서비스
- [설정 참조](#) - 인증 및 구성에 사용할 수 있는 모든 표준 설정에 대한 참조
- [AWS공용 런타임 \(CRT\) 라이브러리](#)— 거의 모든 SDK에서 사용할 수 있는 공유 AWS 공용 런타임 (CRT) 라이브러리 개요.
- [AWS SDK 및 도구 유지 관리 정책](#)모바일 및 사물 인터넷 (IoT) SDK를 비롯한 AWS 소프트웨어 개발 키트 (SDK) 및 도구의 유지 관리 정책 및 버전 관리와 기본 종속성을 다룹니다.

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

사용 중인 제품:	설명서 관련 섹션:
<ul style="list-style-type: none"> • 모든 SDK 또는 도구 	AWS SDK 및 도구 유지 관리 정책
<ul style="list-style-type: none"> • AWS Cloud Development Kit (AWS CDK) 개발자 안내서 • AWS Serverless Application Model 개발자 안내서 • AWS Toolkit for Eclipse 사용 설명서 • AWS Toolkit for JetBrains 사용 설명서 • AWS Toolkit for Visual Studio 사용 설명서 • AWS Toolkit for Visual Studio Code 사용 설명서 	구성 인증 및 액세스 AWS SDK 및 도구 유지 관리 정책
<ul style="list-style-type: none"> • AWS Command Line Interface 사용 설명서 	구성

사용 중인 제품:	설명서 관련 섹션:
<ul style="list-style-type: none"> • AWS SDK for C++ 개발자 안내서 • AWS SDK for Go 개발자 안내서 • AWS SDK for Java 개발자 안내서 • AWS SDK for JavaScript 개발자 안내서 • AWS SDK for Kotlin • AWS SDK for .NET 개발자 안내서 • AWS SDK for PHP 개발자 안내서 • AWS 파이썬용 SDK (Boto3) 시작하기 • AWS SDK for Ruby 개발자 안내서 • AWS SDK for Rust • AWS SDK for Swift • AWS Tools for Windows PowerShell 사용 설명서 	<ul style="list-style-type: none"> • 인증 및 액세스 • 설정 참조 • AWS공용 런타임 (CRT) 라이브러리 • AWS SDK 및 도구 유지 관리 정책 • AWS SDK 및 툴 버전 지원 매트릭스

개발자 리소스

애플리케이션 개발에 도움이 되는 도구에 대한 개요는 빌드 AWS기반 [도구를](#) 참조하십시오. AWS지원에 대한 자세한 내용은 [AWS 지식 센터](#)를 참조하십시오.

Amazon Q Developer는 애플리케이션을 이해, 구축, 확장 및 운영하는 데 도움이 되는 생성형 AI 기반 대화형 도우미입니다. AWS 개발을 가속화하기 위해 Amazon Q를 지원하는 모델을 고품질 AWS 콘텐츠로 보강하여 보다 완전하고 실행 가능하며 참조할 수 있는 답변을 제공합니다. AWS자세한 내용은 [Amazon Q 개발자란 무엇입니까?](#) 를 참조하십시오. Amazon Q 개발자 사용 설명서에서 확인할 수 있습니다.

툴킷 텔레메트리 알림

AWS 통합 개발 환경 (IDE) 툴킷은 IDE에서 AWS 서비스에 액세스할 수 있게 해주는 플러그인 및 확장입니다. 각 IDE 툴킷에 대한 자세한 내용은 위 표의 툴킷 사용자 안내서를 참조하십시오.

AWS IDE 툴킷은 클라이언트측 텔레메트리 데이터를 수집 및 저장하여 향후 툴킷 릴리스에 대한 결정을 내릴 수 있습니다. AWS 수집된 데이터는 툴킷 사용량을 수치화합니다. AWS

모든 AWS IDE 툴킷에서 수집되는 텔레메트리 데이터에 대해 자세히 알아보려면 Github 리포지토리의 [CommonDefinitions.json](#) 문서를 참조하십시오. `aws-toolkit-common`

각 AWS IDE 툴킷에서 수집한 텔레메트리 데이터에 대한 자세한 내용은 다음 툴킷의 Github 리포지토리에 있는 리소스 문서를 참조하십시오. AWS

- [AWS Toolkit for Visual Studio](#)
- [AWS Toolkit for Visual Studio Code](#)
- [AWS Toolkit for JetBrains](#)

툴킷에서 액세스할 수 있는 특정 AWS 서비스는 추가 클라이언트 측 원격 측정 데이터를 수집할 수 있습니다. AWS 각 개별 AWS 서비스에서 수집하는 데이터 유형에 대한 자세한 내용은 관심 있는 특정 서비스의 [AWS 설명서](#) 항목을 참조하십시오.

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 AWS Command Line Interface (AWS CLI), AWS SDK 및 기타 도구에서 사용되는 카값 쌍의 구성 설정 세트입니다. 구성 값은 프로파일이 사용될 때 SDK/도구의 일부 측면을 구성하기 위해 프로파일에 첨부됩니다. 이러한 파일은 값이 사용자의 로컬 환경에 있는 모든 애플리케이션, 프로세스 또는 SDK에 적용된다는 점에서 “공유”됩니다.

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

프로파일

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

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

Note

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

이름이 지정된 프로필 설정

[default]프로필과 이름이 지정된 여러 프로필이 같은 파일에 존재할 수 있습니다. 다음 설정을 사용하여 코드를 실행할 때 SDK 또는 도구에서 사용할 프로필 설정을 선택합니다. 코드를 사용하여 작업할 때 코드 내에서 프로필을 선택하거나 명령어별로 프로필을 선택할 수도 있습니다. AWS CLI

다음 중 하나를 설정하여 이 기능을 구성하십시오.

AWS_PROFILE- 환경 변수

이 환경 변수를 명명된 프로필 또는 “기본”으로 설정하면 모든 SDK 코드 및 AWS CLI 명령이 해당 프로필의 설정을 사용합니다.

명령 행을 통한 환경 변수 설정의 Linux/macOS 예제:

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

명령줄을 통해 환경 변수를 설정하는 Windows 예제:

```
setx AWS_PROFILE "my_default_profile_name"
```

aws.profile- JVM 시스템 속성

[JVM의 Kotlin용 SDK와 Java 2.x용 SDK의 경우 시스템 속성을 설정할 수 있습니다.](#) `aws.profile` SDK는 서비스 클라이언트를 생성할 때 해당 설정이 코드에서 재정의되지 않는 한 명명된 프로필의 설정을 사용합니다. Java 1.x용 SDK는 이 시스템 속성을 지원하지 않습니다.

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]profile 섹션 식별자가 필요하지 않은 유일한 프로파일입니다.

다음은 [default] 프로파일이 있는 기본 config 파일을 보여주는 예입니다. [region](#) 설정값을 설정합니다. 다른 섹션 정의를 찾을 때까지 이 줄 뒤에 오는 모든 설정이 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 라인은 다양한 개발 시나리오에 적용할 수 있는 명명된 구성 그룹입니다. 명명된 프로파일의 이해를 높이려면 프로파일의 이전 섹션을 참조하십시오.

다음 예제는 profile 섹션 정의 라인과 이라는 이름의 config 프로파일이 있는 파일을 보여줍니다. foo 다른 섹션 정의를 찾을 때까지 이 줄 뒤에 오는 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 라인은 AWS 자격 증명을 확인하는 데 사용하는 프로파일을 구성하는 데 사용하는 설정 그룹의 이름을 지정합니다. AWS IAM Identity Center Single Sign-On 인증 구성에 대한 자세한 내용은 [IAM 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 예에서는 “my-sso”의 토큰을 사용하여 “SampleRole111122223333” 계정의 IAM 역할에 대한 단기 AWS 자격 증명을 가져올 프로필을 구성합니다. “my-sso” sso-session 섹션은 sso-session 키를 사용하여 profile 섹션에서 이름으로 참조됩니다.

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

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

섹션 유형: **services**

섹션 정의 줄의 예: [services dev]

Note

이 services 섹션은 서비스별 엔드포인트 사용자 정의를 지원하며 이 기능이 포함된 SDK 및 도구에서만 사용할 수 있습니다. SDK에서 이 기능을 사용할 수 있는지 확인하려면 서비스별 엔드포인트에 대한 [AWS SDK와의 호환성](#)(를)참조하십시오.

services 섹션 정의 라인은 요청에 대한 사용자 지정 엔드포인트를 구성하는 설정 그룹의 이름을 지정합니다. AWS 서비스 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 구분됩니다. 여기서 <SERVICE> 는 식별자 키입니다. AWS 서비스 AWS 서비스 식별자는 모든 공백을 serviceId 밑줄로 바꾸고 모든 문자를 소문자로 바꾸는 API 모델을 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 입니다. dynamodb Amazon DynamoDB 서비스 하위 섹션은 dynamodb = 라인에서 시작됩니다. 들여쓰기된 바로 다음 줄은 해당 하위 섹션에 포함되며 해당 서비스에 적용됩니다.

공유 config 및 credentials 파일 위치

공유 AWS config 및 credentials 파일은 AWS SDK 및 도구의 구성 정보가 들어 있는 일반 텍스트 파일입니다. 파일은 사용자 환경에 로컬로 존재하며 SDK 코드나 해당 환경에서 AWS CLI 실행하는 명령에 의해 자동으로 사용됩니다. 자체 컴퓨터에서 또는 Amazon Elastic Compute Cloud 인스턴스에서 개발하는 경우를 예로 들 수 있습니다.

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

기본적으로 파일은 사용자 home 또는 사용자 폴더에 .aws 있는 이름이 지정된 폴더에 있습니다.

운영 체제	기본 위치 및 파일 이름
Linux 및 macOS	~/.aws/config ~/.aws/credentials
Windows	%USERPROFILE%\aws\config %USERPROFILE%\aws\credentials

홈 디렉터리 해상도

~다음과 같은 경우에만 홈 디렉터리 확인에 사용됩니다.

- 경로를 시작합니다.
- 바로 뒤에 / 또는 플랫폼별 구분 기호가 옵니다. Windows에서는 ~/ ~\ 둘 다 홈 디렉터리로 확인됩니다.

홈 디렉터를 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 [환경 변수 지원](#)을 참조하십시오.

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. 대부분의 SDK에서 지원하는 환경 변수 목록은 [환경 변수 목록](#) 섹션을 참조하십시오.

옵션의 우선 순위

- 환경 변수를 사용하여 설정을 지정할 경우, 공유 AWS config 및 `credentials` 파일의 프로파일에서 로드된 값을 재정의합니다.
- AWS CLI 명령줄에서 파라미터를 사용하여 설정을 지정할 경우, 구성 파일에서 해당하는 환경 변수 또는 프로파일의 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 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 AWS_ACCESS_KEY_ID AKIAIOSFODNN7EXAMPLE
C:\> setx AWS_SECRET_ACCESS_KEY wJalrXUtnFEMI/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 예에 표시된 대로 PowerShell 프롬프트에서 환경 변수를 설정하면 현재 세션 기간에만 해당 값이 저장됩니다. 모든 PowerShell 및 명령 프롬프트 세션에서 환경 변수 설정이 영구적으로 적용되도록 하려면 제어판에서 시스템 애플리케이션을 사용하여 해당 설정을 저장합니다. 또는 PowerShell 프로파일에 변수를 추가하여 향후 모든 PowerShell 세션에 적용되도록 변수를 설정할 수 있습니다. 환경 변수 저장 또는 세션에 영구적 적용에 대한 자세한 내용은 [PowerShell 설명서](#)를 참조하세요.

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 Lambda를 사용하면 환경 변수를 직접 설정할 수 있습니다. 자세한 내용은 AWS Lambda 개발자 안내서의 [AWS Lambda 환경 변수 사용](#)을 참조하십시오.

서버리스 프레임워크에서는 환경 설정 아래의 공급자 키 아래에 있는 `serverless.yml` 파일에 SDK 환경 변수를 설정할 수 있는 경우가 많습니다. `serverless.yml` 파일에 대한 자세한 내용은 서버리스 프레임워크 설명서의 [일반 함수 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 참조를 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 쉘 세션이 종료될 때까지 또는 변수를 다른 값으로 설정할 때까지 유지됩니다. 쉘의 스타트업 스크립트에서 변수를 설정하면 해당 변수가 향후 세션에서도 영구적으로 적용되도록 할 수 있습니다.

Windows Command Prompt

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

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

AWS 서비스로 개발할 때는 코드가 AWS를 사용하여 인증하는 방법을 설정해야 합니다. 환경 및 사용 가능한 AWS 액세스 권한에 따라 다양한 방식으로 AWS 리소스에 대한 프로그래밍 방식의 액세스를 구성할 수 있습니다.

로컬 (AWS가 아닌 지역) 에서 실행되는 코드의 인증 옵션

- [IAM Identity Center 인증](#) - 보안 모범 사례로 IAM Identity Center 에서 AWS Organizations를 사용하여 모든 사용자 AWS 계정 액세스를 관리하는 것이 좋습니다. AWS IAM Identity Center에서 사용자를 생성하거나 Microsoft Active Directory를 사용하거나, SAML 2.0 ID 공급자 (IdP)를 사용하거나, IdP를 개별적으로 AWS 계정에 페더레이션할 수 있습니다. 사용자 리전에서 IAM Identity Center를 지원하는지 확인하려면 Amazon Web Services 일반 참조의 [AWS IAM Identity Center 엔드포인트 및 할당량](#)을 참조하세요.
- [IAM Roles Anywhere](#) - IAM Roles Anywhere를 사용하여 AWS 외부에서 실행되는 서버, 컨테이너 및 애플리케이션과 같은 워크로드에 대한 IAM의 임시 보안 인증을 얻을 수 있습니다. IAM Roles Anywhere를 사용하려면 워크로드에 X.509 인증서를 사용해야 합니다.
- [역할 수임](#) - IAM 역할을 수임하여 다른 방법으로는 액세스할 수 없는 AWS 리소스에 일시적으로 액세스할 수 있습니다.
- [AWS 액세스 키](#) — 편리하지 않거나 AWS 리소스의 보안 위험을 증가시킬 수 있는 기타 옵션.

AWS 환경 내에서 실행되는 코드의 인증 옵션

- [Amazon EC2 인스턴스의 IAM 역할 사용](#) - Amazon EC2 인스턴스에서 IAM 역할을 사용하여 사용자 애플리케이션을 안전하게 실행합니다.
- 다음과 같은 방법으로 IAM Identity Center를 사용하여 프로그래밍 방식으로 AWS와 상호 작용할 수 있습니다.
 - 콘솔에서 AWS CLI 명령을 실행하는 데 [AWS CloudShell](#)을 사용합니다.
 - AWS 리소스와 포함된 통합 개발 환경 (IDE)을 AWS 에서 프로그래밍을 시작할 때 [AWS Cloud9](#)을 사용합니다.
 - 소프트웨어 개발 팀을 위한 클라우드 기반 협업 공간은 [Amazon CodeCatalyst](#)를 고려하십시오.

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

AWS에 대한 액세스가 필요한 모바일 애플리케이션이나 클라이언트 기반 웹 애플리케이션을 생성하는 경우, 웹 ID 페더레이션을 사용하여 임시 AWS 보안 인증 정보를 요청하도록 앱을 구축하십시오.

웹 ID 페더레이션을 사용하면 사용자 정의 로그인 코드를 생성하거나 자신의 사용자 보안 인증을 관리할 필요가 없습니다. 대신에, 앱 사용자는 Login with Amazon, Facebook, Google 또는 다른 OpenID Connect(OIDC) 호환 IdP와 같은 널리 알려진 외부 ID 제공업체 (IdP)를 사용해 로그인할 수 있습니다. 앱 사용자는 인증 토큰을 받은 다음, 해당 토큰을 AWS 계정의 리소스를 사용할 수 있는 권한을 가진 IAM 역할에 매핑되는 임시 보안 인증으로 바꿉니다.

SDK 또는 도구에 맞게 이를 구성하는 방법을 알아보려면 [웹 ID 또는 OpenID Connect와 페더레이션을](#) 참조하십시오.

모바일 애플리케이션의 경우 Amazon Cognito를 사용하는 것이 좋습니다. Amazon Cognito는 ID 브로커로 활동하며 사용자를 대신하여 상당한 페더레이션을 합니다. 자세한 정보는 IAM 사용 설명서의 [모바일 애플용 Amazon Cognito](#)를 참조하십시오.

액세스 관리에 대한 추가 정보

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

- [IAM ID \(사용자, 그룹 및 역할\)](#) - AWS의 기본 ID를 숙지하십시오.
- [IAM의 보안 모범 사례](#) — [공유 책임 모델](#)에 따라 AWS 애플리케이션을 개발할 때 따라야 할 보안 권장 사항.

Amazon Web Services 일반 참조에는 다음에 대한 기본 사항이 있습니다.

- [AWS보안 인증 이해 및 취득](#) — 콘솔 및 프로그래밍 방식 액세스 모두에 대한 액세스 키 옵션 및 관리 관행.

AWS Builder ID

이미 소유하고 있거나 생성하고 싶은 모든 AWS 계정의 보완. AWS 계정이 상요자가 생성하고 해당 리소스의 보안 경계가 되는 AWS 리소스의 컨테이너 역할을 하는 동안 사용자의 AWS Builder ID는 사용자를 개인으로 나타냅니다. AWS Builder ID로 로그인하여 Amazon CodeWhisperer 및 Amazon CodeCatalyst와 같은 개발자 도구 및 서비스에 액세스할 수 있습니다.

- AWS 로그인 사용 설명서의 [AWS Builder ID으로 로그인](#) - AWS Builder ID의 생성 및 사용 방법을 알아보고 Builder ID가 제공하는 내용을 알아봅니다.

- AWS Toolkit와 CodeWhisperer 사용한 인증- CodeWhisperer사용자 설명서의 ID 구축기 — AWS Builder ID를 사용한 앱 사용 방식을 알아보세요.
- Amazon CodeCatalyst 사용 설명서의 [CodeCatalyst 개념 - AWS Builder ID](#) - CodeCatalyst에서 AWS Builder ID를 사용하는 방법을 알아보세요.

IAM Identity Center 인증

AWS IAM Identity Center 비AWS 컴퓨팅 서비스에서 개발할 때 AWS 자격 증명을 제공하는 권장 방법입니다. 예를 들어 이는 사용자의 로컬 개발 환경과 같은 것입니다. Amazon Elastic Compute Cloud (Amazon EC2) AWS Cloud9 또는 같은 AWS 리소스에서 개발하는 경우 대신 해당 서비스에서 자격 증명을 받는 것이 좋습니다.

이 자습서에서는 IAM ID 센터 액세스를 설정하고 AWS 액세스 포털과 를 사용하여 SDK 또는 도구에 맞게 구성합니다. AWS CLI

- AWS 액세스 포털은 IAM ID 센터에 수동으로 로그인하는 웹 위치입니다. URL 형식은 d-xxxxxxx.awsapps.com/start 또는 *your_subdomain*.awsapps.com/start입니다. AWS 액세스 포털에 로그인하면 해당 사용자에 대해 구성된 역할을 볼 AWS 계정 수 있습니다. 이 절차에서는 AWS 액세스 포털을 사용하여 SDK/도구 인증 프로세스에 필요한 구성 값을 가져옵니다.
- AWS CLI 코드에 의한 API 호출에 대해 IAM Identity Center 인증을 사용하도록 SDK 또는 도구를 구성하는 데 사용됩니다. 이 일회성 프로세스는 공유 AWS config 파일을 업데이트하여 코드를 실행할 때 SDK 또는 도구에서 이 파일을 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 사용 [설명서의 IAM ID 센터 활성화를 참조하십시오](#). AWS IAM Identity Center

다음 방법 중 하나를 선택하여 자격 증명에 액세스하십시오. AWS

IAM ID 센터를 통한 액세스 권한을 설정하지 않았습니다.

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

- AdministratorAccess 권한 집합은 정규 개발에 사용해서는 안 됩니다. 고용주가 이러한 목적으로 사용자 정의 PowerUserAccess 권한 집합을 만들지 않은 이상, 대신 사전 정의된 권한 집합을 사용하는 것이 좋습니다.

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

- 그룹을 생성하는 대신 *Admin team Dev team* 그룹을 생성한 다음 지침에서 이를 대체하십시오.
- 기존 사용자를 사용할 수 있지만 새 *Dev team* 그룹에 사용자를 추가해야 합니다.
- 사용 권한 집합을 만드는 대신 *AdministratorAccess* 사용 권한 집합을 만든 다음 지침에서 *PowerUserAccess* 사용 권한 집합을 만든 후 이를 대체하십시오.

작업을 마쳤으면 다음 사항을 준비해야 합니다.

- Dev team 그룹.
 - Dev team 그룹에 첨부된 PowerUserAccess 권한 세트입니다.
 - 사용자가 Dev team 그룹에 추가되었습니다.
- 포털을 종료하고 다시 로그인하여 Administrator OR에 대한 AWS 계정 및 옵션을 확인하세요. PowerUserAccess 도구/SDK로 작업할 때 선택하세요.

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

ID 제공업체 포털을 AWS 통해 로그인하세요. 클라우드 관리자가 사용자 PowerUserAccess (개발자)에게 권한을 부여한 경우 액세스 권한이 AWS 계정 있는 권한과 권한 설정을 확인할 수 있습니다. 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 AWS 액세스 포털을 AWS 통해 이미 접속했습니다.

AWS 액세스 포털을 AWS 통해 로그인하세요. 클라우드 관리자가 사용자 PowerUserAccess (개발자)에게 권한을 부여한 경우 액세스할 수 있는 AWS 계정 있는 권한과 권한 설정을 확인할 수 있습니다. 권한 집합 이름 옆에는 해당 권한 집합을 사용하여 수동으로 또는 프로그래밍 방식으로 계정에 액세스할 수 있는 옵션이 표시됩니다.

고용주가 관리하는 페더레이션 사용자 지정 ID 공급자를 AWS 통해 이미 액세스 권한을 가지고 있습니다.

IT 팀에 문의하십시오.

2 단계: IAM Identity Center를 사용할 SDK 및 도구 구성

1. 사용자 개발 시스템에 최신 AWS CLI을 설치합니다.
 - a. 자세한 내용은 AWS Command Line Interface 사용 설명서의 [AWS CLI 최신 버전의 설치 또는 업데이트](#)를 참조하십시오.
 - b. (선택 사항) 제대로 작동하는지 확인하려면 명령 프롬프트를 열고 `aws --version` 명령을 실행합니다. AWS CLI
2. AWS 액세스 포털에 로그인합니다. 사용자의 고용주가 이 URL을 제공할 수도 있으며 1단계: 액세스 설정 후 이메일로 받을 수도 있습니다. 그렇지 않은 경우 <https://console.aws.amazon.com/singlesignon/> 대시보드에서 AWS 액세스 포털 URL을 찾아보세요.
 - a. AWS 액세스 포털의 계정 탭에서 관리할 개별 계정을 선택합니다. 사용자의 역할이 표시됩니다. 적절한 권한 집합에 대한 명령줄 또는 프로그래밍 방식 액세스를 위한 자격 증명을 가져오려면 액세스 키를 선택합니다. 사전 정의된 PowerUserAccess 권한 세트를 사용하거나, 사용자 또는 고용주가 개발을 위한 최소 권한을 적용하기 위해 생성한 권한 세트를 사용할 수 있습니다.
 - b. 보안 인증 가져오기 대화 상자에서 운영 체제에 따라 MacOS 및 Linux 또는 Windows를 선택합니다.
 - c. IAM Identity Center 보안 인증 방법을 선택하여 다음 단계에 필요한 SSO Start URL 및 SSO Region 값을 가져옵니다.
3. AWS CLI 명령 프롬프트에서 명령을 실행합니다. `aws configure sso` 메시지가 표시되면 이전 단계에서 수집한 구성 값을 입력합니다. 이 AWS CLI 명령에 대한 자세한 내용은 [aws configure sso 마법사를 통한 프로필 구성](#)을 참조하십시오.
 - CLI 프로파일 이름의 경우 시작할 때 `###`을 입력하는 것이 좋습니다. 기본이 아닌 (명명된) 프로파일 및 관련 환경 변수를 설정하는 방법에 대한 자세한 내용은 [프로파일](#)을 참조하십시오.
4. (선택 사항) AWS CLI 명령 프롬프트에서 명령을 실행하여 활성 세션 ID를 확인합니다. `aws sts get-caller-identity` 응답에는 구성된 IAM Identity Center 권한 세트가 표시되어야 합니다.
5. AWS SDK를 사용하는 경우 개발 환경에서 SDK용 애플리케이션을 생성하십시오.

- a. 일부 SDK의 경우 IAM Identity Center 인증을 사용하려면 먼저 SSO과 SSO0IDC와 같은 추가 패키지를 애플리케이션에 추가해야 합니다. 자세한 내용은 특정 SDK를 참조하십시오.
- b. 이전에 액세스를 AWS구성한 경우 공유 AWS credentials 파일에 대한 액세스 권한이 있는지 검토하세요. [AWS 액세스 키 보안 인증 공급자 체인](#) 우선 순위로 인해 SDK 또는 도구에서 IAM Identity Center 보안 인증을 사용하려면 먼저 정적 보안 인증을 반드시 제거해야 합니다.

SDK 및 도구가 이 구성을 사용하여 보안 인증을 사용하고 새로 고치는 방법에 대한 자세한 내용은 [IAM Identity Center 인증 이해하기](#)를 참조하십시오.

사용자 구성 세션 길이에 따라 사용자의 액세스가 결국 만료되고 SDK와 도구는 인증 오류를 만나게 됩니다. 필요할 때 액세스 포털 세션을 다시 새로 AWS CLI 고치려면 를 사용하여 `aws sso login` 명령을 실행합니다.

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

- IAM Identity Center 세션 기간 - [사용자의 AWS 액세스 포털 세션 기간을 구성합니다.](#)
- 권한 설정 세션 기간 - 세션 기간을 [설정합니다](#)

SDK 및 도구의 모든 IAM Identity Center 공급자 설정에 대한 자세한 내용은 이 안내서의 [IAM 아이덴티티 센터 보안 인증 공급자](#)를 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.

역할 수입

역할 수입에는 액세스 권한이 없을 수 있는 AWS 리소스에 액세스하기 위해 일련의 임시 보안 자격 증명을 사용하는 것이 포함됩니다. 이러한 임시 자격 증명은 액세스 키 ID, 보안 액세스 키 및 보안 토큰

으로 구성됩니다. AWS Security Token Service(AWS STS) API 요청에 대해 자세히 알아보려면 AWS Security Token Service API 참조의 [작업](#)을 참조하세요.

역할을 수임하도록 SDK 또는 도구를 설정하려면 먼저 수임할 특정 역할을 만들거나 식별해야 합니다. IAM 역할은 Amazon 리소스 이름(ARN) 역할로 고유하게 식별됩니다. 역할은 다른 엔티티와 신뢰 관계를 구축합니다. 역할을 사용하는 신뢰할 수 있는 엔티티는 AWS 서비스, 또 다른 AWS 계정, 웹 ID 공급자 또는 OIDC 또는 SAML 페더레이션일 수 있습니다. IAM 역할에 대한 자세한 내용은 IAM 사용 설명서의 [IAM 역할](#) 섹션을 참조하십시오.

IAM 역할을 식별한 후 해당 역할을 신뢰할 수 있는 경우 해당 역할에서 부여한 권한을 사용하도록 SDK 또는 도구를 구성할 수 있습니다. 이 작업을 수행하려면 [IAM 역할 수임](#) 또는 [웹 ID 또는 OpenID Connect와 페더레이션](#) 둘 중 하나를 선택하세요.

IAM 역할 수임

역할 수임 시 AWS STS는 일련의 임시 보안 자격 증명을 반환합니다. 이러한 자격 증명은 다른 프로파일이나 코드가 실행되는 인스턴스 또는 컨테이너에서 제공됩니다. 역할 수임의 다른 예로는 Amazon EC2에서 여러 AWS 계정 계정을 관리하거나, AWS 계정 전체의 AWS CodeCommit을 사용하거나, AWS CodeBuild와 다른 계정에 액세스하는 경우를 들 수 있습니다.

1단계: IAM 역할 설정

역할을 수임하도록 SDK 또는 도구를 설정하려면 먼저 수임할 특정 역할을 만들거나 식별해야 합니다. IAM 역할은 역할 [ARN](#)을 사용하여 고유하게 식별됩니다. 역할은 일반적으로 계정 내에서 또는 크로스 계정 액세스를 위해 다른 엔티티와 신뢰 관계를 구축합니다. 자세한 내용은 IAM 사용 설명서의 [IAM 역할 생성](#)을 참조하십시오.

2단계: SDK 또는 도구 구성

`credential_source` 또는 `source_profile`에서 자격 증명을 소싱하도록 SDK 또는 도구를 구성합니다.

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

다른 프로파일에서 자격 증명을 소싱하는 데 `source_profile`을(를) 사용합니다.

`source_profile`은 또한 수임된 역할을 사용하여 다른 역할을 수임하는 프로파일 계층 구조인 역할 체인을 지원합니다.

프로파일에서 이를 지정할 때 SDK 또는 도구에서 해당 AWS STS [AssumeRole](#) API 호출을 수행합니다. 역할을 수임하여 임시 자격 증명을 검색하고 사용하려면 공유 AWS config 파일에서 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 ID 또는 OpenID Connect와 페더레이션

AWS에 액세스가 필요한 모바일 애플리케이션 또는 클라이언트 기반 웹 애플리케이션을 생성할 때 AWS STS에서는 퍼블릭 ID 제공업체(idP)를 통해 인증된 페더레이션 사용자의 임시 보안 자격 증명 세트를 반환합니다. 퍼블릭 자격 증명 공급자의 예에는 Login with Amazon, Facebook, Google 또는 OpenID Connect(OIDC) 호환 자격 증명 공급자 등이 있습니다. 이 방법을 사용하면 사용자는 자신의 AWS 또는 IAM ID가 필요하지 않습니다.

Amazon Elastic Kubernetes Service를 사용하는 경우 이 기능을 사용하면 각 컨테이너에 대해 서로 다른 IAM 역할을 지정할 수 있습니다. Kubernetes는 이 자격 증명 공급자가 임시 자격 증명을 획득하는데 사용하는 OIDC 토큰을 컨테이너에 배포하는 기능을 제공합니다. 이 Amazon EKS 구성에 대한 자세한 내용은 Amazon EKS 사용 설명서의 [서비스 계정에 대한 IAM 역할](#)을 참조하세요. 그러나 더 간단한 옵션을 원하면 [SDK에서 지원하는 경우 Amazon EKS Pod Identity](#)를 대신 사용하는 것이 좋습니다.

1단계: 자격 증명 공급자 및 IAM 역할 설정

외부 idP와의 페더레이션을 구성하려는 경우 IAM ID 제공업체를 생성하여 외부 IdP 및 구성에 대해 AWS에 알려줍니다. 이렇게 하면 AWS 계정 계정과 외부 IdP 사이에 신뢰가 구축됩니다. 인증에 웹 자격 증명 토큰을 사용하도록 SDK를 구성하기 전에 먼저 ID 제공업체(idP)와 액세스에 사용되는 IAM 역할을 설정해야 합니다. 이러한 설정을 하려면 IAM 사용 설명서의 [웹 자격 증명 또는 OpenID Connect 페더레이션을 위한 역할 생성\(콘솔\)](#)을 참조하십시오.

2단계: SDK 또는 도구 구성

인증을 위해 AWS STS의 웹 ID 토큰을 사용하도록 SDK 또는 도구를 구성합니다.

프로파일에서 이를 지정할 때 SDK 또는 도구에서 해당 AWS STS [AssumeRoleWithWebIdentity](#) API 호출을 수행합니다. 웹 ID 페더레이션을 사용하여 임시 자격 증명을 검색하고 사용하려면 공유 AWS config 파일에서 다음 구성 값을 지정합니다. 이러한 설정에 대한 자세한 내용은 [역할 보안 인증 제공자 수입 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. 하지만 Amazon Cognito 자격 증명 공급자는 다른 자격 증명 공급자처럼 SDK 및 도구 코어 라이브러리에 포함되지 않습니다. Amazon Cognito API에 액세스하려면 SDK 또는 도구용 빌드 또는 라이브러리에 Amazon Cognito 서비스 클라이언트를 포함시키십시오. AWS SDK와 함께 사용하는 방법은 Amazon Cognito 개발자 안내서의 [코드 예제](#)를 참조하십시오.

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

AWS 액세스 키

단기 보안 인증 정보를 사용합니다

연장된 세션 기간 옵션 사용에 [IAM 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 옵션과 [IAM Identity Center 인증](#)을 함께 사용하여 SDK 또는 도구를 구성하는 것이 좋습니다. 하지만 AWS 액세스 포털에서 사용할 수 있는 임시 보안 인증을 복사하여 사용할 수 있습니다. 보안 인증이 만료되면 새 보안 인증을 복사해야 합니다. 프로필에서 임시 보안 인증을 사용하거나 이를 시스템 속성 및 환경 변수의 값으로 사용할 수 있습니다.

AWS 액세스 포털에서 검색한 단기 보안 인증을 사용하여 보안 인증 파일을 설정합니다.

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>
```


IAM 사용자를 사용하여 코드를 실행하면 개발 환경의 SDK 또는 도구가 공유 파일의 장기 IAM 사용자 자격 증명을 사용하여 인증합니다. AWS credentials [IAM의 보안 모범 사례](#) 주제를 검토하고 가능한 한 빨리 IAM Identity Center 또는 기타 임시 보안 인증으로 전환하십시오.

보안 인증에 대한 중요 경고 및 지침

보안 인증에 대한 경고

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

보안 인증 정보를 안전하게 관리하기 위한 추가 지침

AWS 자격 증명을 안전하게 관리하는 방법에 대한 일반적인 설명은 [AWS 액세스 키 관리 모범 사례](#)를 참조하십시오. [AWS 일반 참조](#) 해당 설명과 더불어 다음 사항을 고려하십시오.

- Amazon Elastic Container Service(Amazon ECS) 작업에 [작업용 IAM 역할](#)을 사용하십시오.
- Amazon EC2 인스턴스에서 실행 중인 애플리케이션에 [IAM 역할](#)을 사용하십시오.

사전 요구 사항: 계정 생성 AWS

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

1. 계정을 생성합니다.

AWS 계정을 만들려면 [시작하기: 처음 AWS 사용하는 사용자이신가요?](#)를 참조하십시오. AWS Account Management 참조 안내서에서.

2. 관리 사용자를 생성합니다.

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

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

이러한 계정 중 어느 것도 개발을 AWS 하거나 에서 애플리케이션을 실행하는 데 적합하지 AWS않습니다. 모범 사례로서 이러한 작업에 적합한 사용자, 권한 집합 및 서비스 역할을 만들어야 합니다. 자세한 정보는 IAM 사용 설명서의 [최소 권한 적용](#)을 참조하십시오.

1단계: IAM 사용자 생성

- IAM 사용 설명서의 [IAM 사용자 생성\(콘솔\)](#) 절차에 따라 IAM 사용자를 생성합니다. IAM 사용자를 생성할 때:
 - 에 대한 사용자 액세스 제공을 선택하는 것이 좋습니다. AWS Management Console 이렇게 하면 AWS CloudTrail 진단 로그를 확인하거나 Amazon Simple Storage Service에 파일을 업로드하는 등 시각적 환경에서 실행 중인 코드와 AWS 서비스 관련된 내용을 볼 수 있어 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. IAM 콘솔의 탐색 창에서 사용자를 선택한 다음 이전에 생성한 사용자의 **User name**를 선택합니다.
2. 사용자 페이지에서 보안 보안 인증 페이지를 선택합니다. 그런 다음 액세스 키에서 액세스 키 생성을 선택합니다.
3. 액세스 키 생성 1단계에서 명령줄 인터페이스(CLI)또는 로컬 코드를 선택합니다. 두 옵션 모두 SDK와 함께 사용할 수 있는 동일한 유형의 키를 생성합니다. AWS CLI
4. 액세스 키 만들기 2단계에서 선택적 태그를 입력하고 다음을 선택합니다.
5. 액세스 키 생성 3단계에서 .csv 파일 다운로드를 선택하여 IAM 사용자의 액세스 키 및 보안 액세스 키와 함께 .csv 파일을 저장합니다. 나중에 이 정보가 필요합니다.

⚠ Warning

적절한 보안 조치를 사용하여 이러한 보안 인증을 안전하게 유지합니다.

6. 완료(Done)를 선택합니다.

3단계: credentials 파일 업데이트

1. 공유 AWS credentials 파일을 생성하거나 엽니다. 이 파일은 Linux 및 macOS 시스템의 경우 `~/.aws/credentials`이며, Windows의 경우 `%USERPROFILE%\aws\credentials`입니다. 자세한 내용은 [보안 인증 파일 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 보안 인증으로 전환하는 것이 좋습니다. 장기 보안 인증을 사용하지 않도록 전환한 후에는 공유 credentials 파일에서 해당 보안 인증을 삭제해야 합니다.

Amazon EC2 인스턴스의 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 인스턴스의 경우 IAM 역할을 생성한 뒤 Amazon EC2 인스턴스에 해당 역할에 대한 액세스 권한을 부여합니다. 자세한 내용은 Amazon EC2 사용 설명서의 [Amazon EC2의 IAM 역할](#) 또는 Amazon EC2 사용 설명서의 [Amazon EC2의 IAM 역할](#)을 참조하십시오.

IAM 역할 생성

Amazon S3에 대한 읽기 전용 액세스를 부여하는 IAM 역할을 생성합니다.

1. AWS Management Console [로그인하고 https://console.aws.amazon.com/iam/](https://console.aws.amazon.com/iam/) 에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 역할을 선택한 다음 역할 생성을 선택합니다.
3. 신뢰할 수 있는 엔터티 선택을 위해 신뢰할 수 있는 엔터티 유형 아래에서 AWS 서비스 서비스를 선택합니다.
4. 사용 사례에서 Amazon EC2를 선택한 후 다음을 선택합니다.
5. 권한 추가의 경우 정책 목록에서 Amazon S3 읽기 전용 액세스 확인란을 선택한 후 다음을 선택합니다.
6. 역할 이름을 입력한 다음 역할 생성을 선택합니다. Amazon EC2 인스턴스를 시작할 때 필요하므로 이 이름을 기억해 두십시오.

Amazon EC2 인스턴스 시작과 IAM 역할 지정

Amazon EC2 콘솔을 사용하여 IAM 역할로 Amazon EC2 인스턴스를 시작할 수 있습니다.

[Amazon EC2 사용 설명서 또는 Amazon EC2 사용 설명서의 지침을 따라 인스턴스를 시작하십시오.](#)

Review Instance Launch(인스턴스 시작 검토) 페이지가 표시되면 Edit instance details(인스턴스 세부 정보 편집)를 선택합니다. IAM 역할에서 이전에 생성한 IAM 역할을 선택합니다. 안내에 따라 절차를 완료합니다.

Note

보안 그룹을 생성하거나 기존 보안 그룹 및 키 페어를 사용하여 인스턴스에 접속해야 합니다.

이 IAM 및 Amazon EC2 설정에서 애플리케이션을 Amazon EC2 인스턴스로 배포하여 Amazon S3 서비스에 읽기 액세스를 할 수 있습니다.

EC2 인스턴스에 연결

EC2 인스턴스에 연결하면 샘플 애플리케이션을 전송한 다음 애플리케이션을 실행할 수 있습니다. 인스턴스를 시작하는 데 사용한 키 페어의 비공개 부분이 포함된 파일, 즉 PEM 파일이 필요합니다.

[Amazon EC2 사용 설명서 또는 Amazon EC2 사용 설명서의 연결 절차에 따라 이 작업을 수행할 수 있습니다.](#) 연결할 때는 개발 머신에서 인스턴스로 파일을 전송할 수 있는 방식으로 연결합니다.

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

EC2 인스턴스에서 샘플 애플리케이션 실행

1. 애플리케이션 파일을 로컬 드라이브에서 인스턴스로 복사합니다.

[인스턴스로 파일을 전송하는 방법에 대한 자세한 내용은 Amazon EC2 사용 설명서 또는 Amazon EC2 사용 설명서를 참조하십시오.](#)

2. 애플리케이션을 시작하고 개발 시스템에서 동일한 결과로 실행되는지 확인합니다.
3. (선택 사항) 애플리케이션이 IAM 역할에서 제공하는 보안 인증을 사용하는지 확인합니다.
 - a. AWS Management Console [로그인하고 https://console.aws.amazon.com/ec2/](https://console.aws.amazon.com/ec2/) 에서 [Amazon EC2 콘솔을 엽니다.](#)
 - b. 인스턴스를 선택하고 나서 작업, 인스턴스 설정, IAM 역할 연결/바꾸기를 통해 IAM 역할을 분리합니다.
 - c. 애플리케이션을 다시 실행하고 인증 오류가 반환되는지 확인합니다.

설정 참조

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

구성 설정에 대한 자세한 내용은 다음 섹션에서 확인할 수 있습니다.

- [AWS SDK 및 도구 표준화된 자격 증명 제공업체](#) - 여러 SDK에 표준화된 공통 보안 인증 공급자.
- [AWS SDK 및 도구 표준화된 기능](#) - 여러 SDK에 표준화된 공통 기능.

서비스 클라이언트 생성

프로그래밍 방식으로 AWS 서비스 액세스하기 위해 SDK는 각각에 대한 클라이언트 클래스/객체를 사용합니다. AWS 서비스 예를 들어, 애플리케이션이 Amazon EC2에 액세스해야 하는 경우, 애플리케이션은 Amazon EC2 클라이언트 객체를 생성하여 해당 서비스와 인터페이스 합니다. 그런 다음 서비스 클라이언트를 사용하여 요청을 AWS 서비스에 보내면 됩니다. 대부분의 SDK에서 서비스 클라이언트 개체는 변경할 수 없으므로 요청을 보내는 각 서비스와 다른 구성을 사용하여 동일한 서비스에 요청을 보낼 수 있도록 새 클라이언트를 만들어야 합니다.

설정의 우선 순위

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

1. 코드나 서비스 클라이언트 자체에 설정된 모든 명시적 설정은 다른 모든 설정보다 우선합니다.
 - 일부 설정은 작업별로 설정할 수 있으며 간접 호출하는 각 작업에 대해 필요에 따라 변경할 수 있습니다. AWS CLI OR의 AWS Tools for PowerShell 경우 이는 명령줄에 입력하는 작업별 매개 변수의 형태를 취합니다. SDK의 경우 명시적 할당은 AWS 서비스 클라이언트 또는 구성 객체를 인스턴스화할 때 설정하거나 개별 API를 호출할 때 설정하는 파라미터의 형태를 취할 수 있습니다.
2. 자바/Kotlin 전용: 설정의 JVM 시스템 속성을 확인합니다. 설정이 되면 클라이언트 구성에 해당 값이 사용됩니다.
3. 환경 변수를 확인합니다. 설정이 되면 클라이언트 구성에 해당 값이 사용됩니다.

4. SDK는 공유 파일의 설정을 확인합니다. `credentials` 설정되어 있으면 클라이언트가 이를 사용합니다.
5. 설정의 공유 `config` 파일입니다. 설정이 있으면 SDK는 해당 설정을 사용합니다.
 - `AWS_PROFILE` 환경 변수 또는 `aws.profile` JVM 시스템 속성을 사용하여 SDK가 로드할 프로필을 지정할 수 있습니다.
6. SDK 소스 코드 자체에서 제공하는 모든 기본값이 마지막에 사용됩니다.

Note

일부 SDK 및 도구는 순서가 다를 수 있습니다. 또한 일부 SDK 및 도구는 다른 파라미터 저장과 검색 방법을 지원합니다. 예를 들어는 [SDK Store](#)라는 추가 소스를 AWS SDK for .NET 지원합니다. 특정 SDK 또는 도구 제공자에 대한 자세한 내용은 사용 중인 특정 SDK 또는 도구의 설명서를 참조합니다.

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

Config 파일 설정 목록

다음 표에 나열된 설정을 공유 AWS `config` 파일에 할당할 수 있습니다. 이는 전 세계에 적용되며 모든 AWS 서비스에 영향을 미칩니다. SDK 및 도구는 고유한 설정과 환경 변수를 지원할 수도 있습니다. 개별 SDK 또는 도구에서만 지원되는 설정 및 환경 변수를 보려면 해당 SDK 또는 도구 가이드를 참조하십시오.

설정 이름	Details
<code>api_versions</code>	일반 구성 설정
<code>aws_access_key_id</code>	AWS 액세스 키

설정 이름	Details
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
duration_seconds	역할 보안 인증 제공자 수입
ec2_metadata_service_endpoint	IMDS 보안 인증 제공자
ec2_metadata_service_endpoint_mode	IMDS 보안 인증 제공자
ec2_metadata_v1_disabled	IMDS 보안 인증 제공자

설정 이름	Details
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
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

설정 이름	Details
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 제공자 수입
sso_account_id	IAM 아이덴티티 센터 보안 인증 공급자
sso_region	IAM 아이덴티티 센터 보안 인증 공급자
sso_registration_scopes	IAM 아이덴티티 센터 보안 인증 공급자
sso_role_name	IAM 아이덴티티 센터 보안 인증 공급자
sso_start_url	IAM 아이덴티티 센터 보안 인증 공급자
sts_regional_endpoints	AWS STS 지역화된 엔드포인트
use_dualstack_endpoint	이중 스택 엔드포인트 및 FIPS 엔드포인트
use_fips_endpoint	이중 스택 엔드포인트 및 FIPS 엔드포인트

설정 이름	Details
web_ident ity_token_file	역할 보안 인증 제공자 수입

Credentials 파일 설정 목록

다음 표에 나열된 설정을 공유 파일에 할당할 수 있습니다. AWS credentials 이는 전 세계에 적용되며 모든 AWS 서비스에 영향을 미칩니다. SDK 및 도구는 고유한 설정과 환경 변수를 지원할 수도 있습니다. 개별 SDK 또는 도구에서만 지원되는 설정 및 환경 변수를 보려면 해당 SDK 또는 도구 가이드를 참조하십시오.

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 SDK에서 지원되는 환경 변수가 아래에 나열되어 있습니다. 이는 전 세계에 적용되며 모든 AWS 서비스에 영향을 미칩니다. SDK 및 도구는 고유한 설정 및 환경 변수를 지원할 수도 있습니다. 개별 SDK 또는 도구에서만 지원되는 설정 및 환경 변수를 보려면 해당 SDK 또는 도구 가이드를 참조하십시오.

설정 이름	Details
AWS_ACCESS_KEY_ID	AWS 액세스 키
AWS_CA_BUNDLE	일반 구성 설정

설정 이름	Details
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_RELATIVE_URI	컨테이너 보안 인증 제공업체
AWS_DEFAULTS_MODE	스마트 구성 기본값
AWS_DISABLE_REQUEST_COMPRESSION	요청 압축
AWS_EC2_METADATA_DISABLED	IMDS 보안 인증 제공자

설정 이름	Details
AWS_EC2_METADATA_SERVICE_ENDPOINT	IMDS 보안 인증 제공자
AWS_EC2_METADATA_SERVICE_ENDPOINT_MODE	IMDS 보안 인증 제공자
AWS_EC2_METADATA_V1_DISABLED	IMDS 보안 인증 제공자
AWS_ENABLE_ENDPOINT_DISCOVERY	엔드포인트 검색
AWS_ENDPOINT_URL	서비스별 엔드포인트
AWS_ENDPOINT_URL_<SERVICE>	서비스별 엔드포인트
AWS_IAM_ROLE_ARN	역할 보안 인증 제공자 수입
AWS_IAM_ROLE_SESSION_NAME	역할 보안 인증 제공자 수입
AWS_IGNORE_CONFIGURED_ENDPOINT_URLS	서비스별 엔드포인트

설정 이름	Details
AWS_MAX_A TTEMPTS	재시도 동작
AWS_METAD ATA_SERVI CE_NUM_AT TEMPTS	Amazon EC2 인스턴스 메타데이터
AWS_METAD ATA_SERVI CE_TIMEOUT	Amazon EC2 인스턴스 메타데이터
AWS_PROFILE	공유 config 및 파일 credentials
AWS_REGION	AWS 리전
AWS_REQUE ST_MIN_CO MPRESSION _SIZE_BYTES	요청 압축
AWS_RETRY_MODE	재시도 동작
AWS_S3_DI SABLE_MUL TIREGION_ ACCESS_POINTS	Amazon S3 다중 리전 액세스 포인트
AWS_S3_US E_ARN_REGION	Amazon S3 액세스 포인트
AWS_SDK_U A_APP_ID	애플리케이션 ID
AWS_SECRE T_ACCESS_KEY	AWS 액세스 키

설정 이름	Details
AWS_SESSION_TOKEN	AWS 액세스 키
AWS_SHARED_CREDENTIALS_FILE	공유 config 및 credentials 파일 위치
AWS_STS_REGIONAL_ENDPOINTS	AWS STS 지역화된 엔드포인트
AWS_USE_DEFAULTS_ENDPOINT	이중 스택 엔드포인트 및 FIPS 엔드포인트
AWS_USE_FIPS_ENDPOINT	이중 스택 엔드포인트 및 FIPS 엔드포인트
AWS_WEB_IDENTITY_TOKEN_FILE	역할 보안 인증 제공자 수입

JVM 시스템 속성 목록

AWS SDK for Java 및 AWS SDK for Kotlin (JVM 대상) 에 다음과 같은 JVM 시스템 속성을 사용할 수 있습니다. JVM 시스템 속성을 설정하는 방법에 [the section called “JVM 시스템 속성을 설정하는 방법”](#) 대한 지침은 을 참조하십시오.

설정 이름	Details
<code>aws.accessKeyId</code>	AWS 액세스 키
<code>aws.configFile</code>	공유 config 및 credentials 파일 위치
<code>aws.defaultsMode</code>	스마트 구성 기본값

설정 이름	Details
<code>aws.disableEc2MetadataV1</code>	IMDS 보안 인증 제공자
<code>aws.disableRequestCompression</code>	요청 압축
<code>aws.ec2MetadataServiceEndpoint</code>	IMDS 보안 인증 제공자
<code>aws.ec2MetadataServiceEndpointMode</code>	IMDS 보안 인증 제공자
<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

설정 이름	Details
<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
<code>aws.s3UseArnRegion</code>	Amazon S3 액세스 포인트
<code>aws.secretAccessKey</code>	AWS 액세스 키
<code>aws.sessionToken</code>	AWS 액세스 키
<code>aws.sharedCredentialsFile</code>	공유 config 및 credentials 파일 위치
<code>aws.useDualstackEndpoint</code>	이중 스택 엔드포인트 및 FIPS 엔드포인트
<code>aws.useFipsEndpoint</code>	이중 스택 엔드포인트 및 FIPS 엔드포인트

설정 이름	Details
aws.userAgentAppId	애플리케이션 ID
aws.webIdentityTokenFile	역할 보안 인증 공급자 위임

AWS SDK 및 도구 표준화된 자격 증명 제공업체

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

Important

모든 SDK가 모든 제공자를 지원하는 것은 아니며 공급자 내의 모든 측면을 지원하는 것은 아닙니다.

주제

- [보안 인증 공급자 체인](#)
- [AWS 액세스 키](#)
- [역할 보안 인증 제공자 수입](#)
- [컨테이너 보안 인증 제공업체](#)
- [IAM 아이덴티티 센터 보안 인증 공급자](#)
- [IMDS 보안 인증 제공자](#)
- [프로세스 보안 인증 제공자](#)

보안 인증 공급자 체인

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

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

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

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

AWS 액세스 키

⚠ Warning

보안 위협을 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.

AWS IAM 사용자의 액세스 키를 자격 증명으로 사용할 수 있습니다 AWS . AWS SDK는 이러한 AWS 자격 증명을 사용하여 API 요청에 자동으로 서명하므로 워크로드가 AWS 리소스와 데이터에 안전하고 편리하게 액세스할 수 있습니다. AWS보안 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.

i Note

이러한 임시 자격 증명을 새로 고칠 AWS 수 없는 경우 AWS 워크로드에 영향을 미치지 않도록 자격 증명의 유효 기간을 연장할 수 있습니다.

공유 AWS credentials 파일은 애플리케이션 소스 디렉토리 외부에 안전하게 보관되고 공유 파일의 SDK 관련 설정과 분리되어 있으므로 자격 증명 정보를 저장하는 데 사용하는 것이 좋습니다. config

AWS 자격 증명 및 액세스 키 사용에 대한 자세한 내용은 IAM 사용 [AWS 설명서의 보안 자격 증명 및 IAM 사용자의 액세스 키 관리](#)를 참조하십시오.

다음을 사용하여 이 기능을 구성하십시오.

aws_access_key_id- 공유 파일 설정 AWS config, **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

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

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

사용자를 인증하기 위해 자격 증명의 일부로 사용되는 AWS 비밀 키를 지정합니다.

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

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

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

config 또는 credentials 파일에 이러한 필수 값을 설정하는 예:

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

SDK와의 AWS 호환성

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

SDK	지	참고 또는 추가 정보
AWS CLI v2	예	
SDK for C++	예	공유 config 파일은 지원되지 않습니다.
SDK for Go V2 (1.x)	예	
SDK for Go 1.x (V1)	예	공유 config 파일 설정을 사용하려면 구성 파일에서 로드를 켜야 합니다. 세션 을 참조하십시오.
SDK for Java 2.x	예	
SDK for Java 1.x	예	
3.x용 SDK JavaScript	예	
2.x용 SDK JavaScript	예	
SDK for Kotlin	예	
SDK for .NET 3.x	예	환경 변수는 지원되지 않습니다.
SDK for PHP 3.x	예	
SDK for Python (Boto3)	예	
SDK for Ruby 3.x	예	
SDK for Rust	예	
다음에 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 수임하도록 SDK 또는 도구를 설정하려면 먼저 수임할 특정 역할을 만들거나 식별해야 합니다. IAM 역할은 Amazon 리소스 이름(ARN)역할로 고유하게 식별됩니다. 역할은 다른 엔티티와 신뢰 관계를 구축합니다. 역할을 사용하는 신뢰할 수 있는 주체는 다른 AWS 계정 웹 ID 제공업체 AWS 서비스, OIDC 또는 SAML 페더레이션일 수 있습니다.

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

이러한 설정 사용을 시작하는 방법에 대한 지침은 이 안내서의 [역할 수임](#)를 참조하세요.

역할 보안 인증 제공자 수임 설정

다음을 사용하여 이 기능을 구성하십시오.

credential_source- 공유 파일 설정 AWS config

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

기본값: 없음

유효값:

- 환경 – SDK나 도구가 [AWS_ACCESS_KEY_ID 및 AWS_SECRET_ACCESS_KEY](#) 환경 변수에서 소스 보안 인증을 검색하도록 지정합니다.
- EC2 InstanceMetadata - [EC2 인스턴스 프로필에 연결된 IAM 역할을 사용하여 원본 자격 증명을 가져오도록](#) SDK 또는 도구를 지정합니다.
- EcsContainer— [ECS 컨테이너에 연결된 IAM 역할을 사용하여 소스 자격 증명을 가져오도록](#) SDK 또는 도구를 지정합니다.

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

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

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

duration_seconds- 공유 파일 설정 AWS config

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 설명서의 [역할의 최대 세션 기간 설정 보기](#)를 참조하십시오.

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 [제3자에게 AWS 리소스에 대한 액세스 권한을 부여할 때 외부 ID를 사용하는 방법](#)을 참조하십시오.

config 파일에서 이를 설정하는 예:

```
external_id = unique_value_assigned_by_3rd_party
```

mfa_serial- 공유 파일 설정 AWS config

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

해당 역할에 대한 신뢰 정책에 MFA 인증을 필요로 하는 조건이 포함된 역할을 수입하는 경우 필요합니다.

기본값: 없음.

유효한 값: 이 값은 하드웨어 디바이스용 일련 번호(예: GAHT12345678)또는 가상 MFA 디바이스용 Amazon 리소스 이름(ARN)(예:)일 수 있습니다. 자세한 정보는 IAM 사용 설명서의 [MFA 보호 API 액세스 구성](#)을 참조하십시오.

config 파일에서 이를 설정하는 예:

```
mfa_serial = arn:aws:iam::123456789012:mfa/my-user-name
```

role_arn- 공유 AWS config 파일 설정, **AWS_IAM_ROLE_ARN** - 환경 변수, **aws.roleArn**- JVM 시스템 속성: 자바/코틀린 전용

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

기본값: 없음.

유효한 값: 이 값은 다음과 같은 형식의 IAM 역할의 ARN이어야 합니다.

```
arn:aws:iam::account-id:role/role-name
```

또한 다음 설정 중 하나를 지정해야 합니다.

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

role_session_name AWS config- 공유 파일 설정, **AWS_IAM_ROLE_SESSION_NAME** - 환경 변수, **aws.roleSessionName**- JVM 시스템 속성: 자바/코틀린 전용

역할 세션에 연결할 이름을 지정합니다. 이 이름은 이 세션과 연결된 항목에 대한 AWS CloudTrail 로그에 나타나며, 감사할 때 유용할 수 있습니다.

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

유효한 값: AWS CLI 또는 AWS API가 사용자 대신 작업 (또는 작업과 같은 작업) 을 AssumeRole 호출할 때 RoleSessionName 파라미터에 제공됩니다. AssumeRoleWithWebIdentity 값은 쿼리할 수 있는 위임 역할 사용자 Amazon Resource Name (ARN) 의 CloudTrail 일부가 되며 이 프로파일에서 호출한 작업에 대한 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
```


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

기본값: 없음.

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

AWS SDK와의 호환성

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

SDK	지	참고 또는 추가 정보
AWS CLI v2	예	
SDK for C++	부	credential_source 이 지원되지 않음. duration_seconds 이 지원되지 않음. mfa_serial 이 지원되지 않음.
SDK for Go V2 (1.x)	예	
SDK for Go 1.x (V1)	예	공유 config 파일 설정을 사용하려면 구성 파일에서 로드를 켜야 합니다. 세션 을 참조하십시오.
SDK for Java 2.x	부	mfa_serial 이 지원되지 않음. AWS_ROLE_ARN 대신 AWS_IAM_ROLE_ARN 사용하십시오. AWS_ROLE_SESSION_NAME 대신 사용하세요AWS_IAM_ROLE_SESSION_NAME .
SDK for Java 1.x	부	mfa_serial 이 지원되지 않음. JVM 시스템 속성은 지원되지 않습니다.
3.x용 SDK JavaScript	예	

SDK	지 운	참고 또는 추가 정보
2.x용 SDK JavaScript	부 분	credential_source 이 지원되지 않음.
SDK for Kotlin	예	AWS_ROLE_ARN 대신 사용하세요. AWS_IAM_ROLE_ARN AWS_ROLE_SESSION_NAME 대신 사용하세요AWS_IAM_R OLE_SESSION_NAME .
SDK for .NET 3.x	예	
SDK for PHP 3.x	예	
SDK for Python (Boto3)	예	
SDK for Ruby 3.x	예	
SDK for Rust	예	
를 위한 도구 PowerShell	예	

컨테이너 보안 인증 제공업체

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

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

Amazon EKS를 사용하는 경우 보안 인증 격리, 최소 권한, 감사 가능성, 독립적인 운영, 재사용성 및 확장성을 개선하기 위해 Amazon EKS Pod Identity를 사용하는 것이 좋습니다. 포드와 IAM 역할은 모두 Kubernetes 서비스 계정과 연결되어 애플리케이션에 대한 보안 인증을 관리합니다. Amazon EKS 포드 ID에 대해 자세히 알아보려면 Amazon EKS 사용 설명서의 [Amazon EKS Pod](#)

[Identities](#)를 참조하세요. 구성된 경우 Amazon EKS는 SDK 및 도구가 보안 인증을 얻기 위해 사용하는 `AWS_CONTAINER_CREDENTIALS_FULL_URI` 및 `AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE` 환경 변수를 설정합니다. 설정 정보는 Amazon EKS 사용 설명서의 [Amazon EKS 포트 ID 에이전트 설정](#)을 참조하십시오. 또는 블로그 웹 사이트의 Amazon EKS [포트 아이덴티티는 Amazon EKS 클러스터의 애플리케이션에 대한 IAM 권한을 간소화합니다](#). AWS

다음을 사용하여 이 기능을 구성하십시오.

`AWS_CONTAINER_CREDENTIALS_FULL_URI` - 환경 변수

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

기본값: 없음.

유효한 값: 유효한 URI.

참고: 이 설정은 `AWS_CONTAINER_CREDENTIALS_RELATIVE_URI` 설정의 대안이며 `AWS_CONTAINER_CREDENTIALS_RELATIVE_URI`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

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

기본값: 없음.

유효한 값: 유효한 상대 URI.

명령 행을 통한 환경 변수 설정의 Linux/macOS 예제:

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

AWS_CONTAINER_AUTHORIZATION_TOKEN - 환경 변수

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

기본값: 없음.

유효한 값: 문자열.

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

AWS SDK와의 호환성

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

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

SDK	지 운	참고 또는 추가 정보
SDK for Go V2 (1.x)	예	
SDK for Go 1.x (V1)	예	
SDK for Java 2.x	예	
SDK for Java 1.x	부 분	Amazon EKS Pod Identity 및 AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE 는 지원되지 않습니다.
3.x용 SDK JavaScript	예	
2.x용 SDK JavaScript	예	
SDK for Kotlin	예	
SDK for .NET 3.x	예	
SDK for PHP 3.x	예	
SDK for Python (Boto3)	예	
SDK for Ruby 3.x	예	
SDK for Rust	예	
다음을 위한 도구 PowerShell	예	

IAM 아이덴티티 센터 보안 인증 공급자

이 인증 메커니즘은 AWS IAM Identity Center 코드에 대한 싱글 사인온 (SSO) 액세스를 얻는 데 AWS 서비스 사용됩니다.

Note

AWS SDK API 설명서에서는 IAM ID 센터 자격 증명 공급자를 SSO 자격 증명 공급자라고 합니다.

IAM Identity Center를 활성화한 후에는 공유 파일의 설정에 대한 프로필을 정의합니다. AWS config 이 프로파일은 IAM ID 센터 액세스 포털에 연결하는 데 사용됩니다. 사용자가 IAM Identity Center에서 성공적으로 인증하면 포털은 해당 사용자와 관련된 IAM 역할에 대한 단기 보안 인증을 반환합니다. SDK가 구성에서 임시 자격 증명을 가져와 AWS 서비스 요청에 사용하는 방법을 알아보려면 [IAM Identity Center 인증 이해하기](#)를 참조하십시오.

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

- SSO 토큰 공급자 구성 (권장) — 세션 기간 연장.
- 새로 고칠 수 없는 레거시 구성 - 고정된 8시간 세션을 사용합니다.

두 구성 모두 세션이 만료되면 다시 로그인해야 합니다.

사용자 지정 세션 기간을 설정하려면 SSO 토큰 공급자 구성을 사용해야 합니다.

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

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

필수 조건

먼저 IAM Identity Center를 활성화해야 합니다. IAM Identity Center에 대한 자세한 내용은 AWS IAM Identity Center 사용 설명서의 [시작하기](#)를 참조하십시오.

또는 이 안내서의 [IAM Identity Center 인증](#) 지침을 따르십시오. 이 지침은 IAM Identity Center를 활성화하는 것부터 여기에 나오는 필수 공유 config 파일 구성을 완료하는 것까지 완전한 지침을 제공합니다.

SSO 토큰 공급자 구성

Note

를 사용하여 이 구성을 자동으로 AWS CLI 생성하려면 [의 aws configure sso 마법사를 사용하여 프로필 구성을 참조하십시오](#) AWS CLI.

SSO 토큰 공급자 구성을 사용하면 AWS SDK 또는 도구가 연장된 세션 기간까지 세션을 자동으로 새로 고칩니다. 세션 기간 및 최대 기간에 대한 자세한 내용은 [사용 설명서의 AWS 액세스 포털 및 IAM Identity Center 통합 애플리케이션의 세션 기간 구성](#)을 참조하십시오. AWS IAM Identity Center

이 config 파일의 sso-session 섹션은 SSO 액세스 토큰을 획득하기 위한 구성 변수를 그룹화하는데 사용되며, 이 구성 변수를 사용하여 자격 증명을 획득할 수 있습니다. AWS config 파일 내 섹션 형식 지정에 대한 자세한 내용은 [구성 파일 형식](#) 섹션을 참조하십시오.

sso-session 섹션을 정의하고 프로파일에 연결합니다. sso-session 섹션 내에서 sso_region 및 sso_start_url을 설정해야 합니다. 일반적으로 SDK가 자격 증명을 요청할 수 있도록 profile 섹션에서 sso_role_name 설정해야 합니다. sso_account_id AWS

Note

SDK 및 도구가 이 구성을 사용하여 보안 인증을 사용하고 새로 고치는 방법에 대한 자세한 내용은 [IAM Identity Center 인증 이해하기](#) 섹션을 참조하십시오.

다음 예제는 IAM Identity Center 보안 인증을 요청하도록 SDK를 구성합니다. 또한 자동 토큰 새로 고침도 지원합니다.

```
[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
```

여러 프로파일에서 sso-session 구성을 재사용할 수 있습니다.

```
[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
```

sso_account_id 및 sso_role_name은 SSO 토큰 구성의 모든 시나리오에 필수적이지는 않습니다. 애플리케이션에서 베어러 인증을 AWS 서비스 지원하는 인증만 사용하는 경우 기존 AWS 자격 증명은 필요하지 않습니다. 보유자 인증은 보유자 토큰이라는 보안 토큰을 사용하는 HTTP 인증 체계입니다. 이 시나리오에서는 sso_account_id 및 sso_role_name은 필수가 아닙니다. 베어러 토큰 인증을 지원하는지 AWS 서비스 확인하려면 개별 가이드를 참조하십시오.

등록 범위는 sso-session의 일부로 구성됩니다. 범위는 애플리케이션의 사용자 계정 액세스를 제한하는 OAuth 2.0의 메커니즘입니다. 애플리케이션은 하나 이상의 범위를 요청할 수 있으며 애플리케이션에 발급되는 액세스 토큰은 부여된 범위로 제한됩니다. 이러한 범위는 등록된 OIDC 클라이언트에 대해 인증받기 위해 요청된 권한과 클라이언트가 검색한 액세스 토큰을 정의합니다. 지원되는 액세스 범위 옵션은 AWS IAM Identity Center 사용 설명서의 [액세스 범위](#)를 참조하십시오. 다음 예제는 계정 및 역할 목록에 대한 액세스 권한을 제공하도록 sso_registration_scopes을(를)설정합니다.

```
[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 프로파일에서 다음 설정을 지정해야 합니다.

- sso_start_url
- sso_region
- sso_account_id
- sso_role_name

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

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

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

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

IAM Identity Center 보안 인증 공급자 설정

다음을 사용하여 이 기능을 구성하십시오.

`sso_start_url`- 공유 AWS `config` 파일 설정

조직의 IAM Identity Center 액세스 포털을 가리키는 URL입니다. IAM Identity Center 액세스 포털에 대한 자세한 내용은 [사용 설명서의 AWS 액세스 포털 사용](#)을 참조하십시오. AWS IAM Identity Center

이 값을 찾으려면 [IAM Identity Center 콘솔](#)을 열고 대시보드를 확인한 다음 AWS 액세스 포털 URL을 찾으십시오.

`sso_region`- 공유 AWS `config` 파일 설정

여기에는 IAM ID 센터 포털 호스트, 즉 IAM ID 센터를 활성화하기 전에 선택한 지역이 포함됩니다. AWS 리전 이는 기본 AWS 지역과는 독립적이며 다를 수 있습니다.

AWS 리전 및 해당 코드의 전체 목록은 [의 지역별 엔드포인트를](#) 참조하십시오. Amazon Web Services 일반 참조 이 값을 찾으려면 [IAM Identity Center 콘솔](#)을 열고 대시보드를 확인한 다음 리전을 찾으십시오.

`sso_account_id`- 공유 AWS `config` 파일 설정

인증에 사용하기 위해 AWS Organizations 서비스를 통해 추가된 숫자 ID입니다. AWS 계정

사용 가능한 계정 목록을 보려면 [IAM Identity Center 콘솔](#)로 이동하여 AWS 계정 페이지를 여십시오. AWS IAM Identity Center 포털 API 참조의 [ListAccounts](#) API 방법을 사용하여 사용 가능한 계정 목록을 볼 수도 있습니다. 예를 들어 [리스트](#) 계정 AWS CLI 메서드를 호출할 수 있습니다.

sso_role_name- 공유 파일 설정 AWS config

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

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

사용 가능한 권한 집합 목록을 보려면 [IAM Identity Center 콘솔로](#) 이동하여 AWS 계정 페이지를 여십시오. AWS 계정 표에 나열된 올바른 권한 집합 이름을 선택합니다. AWS 계정 AWS IAM Identity Center 포털 API 참조의 [ListAccountRoles](#) API 방법을 사용하여 사용 가능한 권한 집합 목록을 볼 수도 있습니다. 예를 들어 AWS CLI 메서드를 호출할 수 [list-account-roles](#) 있습니다.

sso_registration_scopes- 공유 AWS config 파일 설정

sso-session에 대해 인증될 토큰으로 구분된 유효한 범위 문자열 목록입니다. 범위는 IAM Identity Center 보유자 토큰 인증 엔드포인트에 대한 액세스를 승인합니다. IAM Identity Center 서비스에서 새로 고침 토큰을 다시 받으려면 sso:account:access의 최소 범위를 부여해야 합니다. 지원되는 액세스 범위 문자열은 AWS IAM Identity Center 사용 설명서의 [액세스 범위](#)를 참조하십시오. 이 설정은 새로 고칠 수 없는 레거시 구성에는 적용되지 않습니다. 레거시 구성을 사용하여 발급된 토큰은 암시적으로 범위 sso:account:access(으)로 제한됩니다.

AWS SDK와의 호환성

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

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

SDK	지 운	참고 또는 추가 정보
SDK for Java 2.x	예	구성 값은 credentials 파일에서도 지원됩니다.
SDK for Java 1.x	아 니 요	
3.x용 SDK JavaScript	예	
2.x용 SDK JavaScript	예	
SDK for Kotlin	예	
SDK for .NET 3.x	예	
SDK for PHP 3.x	예	
SDK for Python (Boto3)	예	
SDK for Ruby 3.x	예	
SDK for Rust	부 분	새로 고칠 수 없는 레거시 구성에만 해당.
다음에 위한 도구 PowerShell	예	

IMDS 보안 인증 제공자

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

다음을 사용하여 이 기능을 구성하십시오.

AWS_EC2_METADATA_DISABLED - 환경 변수

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

기본값: `false`.

유효값:

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

ec2_metadata_v1_disabled- 공유 파일 설정 **AWS config**,

AWS_EC2_METADATA_V1_DISABLED - 환경 변수, **aws.disableEc2MetadataV1**- JVM 시스템 속성: 자바/코틀린 전용

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

Note

새 SDK는 IMDSv1을 지원하지 않으므로 이 설정을 지원하지 않습니다. 자세한 내용은 테이블 [SDK와의 호환성 AWS](#)를 참조하세요.

기본값: `false`

유효값:

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

ec2_metadata_service_endpoint **AWS config**- 공유 파일 설정,

AWS_EC2_METADATA_SERVICE_ENDPOINT - 환경 변수, **aws.ec2MetadataServiceEndpoint**- JVM 시스템 속성: 자바/코틀린 전용

IMDS 엔드포인트.

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` AWS config- 공유 파일 설정, `AWS_EC2_METADATA_SERVICE_ENDPOINT_MODE` - 환경 변수, `aws.ec2MetadataServiceEndpointMode`- JVM 시스템 속성: 자바/코틀린 전용

IMDS의 엔드포인트 모드.

기본값: IPv4.

유효한 값: IPv4, IPv6.

Note

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

IMDS 보안 인증 보안

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

Note

AWS 유효한 자격 증명으로 구성된 SDK 클라이언트는 이러한 설정과 상관없이 IMDS를 사용하여 자격 증명을 검색하지 않습니다.

Amazon EC2 IMDS 보안 인증 비활성화

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

Linux 및 macOS

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

```
$ export AWS_EC2_METADATA_DISABLED=true
```

여러 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.

SDK와의 호환성 AWS

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

SDK	지참고 또는 추가 정보 운
AWS CLI v2	예
SDK for C++	예

SDK	지 운	참고 또는 추가 정보
SDK for Go V2 (1.x)	예	
SDK for Go 1.x (V1)	예	공유 config 파일 설정을 사용하려면 구성 파일에서 로드를 켜야 합니다. 세션 을 참조하십시오.
SDK for Java 2.x	예	
SDK for Java 1.x	부 분	JVM 시스템 속성: <code>com.amazonaws.sdk.disableEc2MetadataV1</code> 대신 <code>aws.ec2MetadataServiceEndpoint</code> 사용하며 <code>aws.disableEc2MetadataV1</code> 지원되지 <code>aws.ec2MetadataServiceEndpointMode</code> 않습니다.
3.x용 SDK JavaScript	예	
2.x용 SDK JavaScript	예	
SDK for Kotlin	예	IMDSv1 폴백을 사용하지 않습니다.
SDK for .NET 3.x	예	
SDK for PHP 3.x	예	
SDK for Python (Boto3)	예	
SDK for Ruby 3.x	예	
SDK for Rust	예	IMDSv1 폴백을 사용하지 않습니다.
다음에 위한 도구 PowerShell	예	를 사용하여 코드에서 IMDSv1 폴백을 명시적으로 비활성화할 수 있습니다. <code>[Amazon.Util.EC2InstanceMetadata]::EC2MetadataV1Disabled = \$true</code>

프로세스 보안 인증 제공자

SDK는 사용자 지정 사용 사례에 맞게 보안 인증 공급자 체인을 확장할 수 있는 방법을 제공합니다.

IAM Roles Anywhere는 외부에서 실행되는 워크로드 또는 프로세스에 대한 임시 자격 증명을 얻을 수 있는 방법을 제공합니다. AWS이 용도로 `credential_process`를 구성하려면 [IAM Roles Anywhere](#)을(를)참조하십시오.

⚠ Warning

다음은 외부 프로세스에서 보안 인증을 소싱하는 방법을 설명합니다. 이는 잠재적으로 위험할 수 있으므로 주의해서 진행하십시오. 가능하면 다른 보안 인증 공급자를 이용하는 것이 좋습니다. 이 옵션을 사용하는 경우 운영 체제의 보안 모범 사례를 사용하여 가능하면 `config` 파일을 잠그도록 해야 합니다. 사용자 지정 보안 인증 도구가 `StdErr`에 어떤 암호 정보도 기록하지 않아야 합니다. SDK 및 AWS CLI가 그러한 정보를 캡처 및 기록할 수 있으므로 승인되지 않은 사용자에게 노출될 위험이 있기 때문입니다.

다음을 사용하여 이 기능을 구성하십시오.

`credential_process`- 공유 AWS `config` 파일 설정

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

📌 Note

프로세스 보안 인증 공급자는 [보안 인증 공급자 체인](#)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 사용을 지정하십시오. 서비스 클라이언트를 생성할 때 보안 인증 소스를 직접 지정할 수 있습니다.

보안 인증 프로그램 경로 지정

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

- 경로와 파일 이름은 A~Z, a~z, 0~9, 하이픈(-), 밑줄(_), 마침표(.), 슬래시(/), 백슬래시(\) 및 공백 등의 문자는 다음과 같습니다.
- 경로 또는 파일 이름에 공백이 있으면 전체 경로와 파일 이름을 큰 따옴표(" ")로 묶습니다.
- 파라미터 이름이나 파라미터 값에 공백이 있으면 해당 요소를 큰 따옴표(" ")로 묶습니다. 전체 페어가 아니라 이름 또는 값만 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.

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"
```

보안 인증 프로그램에서 유효한 출력

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

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


Note

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

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

Note

SDK는 `assume-role` 보안 인증을 받는 방법으로 외부 프로세스 보안 인증을 캐싱하지 않습니다. 캐싱이 필요한 경우에는 외부 프로세스에서 이를 실행해야 합니다.

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

AWS SDK와의 호환성

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

SDK	지 운	참고 또는 추가 정보
AWS CLI v2	예	
SDK for C++	예	
SDK for Go V2 (1.x)	예	
SDK for Go 1.x (V1)	예	공유 config 파일 설정을 사용하려면 구성 파일에서 로드를 켜야 합니다. 세션 을 참조하십시오.
SDK for Java 2.x	예	

SDK	지 운	참고 또는 추가 정보
SDK for Java 1.x	예	
3.x용 SDK JavaScript	예	
2.x용 SDK JavaScript	예	
SDK for Kotlin	예	
SDK for .NET 3.x	예	
SDK for PHP 3.x	예	
SDK for Python (Boto3)	예	
SDK for Ruby 3.x	예	
SDK for Rust	예	
다음에 위한 도구 PowerShell	예	

AWS SDK 및 도구 표준화된 기능

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

Important

모든 SDK가 모든 기능을 지원하거나 기능 내의 모든 측면을 지원하는 것은 아닙니다.

주제

- [애플리케이션 ID](#)
- [Amazon EC2 인스턴스 메타데이터](#)
- [Amazon S3 액세스 포인트](#)

- [Amazon S3 다중 리전 액세스 포인트](#)
- [AWS 리전](#)
- [AWS STS 지역화된 엔드포인트](#)
- [이중 스택 엔드포인트 및 FIPS 엔드포인트](#)
- [엔드포인트 검색](#)
- [일반 구성 설정](#)
- [IMDS 클라이언트](#)
- [재시도 동작](#)
- [요청 압축](#)
- [서비스별 엔드포인트](#)
- [스마트 구성 기본값](#)

애플리케이션 ID

여러 고객 애플리케이션에서 단일 번호로 전화를 걸 AWS 계정 수 AWS 서비스 있습니다. 애플리케이션 ID를 통해 고객은 어떤 소스 애플리케이션이 a를 사용하여 일련의 호출을 했는지 식별할 수 AWS 계정 있습니다. AWS SDK와 서비스는 이 값을 고객 커뮤니케이션에 다시 표시하는 것 외에는 사용하거나 해석하지 않습니다. 예를 들어 이 값을 운영 이메일에 포함하거나 알림과 관련된 애플리케이션을 고유하게 식별하는 데 사용할 수 있습니다. AWS Health Dashboard

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 내에서 어떤 애플리케이션을 호출하는지 식별하기 위해 애플리케이션에 할당하는 고유한 문자열입니다. AWS 계정 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.

AWS SDK와의 호환성

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

SDK	지 운	참고 또는 추가 정보
AWS CLI v2	아 니 요	
SDK for C++	예	공유 config 파일은 지원되지 않습니다.
SDK for Go V2 (1.x)	예	
SDK for Go 1.x (V1)	아 니 요	
SDK for Java 2.x	부 분	공유 config 파일 설정은 지원되지 않습니다. 환경 변수는 지원되지 않습니다.
SDK for Java 1.x	아 니 요	

SDK	지 운	참고 또는 추가 정보
3.x용 JavaScript SDK	예	
2.x용 SDK JavaScript	아 니 요	
SDK for Kotlin	예	
SDK for .NET 3.x	예	환경 변수는 지원되지 않습니다.
SDK for PHP 3.x	예	
SDK for Python (Boto3)	예	
SDK for Ruby 3.x	예	
SDK for Rust	예	
다음에 위한 도구 PowerShell	아 니 요	

Amazon EC2 인스턴스 메타데이터

Amazon EC2는 인스턴스 메타데이터 서비스(IMDS)라는 인스턴스 서비스를 제공합니다. 이 서비스에 대한 자세한 내용은 Amazon EC2 사용 설명서의 [인스턴스 메타데이터 및 사용자 데이터](#) 또는 Amazon EC2 사용 설명서의 [인스턴스 메타데이터 및 사용자 데이터를](#) 참조하십시오. IAM 역할을 사용하여 구성된 Amazon EC2 인스턴스에서 보안 인증을 가져오려면 인스턴스 메타데이터 서비스에 대한 연결을 조정할 수 있습니다.

다음에 사용하여 이 기능을 구성하십시오.

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

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

기본값: 1

유효한 값: 1 보다 크거나 같음.

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

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

기본값: 1

유효한 값: 1 보다 크거나 같음.

config파일에서 이러한 값을 설정하는 예:

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

명령 행을 통한 환경 변수 설정의 Linux/macOS 예제:

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

명령줄을 통해 환경 변수를 설정하는 Windows 예제:

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

AWS SDK와의 호환성

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

SDK	지참고 또는 추가 정보 운
AWS CLI v2	예

SDK	지 원	참고 또는 추가 정보
SDK for C++	아 니 요	
SDK for Go V2 (1.x)	아 니 요	
SDK for Go 1.x (V1)	아 니 요	
SDK for Java 2.x	아 니 요	
SDK for Java 1.x	부 분	AWS_METADATA_SERVICE_TIMEOUT 만 지원됩니다.
3.x용 SDK JavaScript	아 니 요	
2.x용 SDK JavaScript	아 니 요	
SDK for Kotlin	아 니 요	
SDK for .NET 3.x	아 니 요	
SDK for PHP 3.x	예	

SDK	지 운	참고 또는 추가 정보
SDK for Python (Boto3)	예	
SDK for Ruby 3.x	아 니 요	
SDK for Rust	아 니 요	
다음에 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. AWS SDK를 사용하면 버킷 이름을 명시적으로 지정하는 대신 API 작업을 위해 버킷 필드에서 액세스 포인트 Amazon 리소스 이름 (ARN) 을 사용할 수 있습니다. 액세스 포인트 ARN과 [GetObject](#) 을 사용하여 버킷에서 객체를 가져오거나, 액세스 포인트 ARN과 [PutObject](#) 을 사용하여 버킷에 객체를 추가하는 등의 특정 작업에 이 방법을 사용합니다.

Amazon S3 액세스 포인트 및 ARN에 대한 자세한 내용은 Amazon S3 사용 설명서의 [액세스 포인트 사용](#)을 참조하십시오.

다음에 사용하여 이 기능을 구성하십시오.

s3_use_arn_region- 공유 파일 설정 **AWS config**, **AWS_S3_USE_ARN_REGION** - 환경 변수,
aws.s3UseArnRegion- JVM 시스템 속성: 자바/코틀린 전용, 코드에서 값을 직접 구성하려면 특정 SDK를 직접 참조하십시오.

이 설정은 SDK가 액세스 포인트 AWS 리전 ARN을 사용하여 요청에 대한 지역 엔드포인트를 구성할지 여부를 제어합니다. AWS 리전 SDK는 클라이언트 AWS 리전 구성과 동일한 AWS 파티션에서

ARN을 제공하는지 검증하여 실패할 가능성이 높은 파티션 간 호출이 발생하지 않도록 합니다. 다중 정의의 경우 코드로 구성된 설정이 우선 적용되고 환경 변수 설정이 그 뒤를 따릅니다.

기본값: `false`

유효한 값:

- **true**— SDK는 엔드포인트를 구성할 AWS 리전 때 구성된 클라이언트 대신 ARN을 사용합니다. AWS 리전예외: 구성된 클라이언트가 AWS 리전 AWS 리전 FIPS인 경우 ARN과 일치해야 합니다. AWS 리전이렇게 하지 않으면 오류가 발생합니다.
- **false**— SDK는 엔드포인트를 구성할 클라이언트가 구성한 AWS 리전을 사용합니다.

SDK와의 호환성 AWS

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

SDK	지	참고 또는 추가 정보
AWS CLI v2	예	
SDK for C++	예	
SDK for Go V2 (1.x)	예	
SDK for Go 1.x (V1)	예	공유 config 파일 설정을 사용하려면 구성 파일에서 로드를 켜야 합니다. 세션 을 참조하십시오.
SDK for Java 2.x	예	
SDK for Java 1.x	예	JVM 시스템 속성은 지원되지 않습니다.
3.x용 SDK JavaScript	예	
2.x용 SDK JavaScript	예	
SDK for Kotlin	예	

SDK	지 운	참고 또는 추가 정보
SDK for .NET 3.x	예	표준 우선 순위를 따르지 않습니다. 공유 config 파일 값이 환경 변수보다 우선합니다.
SDK for PHP 3.x	예	
SDK for Python (Boto3)	예	
SDK for Ruby 3.x	예	
SDK for Rust	아 니 요	
다음에 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 리전 액세스 포인트는 애플리케이션이 여러 AWS 리전리전에 있는 Amazon S3 버킷의 요청을 이행하는 데 사용할 수 있는 글로벌 엔드포인트를 제공합니다. 다중 리전 액세스 포인트를 사용하여 단일 리전에서 사용되는 것과 동일한 아키텍처로 다중 리전 애플리케이션을 구축하면 전 세계 어디에서나 해당 애플리케이션을 실행할 수 있습니다.

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

다중 리전 액세스 포인트 Amazon 리소스 이름(ARN)에 대한 자세한 내용을 알아보려면 Amazon S3 사용 설명서의 [다중 리전 액세스 포인트를 사용하여 요청 생성하기](#)을 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 [AWS 공용 런타임 \(CRT\) 라이브러리](#)에 대한 종속성을 통해 SDK로 획득됩니다.

다음에 사용하여 이 기능을 구성하십시오.

s3_disable_multiregion_access_points- 공유 AWS **config** 파일 설정, **AWS_S3_DISABLE_MULTIREGION_ACCESS_POINTS** - 환경 변수, **aws.s3DisableMultiRegionAccessPoints**- JVM 시스템 속성: 자바/코틀린 전용, 코드에서 값을 직접 구성하려면 특정 SDK를 직접 참조하십시오.

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

기본값: `false`

유효값:

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

AWS SDK와의 호환성

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

SDK	지 원	참고 또는 추가 정보
AWS CLI v2	예	
SDK for C++	예	
SDK for Go V2 (1.x)	예	
SDK for Go 1.x (V1)	아 니 요	
SDK for Java 2.x	예	
SDK for Java 1.x	아 니 요	
3.x용 SDK JavaScript	예	

SDK	지 운	참고 또는 추가 정보
2.x용 SDK JavaScript	아 니 요	
SDK for Kotlin	예	
SDK for .NET 3.x	예	
SDK for PHP 3.x	예	
SDK for Python (Boto3)	예	
SDK for Ruby 3.x	예	
SDK for Rust	예	
다음에 위한 도구 PowerShell	예	

AWS 리전

AWS 리전 작업 시 이해해야 할 중요한 AWS 서비스개념입니다.

AWS 리전을 사용하면 물리적으로 특정 지역에 AWS 서비스 있는 데이터에 액세스할 수 있습니다. 이는 데이터와 애플리케이션이 귀사 및 귀사의 사용자가 액세스하기 가까운 곳에서 계속 실행되도록 하는데 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 지역과 관련이 있습니다. 한 리전에서 생성한 리소스는 AWS 서비스에서 제공하는 복제 기능을 명시적으로 사용하지 않는 한 다른 리전에 존재하지 않습니다. 예를 들어, Amazon S3와 Amazon EC2 크로스 리전 복제를 지원합니다. IAM과 같은 일부 서비스의 경우 리전 리소스가 없습니다.

AWS 일반 참조에는 다음 정보가 포함됩니다.

- 리전과 엔드포인트 간의 관계를 이해하고 기존 리전 엔드포인트 목록을 보려면 [AWS 서비스 엔드포인트](#)를 참조하십시오.

- AWS 서비스 각각에 대해 지원되는 모든 지역 및 엔드포인트의 현재 목록을 보려면 [서비스 엔드포인트 및 할당량을 참조하십시오](#).

서비스 클라이언트 생성

프로그래밍 방식으로 AWS 서비스 액세스하기 위해 SDK는 각각에 대한 클라이언트 클래스/객체를 사용합니다. AWS 서비스 예를 들어 애플리케이션에서 Amazon EC2에 액세스해야 하는 경우 애플리케이션은 Amazon EC2 클라이언트 객체를 생성하여 해당 서비스와 인터페이스합니다.

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

다음을 사용하여 이 기능을 구성하십시오.

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

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

기본값: 없음. 이 값을 명시적으로 지정해야 합니다.

유효값:

- AWS 일반 참조의 [AWS 서비스 엔드포인트](#)에 나열된 대로 선택한 서비스에서 사용할 수 있는 모든 리전 코드. 예를 들어, 값은 엔드포인트를 AWS 리전 미국 동부 (버지니아 북부) 로 `us-east-1` 설정합니다.
- `aws-global()` 및 Amazon Simple Storage Service AWS Security Token Service (Amazon S3 AWS STS) 와 같은 지역별 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
```

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

SDK와의 AWS 호환성

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

SDK	지 문	참고 또는 추가 정보
AWS CLI v2	예	AWS CLI v2는 값이 <code>AWS_REGION</code> 입력되기 전에 모든 값을 사용합니다 <code>AWS_DEFAULT_REGION</code> (두 변수 모두 검사됨).
AWS CLI v1	예	AWS CLI v1은 이러한 <code>AWS_DEFAULT_REGION</code> 목적으로 명명된 환경 변수를 사용합니다.
SDK for C++	예	
SDK for Go V2 (1.x)	예	
SDK for Go 1.x (V1)	예	공유 config 파일 설정을 사용하려면 구성 파일에서 로드를 켜야 합니다. 세션 을 참조하십시오.
SDK for Java 2.x	예	
SDK for Java 1.x	예	
3.x용 SDK JavaScript	예	
2.x용 SDK JavaScript	예	
SDK for Kotlin	예	

SDK	지참고 또는 추가 정보 온
SDK for .NET 3.x	예
SDK for PHP 3.x	예
SDK for Python (Boto3)	예 이 SDK는 이러한 용도로 명명된 <code>AWS_DEFAULT_REGION</code> 환경 변수를 사용합니다.
SDK for Ruby 3.x	예
SDK for Rust	예
다음에 위한 도구 PowerShell	예

AWS STS 지역화된 엔드포인트

기본적으로 AWS Security Token Service (AWS STS) 는 글로벌 서비스로 사용할 수 있으며 모든 AWS STS 요청은 의 단일 엔드포인트로 이동합니다 <https://sts.amazonaws.com>. 글로벌 요청은 미국 동부 (버지니아 북부) 지역에 매핑됩니다. AWS 글로벌 AWS STS 엔드포인트 대신 리전 엔드포인트를 사용할 것을 권장합니다. AWS STS 엔드포인트에 대한 자세한 내용은 API 참조의 [AWS Security Token Service 엔드포인트](#)를 참조하십시오.

다음에 사용하여 이 기능을 구성하십시오.

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

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

기본값: legacy

Note

2022년 7월 이후에 출시되는 모든 새 SDK 메이저 버전은 regional으로 기본값이 설정됩니다. 새 SDK 메이저 버전에서는 regional 동작을 사용하여 이 설정을 없앨 수 있습니다.

이 변경으로 인한 향후 영향을 줄이려면 가능하면 사용자 애플리케이션에서 `regional`을 사용하여 시작하는 것이 좋습니다.

유효한 값: (권장 값: `regional`)

- **legacy**—,,,`sts.amazonaws.com`,,,,`ap-northeast-1`,,`ap-south-1`,`ap-southeast-1`,`ap-southeast-2`, `aws-global` `ca-central-1` `eu-central-1` `eu-north-1` `eu-west-1`, 및 AWS `us-west-2` 지역에 대해 글로벌 AWS STS 엔드포인트 `eu-west-2` `eu-west-3` `sa-east-1` `us-east-1` `us-east-2` `us-west-1`, 를 사용합니다. 다른 모든 리전은 해당 리전 엔드포인트를 자동으로 사용합니다.
- **regional**— SDK 또는 도구는 항상 현재 구성된 지역의 AWS STS 엔드포인트를 사용합니다. 예를 들어 클라이언트가 사용하도록 구성된 경우 에 대한 모든 AWS STS 호출은 글로벌 엔드포인트 대신 리전 `sts.us-west-2.amazonaws.com` `sts.amazonaws.com` 엔드포인트로 이루어집니다. `us-west-2` 이 설정이 활성화된 상태에서 글로벌 엔드포인트에 요청을 보내려면 리전을 `aws-global`로 설정하면 됩니다.

`config` 파일에서 이러한 값을 설정하는 예:

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

명령 행을 통한 환경 변수 설정의 Linux/macOS 예제:

```
export AWS_STS_REGIONAL_ENDPOINTS=regional
```

명령줄을 통해 환경 변수를 설정하는 Windows 예제:

```
setx AWS_STS_REGIONAL_ENDPOINTS regional
```

AWS SDK와의 호환성

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

SDK	지 운	참고 또는 추가 정보
AWS CLI v2	부 분	기본값은 regional입니다.
SDK for C++	부 분	환경 변수 및 config 파일 설정은 지원되지 않습니다. SDK는 regional 설정으로 수행됩니다.
SDK for Go V2 (1.x)	예	
SDK for Go 1.x (V1)	예	공유 config 파일 설정을 사용하려면 구성 파일에서 로드를 켜야 합니다. 세션 을 참조하십시오.
SDK for Java 2.x	예	
SDK for Java 1.x	예	
3.x용 SDK JavaScript	예	
2.x용 SDK JavaScript	예	
SDK for Kotlin	아 니 요	
SDK for .NET 3.x	예	
SDK for PHP 3.x	예	
SDK for Python (Boto3)	예	
SDK for Ruby 3.x	예	
SDK for Rust	예	
다음에 위한 도구 PowerShell	예	

이중 스택 엔드포인트 및 FIPS 엔드포인트

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

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

기본값: false

유효한 값:

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

use_fips_endpoint AWS config- 공유 파일 설정, **AWS_USE_FIPS_ENDPOINT** - 환경 변수, **aws.useFipsEndpoint**- JVM 시스템 속성: 자바/코틀린 전용

SDK 또는 도구로 FIPS 호환 엔드포인트로 요청을 보낼지 여부 활성화 혹은 비활성화 합니다. 연방 정보 처리 표준 (FIPS)은 데이터 및 암호화에 대한 미국 정부 보안 요구 사항의 집합입니다. 정부 기관, 파트너 및 연방 정부와 거래하려는 기관은 FIPS 지침을 준수해야 합니다. 표준 AWS 엔드포인트와 달리 FIPS 엔드포인트는 FIPS 140-2를 준수하는 TLS 소프트웨어 라이브러리를 사용합니다. 이 설정이 활성화되어 있는데 해당 서비스에 대한 FIPS 엔드포인트가 없는 경우 통화가 실패할 수 있습니다. AWS 리전 AWS [서비스별 엔드포인트](#) 그리고 이 AWS Command Line Interface 설정을 재정의하는 `--endpoint-url` 옵션도 있습니다.

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

기본값: false

유효값:

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

SDK와의 호환성 AWS

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

SDK	지	참고 또는 추가 정보
AWS CLI v2	예	
SDK for C++	예	
SDK for Go V2 (1.x)	예	
SDK for Go 1.x (V1)	예	공유 config 파일 설정을 사용하려면 구성 파일에서 로드를 켜야 합니다. 세션 을 참조하십시오.
SDK for Java 2.x	예	
SDK for Java 1.x	아 니 요	
3.x용 SDK JavaScript	예	
2.x용 SDK JavaScript	예	
SDK for Kotlin	예	
SDK for .NET 3.x	예	
SDK for PHP 3.x	예	
SDK for Python (Boto3)	예	
SDK for Ruby 3.x	예	
SDK for Rust	예	
다음에 위한 도구 PowerShell	예	

엔드포인트 검색

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

다음을 사용하여 이 기능을 구성하십시오.

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

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

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 `이`고 Amazon DynamoDB 요청은 기본적으로 `입니다`. `true` `false`

유효한 값:

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

SDK와의 호환성 AWS

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

SDK	지 운	참고 또는 추가 정보
AWS CLI v2	예	
SDK for C++	예	
SDK for Go V2 (1.x)	예	
SDK for Go 1.x (V1)	예	공유 config 파일 설정을 사용하려면 구성 파일에서 로드를 켜야 합니다. 세션 을 참조하십시오.
SDK for Java 2.x	예	Java 2.x용 SDK는 환경 AWS_ENDPOINT_DISCOVERY_ENABLED 변수 이름으로 사용됩니다.
SDK for Java 1.x	부 분	JVM 시스템 속성은 지원되지 않습니다.
3.x용 SDK JavaScript	예	
2.x용 SDK JavaScript	예	
SDK for Kotlin	예	
SDK for .NET 3.x	예	
SDK for PHP 3.x	예	
SDK for Python (Boto3)	예	
SDK for Ruby 3.x	예	
SDK for Rust	부 분	타임스트림에만 지원됩니다.
다음에 위한 도구 PowerShell	예	

일반 구성 설정

SDK는 전체 SDK 동작을 구성하는 몇 가지 일반 설정을 지원합니다.

다음을 사용하여 이 기능을 구성하십시오.

api_versions- 공유 AWS config 파일 설정

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

기본값: 없음. (SDK에는 최신 API 버전이 사용됩니다.)

유효한 값: 이 값은 각각 하나의 AWS 서비스와 사용할 API 버전을 식별하는 하나 이상의 들여쓰기된 줄이 뒤에 오는 중첩된 설정입니다. 사용 가능한 API 버전을 이해하려면 AWS 서비스 설명서를 참조하십시오.

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

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

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

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

기본값: 없음

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

config 파일에서 이 값을 설정하는 예:

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

명령 행을 통한 환경 변수 설정의 Linux/macOS 예제:

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

명령줄을 통해 환경 변수를 설정하는 Windows 예제:

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

output- 공유 AWS config 파일 설정

AWS CLI 및 기타 AWS SDK와 도구에서 결과 형식을 지정하는 방법을 지정합니다.

기본값: json

유효값:

- **json** - 출력은 [JSON](#) 문자열로 형식이 지정됩니다.
- **yaml** - 출력은 [YAML](#) 문자열로 형식이 지정됩니다.
- **yaml-stream** - 출력은 스트리밍되고 [YAML](#) 문자열로 형식이 지정됩니다. 스트리밍을 통해 대용량 데이터 유형을 빠르게 처리할 수 있습니다.
- **text** - 출력은 여러 줄의 탭으로 구분된 문자열 값으로 형식이 지정됩니다. 출력을 grep, sed 또는 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 파일 설정 AWS config

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

기본값: true

유효값:

- **true** - 기본값입니다. SDK 또는 도구는 명령줄 파라미터를 클라이언트측에서 검증합니다. 이렇게 하면 SDK 또는 도구가 파라미터가 유효한지 확인하고 일부 오류를 포착하는 데 도움이 됩니다. SDK 또는 도구는 AWS 서비스 엔드포인트로 요청을 보내기 전에 유효하지 않은 요청을 거부할 수 있습니다.
- **false** - SDK 또는 도구는 명령줄 매개변수를 서비스 엔드포인트로 보내기 전에 유효성을 검사하지 않습니다. AWS 서비스 엔드포인트는 모든 요청의 유효성을 검사하고 유효하지 않은 요청은 거부합니다.

SDK와의 호환성 AWS

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

SDK	지 운	참고 또는 추가 정보
AWS CLI v2	부 분	api_versions 이 지원되지 않음.
SDK for C++	예	
SDK for Go V2 (1.x)	부 분	api_versions 및 parameter_validation 이 지원되지 않음.
SDK for Go 1.x (V1)	부 분	api_versions 및 parameter_validation 이 지원되지 않음. 공유 config 파일 설정을 사용하려면 구성 파일에서 로드를 켜야 합니다. 세션 을 참조하십시오.
SDK for Java 2.x	아 니 요	
SDK for Java 1.x	아 니 요	
3.x용 SDK JavaScript	예	
2.x용 SDK JavaScript	예	
SDK for Kotlin	아 니 요	
SDK for .NET 3.x	아 니 요	

SDK	지 운	참고 또는 추가 정보
SDK for PHP 3.x	예	
SDK for Python (Boto3)	예	
SDK for Ruby 3.x	예	
SDK for Rust	아 니 요	
다음에 위한 도구 PowerShell	아 니 요	

IMDS 클라이언트

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

다음에 사용하여 이 기능을 구성하십시오.

retries - 클라이언트 구성 객체 멤버

실패한 요청에 대한 추가 재시도 횟수입니다.

기본값: 3

유효한 값: 0보다 큰 숫자.

port - 클라이언트 구성 객체 멤버

엔드포인트 포트.

기본값: 80

유효한 값: 숫자.

token_ttl - 클라이언트 구성 객체 멤버

토큰의 TTL.

기본값: 21,600초(6시간, 최대 할당 시간).

유효한 값: 숫자.

endpoint - 클라이언트 구성 객체 멤버

IMDS 엔드포인트.

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

유효한 값: 유효한 URI.

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

endpoint_mode- 클라이언트 구성 객체 멤버

IMDS의 엔드포인트 모드.

기본값: IPv4

유효한 값: IPv4, IPv6

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

연결이 열릴 때까지 기다리는 시간 (초).

기본값: 1초.

유효한 값: 0보다 큰 숫자.

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

데이터 청크 하나를 읽는 데 걸리는 시간 (초).

기본값: 1초.

유효한 값: 0보다 큰 숫자.

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

디버깅을 위한 출력 스트림을 설정합니다.

기본값: 없음.

유효한 값: STDOUT과 같은 유효한 I/O 스트림.

backoff- 클라이언트 구성 객체 멤버 (이름은 다를 수 있음)

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

기본값: SDK에 따라 다릅니다.

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

AWS SDK와의 호환성

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

SDK	지 운	참고 또는 추가 정보
AWS CLI v2	예	
SDK for C++	아 니 요	IMDSv2는 내부적으로만 사용됩니다. IMDS 보안 인증 제공자 섹션을 참조하십시오.
SDK for Go V2 (1.x)	예	
SDK for Go 1.x (V1)	예	
SDK for Java 2.x	예	
SDK for Java 1.x	예	
3.x용 SDK JavaScript	예	
2.x용 SDK JavaScript	예	
SDK for Kotlin	예	
SDK for .NET 3.x	예	

SDK	지참고 또는 추가 정보 온
SDK for PHP 3.x	예
SDK for Python (Boto3)	예
SDK for Ruby 3.x	예
SDK for Rust	예
다음에 위한 도구 PowerShell	예

재시도 동작

재시도 동작에는 SDK가 AWS 서비스에 보내진 요청으로 인한 장애 복구를 시도하는 방법에 대한 설정이 포함됩니다.

다음에 사용하여 이 기능을 구성하십시오.

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`이 `legacy` 인 경우— 사용자 SDK 고유의 기본값을 사용합니다(사용자 고유 SDK 설명서 또는 `max_attempts` 기본 SDK의 코드베이스를 확인하십시오).
- `retry_mode`이 `standard`인 경우 — 세 번 시도합니다.
- `retry_mode`이 `adaptive`인 경우 — 세 번 시도합니다.

유효한 값: 0보다 큰 숫자.

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

SDK 또는 개발자 도구의 재시도 방법을 지정합니다.

기본값: `legacy`가 기본 재시도 전략입니다.

유효값:

- **legacy** — 사용자 SDK에 고유한 값(사용자 고유 SDK 설명서 또는 기본 SDK의 코드베이스를 확인하십시오).
- **standard** — SDK 전반의 표준 재시도 규칙 세트. AWS 이 모드에는 재시도되는 표준 오류 설정과 재시도 할당량 지원이 포함되어 있습니다. 명시적으로 `max_attempts`으로 구성되어 있지 않으면 이 모드의 기본 최대 시도 횟수는 3회입니다.
- **adaptive** - 표준 모드의 기능을 포함하지만 자동 클라이언트 측 제한도 포함하는 실험용 재시도 모드입니다. 이 모드는 실험용이므로 향후 동작이 변경될 수 있습니다.

모드 선택 `standard` 및 재시도 모드 `adaptive`

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: <ol style="list-style-type: none"> 1. 지연 시간에 민감하지 않습니다. 2. 클라이언트는 단일 리소스에만 액세스하거나 액세스 중인 서비스 리소스별로 클라이언트를 별도로 풀링하는 로직을 제공합니다.
정전 시 SDK가 재시도되지 않도록 회로 차단이 지원됩니다.	정전 시 SDK가 재시도하지 못하도록 회로 차단을 지원합니다.
장애 발생 시 지터링 지수 백오프를 사용합니다.	지연 시간이 증가할 수 있는 대신 동적 백오프 기간을 사용하여 실패한 요청 수를 최소화하려고 합니다.

표준	적응형
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 리소스에 대해 생각하는 것보다 더 세밀하게 조정됩니다. AWS 서비스 AWS 서비스 요청을 제한하는 데 사용하는 추가 차원이 있을 수 있습니다. Amazon DynamoDB 서비스를 예로 들어 보겠습니다. DynamoDB는 요청을 제한하기 위해 액세스 중인 테이블과 함께 AWS 리전 사용합니다. 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 사용 중인 서비스의 특정 AWS 설명서 가이드를 참조하십시오.

재시도 모드 구현 세부 정보

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:

토큰 버킷은 adaptive 재시도 모드에서만 사용됩니다. 토큰 버킷은 요청을 시작하는 데 토큰을 사용할 수 있도록 요구함으로써 최대 요청 비율을 적용합니다. SDK 클라이언트는 요청을 빠르게 중단시키거나 토큰 사용이 가능할 때까지 요청을 차단하도록 구성할 수 있습니다.

클라이언트측 속도 제한은 처음부터 원하는 속도를 토큰 허용량까지 요청을 할 수 있도록 하는 알고리즘입니다. 그러나 병목 현상이 감지된 후에는 클라이언트가 그에 따라 rate-of-request 제한됩니다. 응답이 성공적으로 수신되면 토큰 허용량도 그에 따라 증가합니다.

적응형 속도 제한을 사용하면 SDK가 요청 전송 속도를 늦춰 요청 용량을 더 잘 수용할 수 있습니다. AWS 서비스

SendHttpRequest:

대부분의 AWS SDK는 연결 풀을 사용하는 HTTP 라이브러리를 사용하므로 HTTP 요청 시 기존 연결을 재사용할 수 있습니다. 일반적으로 조절 오류로 인해 요청의 재시도 시 연결을 재사용합니다. 일시적 오류로 인한 재시도 때는 요청이 재사용되지 않습니다.

RequestBookkeeping:

요청이 성공하면 재시도 할당량을 업데이트해야 합니다. adaptive재시도 모드의 경우에만 상태 변수maxsendrate는 수신된 응답 유형에 따라 업데이트됩니다.

Retryable:

이 단계에서는 다음을 기반으로 응답을 재시도할 수 있는지 여부를 결정합니다.

- HTTP 상태 코드 .
- 서비스에서 반환된 오류 코드입니다.
- 연결 오류로, 서비스로부터 HTTP 응답을 받지 못한 SDK에서 수신한 모든 오류로 정의됩니다.

일시적 오류(HTTP 상태 코드 400, 408, 500, 502, 503, 504)와 조절 오류(HTTP 상태 코드 400, 403, 429, 502, 503, 509)는 모두 재시도될 수 있습니다. SDK 재시도 동작은 서비스의 오류 코드 또는 서비스의 기타 데이터와 조합하여 결정됩니다.

MAX_ATTEMPTS:

config 파일 설정 또는 환경 변수로 지정합니다.

HasRetryQuota

이 단계에서는 재시도 할당량 버킷에서 토큰을 사용할 수 있도록 요구하여 재시도 요청을 제한합니다. 재시도 할당량 버킷은 성공할 가능성이 낮은 재시도를 방지하는 메커니즘입니다. 이러한 할당량은 SDK에 따라 달라지고 클라이언트에 따라 달라지는 경우가 많으며 때로는 서비스 엔드포인트에 따라 달라지기도 합니다. 다양한 이유로 요청이 실패하면 사용 가능한 재시도 할당량 토큰은 제거되고 요청이 성공하면 보충됩니다. 토큰이 남아 있지 않으면 재시도 루프가 종료됩니다.

ExponentialBackoff

재시도할 수 있는 오류의 경우 잘린 지수 백오프를 사용하여 재시도 지연을 계산합니다. SDK는 지터가 있는 잘린 이진 지수 백오프를 사용합니다. 다음 알고리즘은 요청에 대한 응답의 절전 시간(초)이 어떻게 정의되는지 보여줍니다. i

```
seconds_to_sleep_i = min(b*r^i, MAX_BACKOFF)
```

위 알고리즘에는 다음 값이 적용됩니다.

b = random number within the range of: $0 \leq b \leq 1$

r = 2

대부분의 SDK용 MAX_BACKOFF = 20 seconds. 사용자 특정 SDK 가이드 또는 소스 코드를 참조하여 확인합니다.

SDK와의 호환성 AWS

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

SDK	지참고 또는 추가 정보 운
AWS CLI v2	예
SDK for C++	예
SDK for Go V2 (1.x)	예
SDK for Go 1.x (V1)	아 니 요

SDK	지 운	참고 또는 추가 정보
SDK for Java 2.x	예	
SDK for Java 1.x	예	JVM 시스템 속성: <code>com.amazonaws.sdk.maxAttempts</code> 대신 사용, <code>com.amazonaws.sdk.retryMode</code> 대신 사용 <code>aws.maxAttempts . aws.retryMode</code>
3.x용 SDK JavaScript	예	
2.x용 SDK JavaScript	아 니 요	최대 재시도 횟수, 지터가 있는 지수 백오프, 재시도 백오프를 위한 사용자 지정 방법 옵션을 지원합니다.
SDK for Kotlin	예	
SDK for .NET 3.x	예	
SDK for PHP 3.x	예	
SDK for Python (Boto3)	예	
SDK for Ruby 3.x	예	
SDK for Rust	예	
다음에 위한 도구 PowerShell	예	

요청 압축

AWS SDK 및 도구는 압축된 페이로드 수신을 지원하는 요청을 전송할 때 페이로드를 자동으로 압축할 수 있습니다. AWS 서비스 페이로드를 서비스로 보내기 전에 클라이언트에서 압축하면 데이터를 서비스로 보내는 데 필요한 전체 요청 수와 대역폭을 줄일 수 있을 뿐만 아니라 페이로드 크기에 대한 서비스 제한으로 인해 실패한 요청도 줄일 수 있습니다. 압축을 위해 SDK 또는 도구는 서비스와 SDK에서 모두 지원하는 인코딩 알고리즘을 선택합니다. 그러나 현재 가능한 인코딩 목록은 gzip으로만 구성되어 있지만 향후 확장될 수 있습니다.

요청 압축은 애플리케이션에서 [Amazon](#)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

요청을 보내기 전에 SDK 또는 도구가 페이로드를 압축할지 여부를 설정하거나 해제합니다.

기본값: false

유효값:

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

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

SDK 또는 도구가 압축해야 하는 요청 본문의 최소 크기(바이트)를 설정합니다. 작은 페이로드는 압축 시 더 길어질 수 있으므로 압축을 수행하는 데 적합한 하한선이 있습니다. 이 값은 포함되며, 이 값보다 크거나 같은 요청 크기는 압축됩니다.

기본값: 10240바이트

유효값: 0~10485760바이트 사이의 정수 값.

AWS SDK와의 호환성

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

SDK	지참고 또는 추가 정보
AWS CLI v2	예

SDK	지 운	참고 또는 추가 정보
SDK for C++	예	
SDK for Go V2 (1.x)	예	
SDK for Go 1.x (V1)	아 니 요	
SDK for Java 2.x	예	
SDK for Java 1.x	아 니 요	
3.x용 SDK JavaScript	예	
2.x용 SDK JavaScript	아 니 요	
SDK for Kotlin	예	
SDK for .NET 3.x	예	
SDK for PHP 3.x	예	
SDK for Python (Boto3)	예	
SDK for Ruby 3.x	예	
SDK for Rust	예	
다음에 위한 도구 PowerShell	예	

서비스별 엔드포인트

서비스별 엔드포인트 구성은 API 요청에 대해 사용자가 선택한 엔드포인트를 사용하고 선택을 유지할 수 있는 옵션을 제공합니다. 이러한 설정은 로컬 엔드포인트, VPC 엔드포인트 및 타사 로컬 AWS 개발 환경을 지원할 수 있는 유연성을 제공합니다. 테스트 환경과 프로덕션 환경에 서로 다른 엔드포인트를 사용할 수 있습니다. 개별 AWS 서비스서비스에 대한 엔드포인트 URL을 지정할 수 있습니다.

다음을 사용하여 이 기능을 구성하십시오.

endpoint_url- 공유 AWS config 파일 설정, **AWS_ENDPOINT_URL** - 환경 변수,
aws.endpointUrl- JVM 시스템 속성: 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 AWS config 파일의 `services` 섹션 내에서 이 설정을 사용하여 특정 서비스에 대한 사용자 지정 엔드포인트를 설정할 수도 있습니다. `services` 섹션에서 사용할 모든 서비스 식별자 키 목록은 [서비스별 엔드포인트 식별자](#) 섹션을 참조하십시오.

기본값: none

유효한 값: 엔드포인트의 스키마와 호스트가 포함된 URL. URL에는 하나 이상의 경로 세그먼트가 포함된 경로 구성 요소가 선택적으로 포함될 수 있습니다.

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

AWS_ENDPOINT_URL_<SERVICE>, **<SERVICE>** 는 AWS 서비스 식별자로, 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 호스트가 포함된 URL. URL에는 하나 이상의 경로 세그먼트가 포함된 경로 구성 요소가 선택적으로 포함될 수 있습니다.

ignore_configured_endpoint_urls- 공유 AWS config 파일 설정, **AWS_IGNORE_CONFIGURED_ENDPOINT_URLS** - 환경 변수, **aws.ignoreConfiguredEndpointUrls**- JVM 시스템 속성: 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 명령어에 `--endpoint-url` 명령줄 매개변수를 포함하거나 클라이언트 생성자에 엔드포인트 URL을 전달하면 항상 적용됩니다. AWS CLI

기본값: `false`

유효값:

- **true**— SDK 또는 도구는 공유 config 파일 또는 환경 변수에서 엔드포인트 URL 설정을 위한 사용자 지정 구성 옵션을 읽지 않습니다.
- **false**— SDK 또는 도구는 공유 config 파일 또는 환경 변수에서 사용 가능한 사용자 제공 엔드포인트를 사용합니다.

환경 변수를 사용한 엔드포인트 구성

모든 서비스에 대한 요청을 사용자 지정 엔드포인트 URL로 라우팅하려면 `AWS_ENDPOINT_URL` 글로벌 환경 변수를 설정하십시오.

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

특정 요청을 사용자 지정 엔드포인트 URL로 AWS 서비스 라우팅하려면 `AWS_ENDPOINT_URL_<SERVICE>` 환경 변수를 사용합니다. Amazon DynamoDB a에는 `serviceId` a가 [DynamoDB](#) 있습니다. 이 서비스의 경우 엔드포인트 URL 환경 변수는 `AWS_ENDPOINT_URL_DYNAMODB`입니다. 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](#). AWS 서비스 식별자는 모든 공백을 `serviceId` 밑줄로 바꾸고 모든 문자를 대문자로 바꾸는 API 모델을 기반으로 합니다. 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](#)을 참조하십시오.

다음 예제에서는 `services` 정의를 사용하여 Amazon S3에 대한 서비스별 엔드포인트 URL과 다른 모든 서비스에 사용되는 사용자 지정 글로벌 엔드포인트를 구성합니다.

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

[services s3-specific]
s3 =
  endpoint_url = https://play.min.io:9000
```

단일 프로파일로 여러 서비스에 대한 엔드포인트를 구성할 수 있습니다. 이 예제는 동일한 프로파일에서 Amazon S3의 서비스별 엔드포인트 URL을 설정하는 방법을 보여줍니다. AWS Elastic Beanstalk에는 `a`가 있습니다. `serviceId` [Elastic Beanstalk](#) AWS 서비스 식별자는 모든 공백을 `serviceId` 밑줄로 바꾸고 모든 문자는 소문자로 바꾸는 API 모델을 기반으로 합니다. 따라서 서비스 식별자 키가 `elastic_beanstalk`이(가)되고 이 서비스에 대한 설정이 `elastic_beanstalk =` 줄에서 시작됩니다. `services` 섹션에서 사용할 모든 서비스 식별자 키 목록은 [서비스별 엔드포인트 식별자](#)을 참조하십시오.

```
[services testing-s3-and-eb]
s3 =
  endpoint_url = http://localhost:4567
elastic_beanstalk =
  endpoint_url = http://localhost:8000

[profile dev]
```

```
services = testing-s3-and-eb
```

서비스 구성 섹션은 여러 프로파일에서 사용할 수 있습니다. 예를 들어 두 프로파일이 동일한 `services` 정의를 사용하면서 다른 프로파일 속성을 변경할 수 있습니다.

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

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

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

역할 기반 보안 인증을 사용하여 프로파일의 엔드포인트 구성

프로파일에 역할 기반 보안 인증 정보가 IAM 가정 역할 기능에 대한 `source_profile` 파라미터를 통해 구성된 경우 SDK는 지정된 프로파일에 대한 서비스 구성만 사용합니다. 역할이 연결된 프로파일은 사용하지 않습니다. 예를 들어 다음과 같은 공유 config 파일을 사용합니다.

```
[profile A]
credential_source = Ec2InstanceMetadata
endpoint_url = https://profile-a-endpoint.aws/

[profile B]
source_profile = A
role_arn = arn:aws:iam::123456789012:role/roleB
services = profileB

[services profileB]
ec2 =
  endpoint_url = https://profile-b-ec2-endpoint.aws
```

프로파일 B를 사용하고 코드에서 Amazon EC2로 호출하는 경우 엔드포인트는 `https://profile-b-ec2-endpoint.aws`로 확인됩니다. 코드에서 다른 서비스에 요청을 하는 경우 엔드포인트 확인은 사용자 지정 로직을 따르지 않습니다. 엔드포인트는 프로파일 A에 정의된 글로벌 엔드포인트로 확인되지 않습니다. 글로벌 엔드포인트가 프로파일 B에 적용되려면 프로파일 B 내에서 직접

`endpoint_url`을 설정해야 합니다. `source_profile` 설정에 대한 자세한 내용은 [역할 보안 인증 제 공자 수 입 단 원](#)을 참조하십시오.

설정의 우선 순위

이 기능의 설정은 동시에 사용할 수 있지만 서비스당 하나의 값만 우선합니다. 주어진 항목에 대한 API 호출의 AWS 서비스 경우 다음 순서에 따라 값을 선택합니다.

1. 코드나 서비스 클라이언트 자체에 설정된 모든 명시적 설정은 다른 모든 설정보다 우선합니다.
 - 의 AWS CLI 경우 이 값은 `--endpoint-url` 명령줄 파라미터가 제공하는 값입니다. SDK의 경우 명시적 할당은 클라이언트 또는 구성 객체를 인스턴스화할 때 설정하는 파라미터의 형태를 취할 수 있습니다. AWS 서비스
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 URL의 기본 엔드포인트 URL이 마지막으로 사용됩니다. AWS 서비스

AWS SDK와의 호환성

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

SDK	지 운	참고 또는 추가 정보
AWS CLI v2	예	
SDK for C++	아 니 요	
SDK for Go V2 (1.x)	예	

SDK	지 운	참고 또는 추가 정보
SDK for Go 1.x (V1)	아 니 요	
SDK for Java 2.x	아 니 요	
SDK for Java 1.x	아 니 요	
3.x용 SDK JavaScript	예	
2.x용 SDK JavaScript	아 니 요	
SDK for Kotlin	예	
SDK for .NET 3.x	예	
SDK for PHP 3.x	예	
SDK for Python (Boto3)	예	
SDK for Ruby 3.x	예	
SDK for Rust	아 니 요	
다음에 위한 도구 PowerShell	예	

서비스별 엔드포인트 식별자

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

serviceId	공 유 시 디 파 일 으 서 비 스 별 식 별 자 키	환경 변수
AccessAnalyzer	ar	AWS_ENDPOINT_URL_ACCESSANALYZER
Account	ar	AWS_ENDPOINT_URL_ACCOUNT
ACM	ar	AWS_ENDPOINT_URL_ACM
ACM PCA	ar	AWS_ENDPOINT_URL_ACM_PCA
Alexa For Business	ar	AWS_ENDPOINT_URL_ALEXA_FOR_BUSINESS
amp	ar	AWS_ENDPOINT_URL_AMP
Amplify	ar	AWS_ENDPOINT_URL_AMPLIFY
AmplifyBackend	ar	AWS_ENDPOINT_URL_AMPLIFYBACKEND

serviceId	공 유 시 스템 의 서 버 스 별 자 크	환경 변수
AmplifyUIBuilder	ar bi	AWS_ENDPOINT_URL_AMPLIFYUIBUILDER
API Gateway	a a	AWS_ENDPOINT_URL_API_GATEWAY
ApiGatewayManagem entApi	a y n	AWS_ENDPOINT_URL_APIGATEWAYMANAGEMENTAPI
ApiGatewayV2	a y	AWS_ENDPOINT_URL_APIGATEWAYV2
AppConfig	a	AWS_ENDPOINT_URL_APPCONFIG
AppConfigData	a d	AWS_ENDPOINT_URL_APPCONFIGDATA
AppFabric	a	AWS_ENDPOINT_URL_APPFABRIC
Appflow	a	AWS_ENDPOINT_URL_APPFLOW

serviceId	공유 서비스 디폴트 서비스별 자카 AWS_ENDPOINT_URL_<SERVICE> 환경 변수
AppIntegrations	a) AWS_ENDPOINT_URL_APPINTEGRATIONS a)
Application Auto Scaling	a) AWS_ENDPOINT_URL_APPLICATION_AUTO_SCALING o) c:
Application Insights	a) AWS_ENDPOINT_URL_APPLICATION_INSIGHTS o) t:
ApplicationCostProfiler	a) AWS_ENDPOINT_URL_APPLICATIONCOSTPROFILER o) f:
App Mesh	a) AWS_ENDPOINT_URL_APP_MESH
AppRunner	a) AWS_ENDPOINT_URL_APPRUNNER
AppStream	a) AWS_ENDPOINT_URL_APPSTREAM

serviceId	공용 서비스 디폴트 엔드포인트 URL 환경 변수
AppSync	a: AWS_ENDPOINT_URL_APPSYNC
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

serviceId	공유 서비스별 환경 변수
Backup Gateway	b: AWS_ENDPOINT_URL_BACKUP_GATEWAY t:
BackupStorage	b: AWS_ENDPOINT_URL_BACKUPSTORAGE r:
Batch	b: AWS_ENDPOINT_URL_BATCH
BCM Data Exports	b: AWS_ENDPOINT_URL_BCM_DATA_EXPORTS e:
Bedrock	b: AWS_ENDPOINT_URL_BEDROCK
Bedrock Agent	b: AWS_ENDPOINT_URL_BEDROCK_AGENT g:
Bedrock Agent Runtime	b: AWS_ENDPOINT_URL_BEDROCK_AGENT_RUNTIME g: ir
Bedrock Runtime	b: AWS_ENDPOINT_URL_BEDROCK_RUNTIME ur

serviceId	공유 서비스별 환경 변수
billingconductor	b: AWS_ENDPOINT_URL_BILLINGCONDUCTOR
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

serviceId	공공 AWS_ENDPOINT_URL_<SERVICE> 환경 변수 유니코드 파일의 서브시스템별 자크
Chime SDK Meetings	c:\AWS_ENDPOINT_URL_CHIME_SDK_MEETINGS_1
Chime SDK Messaging	c:\AWS_ENDPOINT_URL_CHIME_SDK_MESSAGING_1g
Chime SDK Voice	c:\AWS_ENDPOINT_URL_CHIME_SDK_VOICE_1
CleanRooms	c:\AWS_ENDPOINT_URL_CLEANROOMS_S
CleanRoomsML	c:\AWS_ENDPOINT_URL_CLEANROOMSML_SF
Cloud9	c:\AWS_ENDPOINT_URL_CLOUD9
CloudControl	c:\AWS_ENDPOINT_URL_CLOUDCONTROL_I

serviceId	공용 AWS_ENDPOINT_URL_<SERVICE> 환경 변수 유니코드 파일의 서비스별 자크
CloudDirectory	c:\AWS_ENDPOINT_URL_CLOUDDIRECTORY
CloudFormation	c:\AWS_ENDPOINT_URL_CLOUDFORMATION
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

serviceId	공용 서비스 디폴트 엔드포인트 URL	환경 변수
CloudSearch Domain	c: AWS_ENDPOINT_URL_CLOUDSEARCH_DOMAIN	cl
CloudTrail	c: AWS_ENDPOINT_URL_CLOUDTRAIL	l
CloudTrail Data	c: AWS_ENDPOINT_URL_CLOUDTRAIL_DATA	l_
CloudWatch	c: AWS_ENDPOINT_URL_CLOUDWATCH	h
codeartifact	c: AWS_ENDPOINT_URL_CODEARTIFACT	ar
CodeBuild	c: AWS_ENDPOINT_URL_CODEBUILD	
CodeCatalyst	c: AWS_ENDPOINT_URL_CODECATALYST	ys

serviceId	공용 AWS_ENDPOINT_URL_<SERVICE> 환경 변수
CodeCommit	AWS_ENDPOINT_URL_CODECOMMIT
CodeDeploy	AWS_ENDPOINT_URL_CODEDEPLOY
CodeGuru Reviewer	AWS_ENDPOINT_URL_CODEGURU_REVIEWER
CodeGuru Security	AWS_ENDPOINT_URL_CODEGURU_SECURITY
CodeGuruProfiler	AWS_ENDPOINT_URL_CODEGURUPROFILER
CodePipeline	AWS_ENDPOINT_URL_CODEPIPELINE
CodeStar	AWS_ENDPOINT_URL_CODESTAR
CodeStar connections	AWS_ENDPOINT_URL_CODESTAR_CONNECTIONS

serviceId	공용 AWS_ENDPOINT_URL_<SERVICE> 환경 변수 유니코드 파일 시스템 별칭
codestar notifications	AWS_ENDPOINT_URL_CODESTAR_NOTIFICATIONS
Cognito Identity	AWS_ENDPOINT_URL_COGNITO_IDENTITY
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

serviceId	공용 서비스 디폴트 엔드포인트 URL AWS_ENDPOINT_URL_<SERVICE> 환경 변수
Config Service	AWS_ENDPOINT_URL_CONFIG_SERVICE
Connect	AWS_ENDPOINT_URL_CONNECT
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

serviceId	공용 AWS_ENDPOINT_URL_<SERVICE> 환경 변수
Cost Optimization Hub	c: AWS_ENDPOINT_URL_COST_OPTIMIZATION_HUB m: h:
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

serviceId	공용 AWS_ENDPOINT_URL_<SERVICE> 환경 변수 유니코드 파일의 서비스별 자크
DAX	d: AWS_ENDPOINT_URL_DAX
Detective	d: AWS_ENDPOINT_URL_DETECTIVE
Device Farm	d: AWS_ENDPOINT_URL_DEVICE_FARM r
DevOps Guru	d: AWS_ENDPOINT_URL_DEVOPS_GURU r
Direct Connect	d: AWS_ENDPOINT_URL_DIRECT_CONNECT n
Application Discovery Service	a: AWS_ENDPOINT_URL_APPLICATION_DISCOVER o: RY_SERVICE e: c:
DLM	d: AWS_ENDPOINT_URL_DLM

serviceId	공용 AWS_ENDPOINT_URL_<SERVICE> 환경 변수 유니코드 파일 시스템 별칭
Database Migration Service	d: AWS_ENDPOINT_URL_DATABASE_MIGRATION_<SERVICE>
DocDB	d: AWS_ENDPOINT_URL_DOCDB
DocDB Elastic	d: AWS_ENDPOINT_URL_DOCDB_ELASTIC<SERVICE>
drs	d: AWS_ENDPOINT_URL_DRS
Directory Service	d: AWS_ENDPOINT_URL_DIRECTORY_SERVICE<SERVICE>
DynamoDB	d: AWS_ENDPOINT_URL_DYNAMODB
DynamoDB Streams	d: AWS_ENDPOINT_URL_DYNAMODB_STREAMS<SERVICE>
EBS	e: AWS_ENDPOINT_URL_EBS
EC2	e: AWS_ENDPOINT_URL_EC2

serviceId	공용 서비스 디폴트 엔드포인트 URL 환경 변수
EC2 Instance Connect	e: AWS_ENDPOINT_URL_EC2_INSTANCE_CONNECT
ECR	e: AWS_ENDPOINT_URL_ECR
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

serviceId	공용 AWS_ENDPOINT_URL_<SERVICE> 환경 변수 유니코드 파일의 서비스별 자식
Elastic Beanstalk	e: AWS_ENDPOINT_URL_ELASTIC_BEANSTALK e:
Elastic Transcoder	e: AWS_ENDPOINT_URL_ELASTIC_TRANSCODER r:
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:

serviceId	공용 AWS_ENDPOINT_URL_<SERVICE> 환경 변수 유니코드 파일의 서비스별 자크
EntityResolution	e: AWS_ENDPOINT_URL_ENTITYRESOLUTION
Elasticsearch Service	e: AWS_ENDPOINT_URL_ELASTICSEARCH_SERVICE
EventBridge	e: AWS_ENDPOINT_URL_EVENTBRIDGE
Evidently	e: AWS_ENDPOINT_URL_EVIDENTLY
finspace	f: AWS_ENDPOINT_URL_Finspace
finspace data	f: AWS_ENDPOINT_URL_Finspace_data
Firehose	f: AWS_ENDPOINT_URL_FIREHOSE
fis	f: AWS_ENDPOINT_URL_FIS
FMS	fr: AWS_ENDPOINT_URL_FMS

serviceId	공유 서비스별 환경 변수
forecast	f: AWS_ENDPOINT_URL_FORECAST
forecastquery	f: AWS_ENDPOINT_URL_FORECASTQUERY
FraudDetector	f: AWS_ENDPOINT_URL_FRAUDETECTOR
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
Glue	g: AWS_ENDPOINT_URL_GLUE
grafana	g: AWS_ENDPOINT_URL_GRAFANA

serviceId	공공 AWS_ENDPOINT_URL_<SERVICE> 환경 변수 유니코드 파일의 서비스별 자크
Greengrass	g: AWS_ENDPOINT_URL_GREENGRASS s
GreengrassV2	g: AWS_ENDPOINT_URL_GREENGRASSV2 s\
GroundStation	g: AWS_ENDPOINT_URL_GROUNDSTATION t:
GuardDuty	g: AWS_ENDPOINT_URL_GUARDDUTY
Health	h: AWS_ENDPOINT_URL_HEALTH
HealthLake	h: AWS_ENDPOINT_URL_HEALTHLAKE e
Honeycode	h: AWS_ENDPOINT_URL_HONEYCODE
IAM	i: AWS_ENDPOINT_URL_IAM

serviceId	공유 서비스 디폴트 엔드포인트 URL 환경 변수
identitystore	AWS_ENDPOINT_URL_IDENTITYSTORE
imagebuilder	AWS_ENDPOINT_URL_IMAGEBUILDER
ImportExport	AWS_ENDPOINT_URL_IMPORTEXPORT
Inspector	AWS_ENDPOINT_URL_INSPECTOR
Inspector Scan	AWS_ENDPOINT_URL_INSPECTOR_SCAN
Inspector2	AWS_ENDPOINT_URL_INSPECTOR2
InternetMonitor	AWS_ENDPOINT_URL_INTERNETMONITOR
IoT	AWS_ENDPOINT_URL_IOT

serviceId	공용 AWS_ENDPOINT_URL_<SERVICE> 환경 변수 유니코드 파일의 서브시스템별 자크
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
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

serviceId	공용 서비스 디폴트 엔드포인트 URL	환경 변수
IoT Events Data	iot-aws-endpoint-url-iot-events-data	
IoTFleetHub	iot-aws-endpoint-url-iotfleethub	
IoTFleetWise	iot-aws-endpoint-url-iotfleetwise	
IoTSecureTunneling	iot-aws-endpoint-url-iotsecuretunneling	
IoTSiteWise	iot-aws-endpoint-url-iotsitewise	
IoTThingsGraph	iot-aws-endpoint-url-iotthingsgraph	
IoTTwinMaker	iot-aws-endpoint-url-iottwinmaker	

serviceId	공용 서비스 디폴트 서비스별 자	환경 변수
IoT Wireless	i	AWS_ENDPOINT_URL_IOT_WIRELESS
ivs	i	AWS_ENDPOINT_URL_IVS
IVS RealTime	i	AWS_ENDPOINT_URL_IVS_REALTIME
ivschat	i	AWS_ENDPOINT_URL_IVSCHAT
Kafka	k	AWS_ENDPOINT_URL_KAFKA
KafkaConnect	k	AWS_ENDPOINT_URL_KAFKACONNECT
kendra	k	AWS_ENDPOINT_URL_KENDRA
Kendra Ranking	k	AWS_ENDPOINT_URL_KENDRA_RANKING
Keyspaces	k	AWS_ENDPOINT_URL_KEYSPACES
Kinesis	k	AWS_ENDPOINT_URL_KINESIS

serviceId	공유 서비스 디폴트 사용자 서비스 별칭 AWS_ENDPOINT_URL_<SERVICE> 환경 변수
Kinesis Video Archived Media	k: AWS_ENDPOINT_URL_KINESIS_VIDEO_ARCHIVED_MEDIA
Kinesis Video Media	k: AWS_ENDPOINT_URL_KINESIS_VIDEO_MEDIA
Kinesis Video Signaling	k: AWS_ENDPOINT_URL_KINESIS_VIDEO_SIGNALING
Kinesis Video WebRTC Storage	k: AWS_ENDPOINT_URL_KINESIS_VIDEO_WEBRTC_STORAGE
Kinesis Analytics	k: AWS_ENDPOINT_URL_KINESIS_ANALYTICS

serviceId	공유 서비스별 환경 변수
Kinesis Analytics V2	k: AWS_ENDPOINT_URL_KINESIS_ANALYTICS_V2 n: v:
Kinesis Video	k: AWS_ENDPOINT_URL_KINESIS_VIDEO i:
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_ _I SERVICE _:

serviceId	공공 AWS_ENDPOINT_URL_<SERVICE> 환경 변수 유니코드 파일 시스템 별칭
Lex Runtime Service	1: AWS_ENDPOINT_URL_LEX_RUNTIME_SERVICE
Lex Models V2	1: AWS_ENDPOINT_URL_LEX_MODELS_V2
Lex Runtime V2	1: AWS_ENDPOINT_URL_LEX_RUNTIME_V2
License Manager	1: AWS_ENDPOINT_URL_LICENSE_MANAGER
License Manager Linux Subscriptions	1: AWS_ENDPOINT_URL_LICENSE_MANAGER_LINUX_SUBSCRIPTIONS
License Manager User Subscriptions	1: AWS_ENDPOINT_URL_LICENSE_MANAGER_USER_SUBSCRIPTIONS

serviceId	공공 유망 서비스 디폴트 서비스별 자카	AWS_ENDPOINT_URL_<SERVICE> 환경 변수
Lightsail	l:	AWS_ENDPOINT_URL_LIGHTSAIL
Location	l:	AWS_ENDPOINT_URL_LOCATION
CloudWatch Logs	c: h_	AWS_ENDPOINT_URL_CLOUDWATCH_LOGS
CloudWatch Logs	c: h_	AWS_ENDPOINT_URL_CLOUDWATCH_LOGS
LookoutEquipment	l: u:	AWS_ENDPOINT_URL_LOOKOUTEQUIPMENT
LookoutMetrics	l: t:	AWS_ENDPOINT_URL_LOOKOUTMETRICS
LookoutVision	l: s:	AWS_ENDPOINT_URL_LOOKOUTVISION
m2	m:	AWS_ENDPOINT_URL_M2

serviceId	공용 AWS_ENDPOINT_URL_<SERVICE> 환경 변수 유니코드 파일의 서비스별 자크
Machine Learning	m: AWS_ENDPOINT_URL_MACHINE_LEARNING e:
Macie2	m: AWS_ENDPOINT_URL_MACIE2
ManagedBlockchain	m: AWS_ENDPOINT_URL_MANAGEDBLOCKCHAIN o:
ManagedBlockchain Query	m: AWS_ENDPOINT_URL_MANAGEDBLOCKCHAIN_QUERY o: q:
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:

serviceId	공유 서비스별 환경 변수
Marketplace Entitlement Service	m: AWS_ENDPOINT_URL_MARKETPLACE_ENTITLEMENT_SERVICE c: e: v:
Marketplace Commerce Analytics	m: AWS_ENDPOINT_URL_MARKETPLACE_COMMERCIAL_ANALYTICS c: i:
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:

serviceId	공공 유식 디파일의 서브스별자	AWS_ENDPOINT_URL_<SERVICE> 환경 변수
MediaPackage Vod	m: a	AWS_ENDPOINT_URL_MEDIAPACKAGE_VOD
MediaPackageV2	m: a	AWS_ENDPOINT_URL_MEDIAPACKAGEV2
MediaStore	m: e	AWS_ENDPOINT_URL_MEDIASTORE
MediaStore Data	m: e_	AWS_ENDPOINT_URL_MEDIASTORE_DATA
MediaTailor	m: o:	AWS_ENDPOINT_URL_MEDIATAILOR
Medical Imaging	m: m:	AWS_ENDPOINT_URL_MEDICAL_IMAGING
MemoryDB	m:	AWS_ENDPOINT_URL_MEMORYDB
Marketplace Metering	m: c: n:	AWS_ENDPOINT_URL_MARKETPLACE_METERING

serviceId	공유 서비스별 자카	AWS_ENDPOINT_URL_<SERVICE> 환경 변수
Migration Hub	m-	AWS_ENDPOINT_URL_MIGRATION_HUB
mgn	m-	AWS_ENDPOINT_URL_MGN
Migration Hub Refactor Spaces	m-	AWS_ENDPOINT_URL_MIGRATION_HUB_REFAC TOR_SPACES
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

serviceId	공용 서비스 디폴트 엔드포인트 URL	환경 변수
mq	mq	AWS_ENDPOINT_URL_MQ
MTurk	mt	AWS_ENDPOINT_URL_MTURK
MWAA	mw	AWS_ENDPOINT_URL_MWAA
Neptune	ne	AWS_ENDPOINT_URL_NEPTUNE
Neptune Graph	ne	AWS_ENDPOINT_URL_NEPTUNE_GRAPH
neptunedata	ne	AWS_ENDPOINT_URL_NEPTUNEDATA
Network Firewall	nf	AWS_ENDPOINT_URL_NETWORK_FIREWALL
NetworkManager	nm	AWS_ENDPOINT_URL_NETWORKMANAGER
NetworkMonitor	nm	AWS_ENDPOINT_URL_NETWORKMONITOR
nimble	nm	AWS_ENDPOINT_URL_NIMBLE

serviceId	공 유 시 디 파 일 으 서 비 스 스 별 자 크	환경 변수
OAM	o:	AWS_ENDPOINT_URL_OAM
OmicS	or	AWS_ENDPOINT_URL_OMICS
OpenSearch	o: h	AWS_ENDPOINT_URL_OPENSEARCH
OpenSearchServerless	o: h: s:	AWS_ENDPOINT_URL_OPENSEARCHSERVERLESS
OpsWorks	o:	AWS_ENDPOINT_URL_OPSWORKS
OpsWorksCM	o: m	AWS_ENDPOINT_URL_OPSWORKSCM
Organizations	o: i:	AWS_ENDPOINT_URL_ORGANIZATIONS
OSIS	o:	AWS_ENDPOINT_URL_OSIS
Outposts	o:	AWS_ENDPOINT_URL_OUTPOSTS
p8data	p:	AWS_ENDPOINT_URL_P8DATA

serviceId	공유 서비스 디폴트 엔드포인트 URL 별칭	AWS_ENDPOINT_URL_<SERVICE> 환경 변수
p8data	p8data	AWS_ENDPOINT_URL_P8DATA
Panorama	panorama	AWS_ENDPOINT_URL_PANORAMA
Payment Cryptography	payment-cryptography	AWS_ENDPOINT_URL_PAYMENT_CRYPTOGRAPHY
Payment Cryptography Data	payment-cryptography-data	AWS_ENDPOINT_URL_PAYMENT_CRYPTOGRAPHY_DATA
Pca Connector Ad	pca-connector-ad	AWS_ENDPOINT_URL_PCA_CONNECTOR_AD
Personalize	personalize	AWS_ENDPOINT_URL_PERSONALIZE
Personalize Events	personalize-events	AWS_ENDPOINT_URL_PERSONALIZE_EVENTS

serviceId	공용 AWS_ENDPOINT_URL_<SERVICE> 환경 변수 유니코드 파일의 서브시스템 별칭 자크
Personalize Runtime	p: AWS_ENDPOINT_URL_PERSONALIZE_RUNTIME
PI	p: AWS_ENDPOINT_URL_PI
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

serviceId	공유 서비스별 환경 변수
Pricing	p: AWS_ENDPOINT_URL_PRICING
PrivateNetworks	p: AWS_ENDPOINT_URL_PRIVATENETWORKS t:
Proton	p: AWS_ENDPOINT_URL_PROTON
QBusiness	q: AWS_ENDPOINT_URL_QBUSINESS
QConnect	q: AWS_ENDPOINT_URL_QCONNECT
QLDB	q: AWS_ENDPOINT_URL_QLDB
QLDB Session	q: AWS_ENDPOINT_URL_QLDB_SESSION i:
QuickSight	q: AWS_ENDPOINT_URL_QUICKSIGHT t:
RAM	r: AWS_ENDPOINT_URL_RAM
rbin	r: AWS_ENDPOINT_URL_RBIN

serviceId	공유 서비스별 엔드포인트 URL	환경 변수
RDS	<code>AWS_ENDPOINT_URL_RDS</code>	
RDS Data	<code>AWS_ENDPOINT_URL_RDS_DATA</code>	
Redshift	<code>AWS_ENDPOINT_URL_REDSHIFT</code>	
Redshift Data	<code>AWS_ENDPOINT_URL_REDSHIFT_DATA</code>	
Redshift Serverless	<code>AWS_ENDPOINT_URL_REDSHIFT_SERVERLESS</code>	
Rekognition	<code>AWS_ENDPOINT_URL_REKOGNITION</code>	
repostspace	<code>AWS_ENDPOINT_URL_REPOSTSPACE</code>	
resiliencehub	<code>AWS_ENDPOINT_URL_RESILIENCEHUB</code>	

serviceId	공용 AWS_ENDPOINT_URL_<SERVICE> 환경 변수 유니코드 파일의 서비스별 자카
Resource Explorer 2	r: AWS_ENDPOINT_URL_RESOURCE_EXPLORER_2 e: 2
Resource Groups	r: AWS_ENDPOINT_URL_RESOURCE_GROUPS g:
Resource Groups Tagging API	r: AWS_ENDPOINT_URL_RESOURCE_GROUPS_TAG g: GING_API g:
RoboMaker	r: AWS_ENDPOINT_URL_ROBOMAKER
RolesAnywhere	r: AWS_ENDPOINT_URL_ROLESEANYWHERE h:
Route 53	r: AWS_ENDPOINT_URL_ROUTE_53
Route53 Recovery Cluster	r: AWS_ENDPOINT_URL_ROUTE53_RECOVERY_CLUSTER e: l:

serviceId	공용 AWS_ENDPOINT_URL_<SERVICE> 환경 변수 유니코드 파일의 서브시스템별 자크
Route53 Recovery Control Config	i: AWS_ENDPOINT_URL_ROUTE53_RECOVERY_CONTROL_CONFIG
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

serviceId	공용 AWS_ENDPOINT_URL_<SERVICE> 환경 변수
S3Outposts	s: AWS_ENDPOINT_URL_S3OUTPOSTS
SageMaker	s: AWS_ENDPOINT_URL_SAGEMAKER
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

serviceId	공유 서비스별 환경 변수
SageMaker Metrics	s: AWS_ENDPOINT_URL_SAGEMAKER_METRICS
SageMaker Runtime	s: AWS_ENDPOINT_URL_SAGEMAKER_RUNTIME
savingsplans	s: AWS_ENDPOINT_URL_SAVINGSPLANS
Scheduler	s: AWS_ENDPOINT_URL_SCHEDULER
schemas	s: AWS_ENDPOINT_URL_SCHEMAS
SimpleDB	s: AWS_ENDPOINT_URL_SIMPLEDB
Secrets Manager	s: AWS_ENDPOINT_URL_SECRETS_MANAGER
SecurityHub	s: AWS_ENDPOINT_URL_SECURITYHUB

serviceId	공유 서비스별 환경 변수
SecurityLake	AWS_ENDPOINT_URL_SECURITYLAKE
ServerlessApplicationRepository	AWS_ENDPOINT_URL_SERVERLESSAPPLICATIONREPOSITORY
Service Quotas	AWS_ENDPOINT_URL_SERVICE_QUOTAS
Service Catalog	AWS_ENDPOINT_URL_SERVICE_CATALOG
Service Catalog AppRegistry	AWS_ENDPOINT_URL_SERVICE_CATALOG_APPREGISTRY
ServiceDiscovery	AWS_ENDPOINT_URL_SERVICEDISCOVERY
SES	AWS_ENDPOINT_URL_SES

serviceId	공유 서비스별 환경 변수
SESV2	se: AWS_ENDPOINT_URL_SESV2
Shield	sh: AWS_ENDPOINT_URL_SHIELD
signer	s: AWS_ENDPOINT_URL_SIGNER
SimSpaceWeaver	s: AWS_ENDPOINT_URL_SIMSPACEWEAVER ei:
SMS	sr: AWS_ENDPOINT_URL_SMS
Snow Device Management	sd: AWS_ENDPOINT_URL_SNOW_DEVICE_MANAGEMENT cm:
Snowball	sb: AWS_ENDPOINT_URL_SNOWBALL
SNS	sn: AWS_ENDPOINT_URL_SNS
SQS	sq: AWS_ENDPOINT_URL_SQS
SSM	ss: AWS_ENDPOINT_URL_SSM

serviceId	공용 AWS_ENDPOINT_URL_<SERVICE> 환경 변수 유니코드 파일의 서비스별 자크
SSM Contacts	s: AWS_ENDPOINT_URL_SSM_CONTACTS c1
SSM Incidents	s: AWS_ENDPOINT_URL_SSM_INCIDENTS ei
Ssm Sap	s: AWS_ENDPOINT_URL_SSM_SAP
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 a1
STS	s: AWS_ENDPOINT_URL_STS
SupplyChain	s: AWS_ENDPOINT_URL_SUPPLYCHAIN i1

serviceId	공유 서비스별 엔드포인트 URL 환경 변수
Support	SI_AWS_ENDPOINT_URL_SUPPORT
Support App	SI_AWS_ENDPOINT_URL_SUPPORT_APP
SWF	SI_AWS_ENDPOINT_URL_SWF
synthetics	SI_AWS_ENDPOINT_URL_SYNTHETICS
Textract	TI_AWS_ENDPOINT_URL_TEXTRACT
Timestream InfluxDB	TI_AWS_ENDPOINT_URL_TIMESTREAM_INFLUXDB
Timestream Query	TI_AWS_ENDPOINT_URL_TIMESTREAM_QUERY
Timestream Write	TI_AWS_ENDPOINT_URL_TIMESTREAM_WRITE
tnb	TI_AWS_ENDPOINT_URL_TNB

serviceId	공용 서비스별 엔드포인트 URL 환경 변수
Transcribe	t: AWS_ENDPOINT_URL_TRANSCRIBE
Transfer	t: AWS_ENDPOINT_URL_TRANSFER
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

serviceId	공 유 시 디 파 일 으 서 비 스 스 별 자 크	환경 변수
WAF Regional	w:	AWS_ENDPOINT_URL_WAF_REGIONAL
WAFV2	w:	AWS_ENDPOINT_URL_WAFV2
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

serviceId	공유 시스템 속성: 자바/코틀린 전용	환경 변수
WorkSpaces Thin Client	wc	AWS_ENDPOINT_URL_WORKSPACES_THIN_CLIENT
WorkSpaces Web	wc	AWS_ENDPOINT_URL_WORKSPACES_WEB
XRay	x:	AWS_ENDPOINT_URL_XRAY

스마트 구성 기본값

스마트 구성 기본값 기능을 통해 AWS SDK는 다른 구성 설정에 대해 미리 정의되고 최적화된 기본값을 제공할 수 있습니다.

다음을 사용하여 이 기능을 구성하십시오.

defaults_mode- 공유 파일 설정 **AWS config**, **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. AWS SDK 설정에 명시적으로 설정된 값이 있는 경우 해당 값이 항상 우선합니다. AWS SDK 설정에 명시적으로 설정된 값이 없고 기존 설정과 **defaults_mode** 동일하지 않은 경우 이 기능은 애플리케이션에 최적화된 다양한 설정에 대해 다른 기본값을 제공할

수 있습니다. 설정에는 HTTP 통신 설정, 재시도 동작, 서비스 리전 엔드포인트 설정 및 잠재적으로 모든 SDK 관련 구성이 포함될 수 있습니다. 이 기능을 사용하는 고객은 일반 사용 시나리오에 맞게 조정된 새 구성 기본값을 얻을 수 있습니다. `defaults_mode`이 `legacy`가 아닌 경우에 SDK를 업그레이드할 때 사용자 애플리케이션 테스트를 수행하는 것이 좋습니다. 제공된 기본값은 모범 사례가 발전함에 따라 변경될 수 있기 때문입니다.

기본값: `legacy`

참고: 새 주요 버전의 SDK는 `standard`로 기본 설정됩니다.

유효값:

- `legacy`— SDK에 따라 달라지고 `defaults_mode`의 설정 이전에 존재했던 기본 설정을 제공합니다.
- `standard`— 대부분의 시나리오에서 안전하게 실행할 수 있는 최신 권장 기본값을 제공합니다.
- `in-region`— 표준 모드를 기반으로 하며 동일한 모드에서 호출하는 애플리케이션에 맞게 AWS 서비스 조정된 최적화를 포함합니다. [AWS 리전](#)
- `cross-region`— 표준 모드를 기반으로 하며 다른 AWS 서비스 지역에서 호출하는 애플리케이션에 맞게 조정된 최적화를 포함합니다.
- `mobile`— 표준 모드를 기반으로 구축하며 모바일 애플리케이션에 맞게 조정된 최적화를 포함합니다.
- `auto`— 표준 모드를 기반으로 구축하며 실험적 기능을 포함합니다. SDK는 런타임 환경을 검색하여 적절한 설정을 자동으로 결정합니다. 자동 감지는 휴리스틱 기반이며 정확도가 100%는 아닙니다. 런타임 환경을 확인할 수 없는 경우 `standard` 모드가 사용됩니다. 자동 탐지는 [인스턴스 메타데이터와 사용자 데이터](#)를 쿼리할 수 있으며, 이로 인해 지연이 발생할 수 있습니다. 시차 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 재시도 방법을 지정합니다. [재시도 동작](#) 섹션을 참조하십시오.
- `stsRegionalEndpoints`— SDK가 AWS Security Token Service (AWS STS)와 통신하는 데 사용하는 AWS 서비스 엔드포인트를 결정하는 방법을 지정합니다. [AWS STS 지역화된 엔드포인트](#) 섹션을 참조하십시오.

- `s3UsEast1RegionalEndpoints`— SDK가 해당 `us-east-1` 지역의 Amazon S3와 통신하는데 사용하는 AWS 서비스 엔드포인트를 결정하는 방법을 지정합니다.
- `connectTimeoutInMillis`— 소켓에서 처음 연결을 시도한 후 제한 시간이 초과되기까지의 시간. 클라이언트가 연결 핸드셰이크 완료를 수신하지 못하면 클라이언트는 작업을 포기하고 중단합니다.
- `tlsNegotiationTimeoutInMillis`— CLIENT HELLO 메시지가 전송된 시점부터 클라이언트와 서버가 암호를 완전히 협상하고 키를 교환할 때까지 TLS 핸드셰이크에 걸릴 수 있는 최대 시간입니다.

각 설정의 기본값은 사용자 응용 프로그램에서 선택한 `defaults_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).

SDK와의 AWS 호환성

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

SDK	지원	참고 또는 추가 정보
AWS CLI v2	아니요	
SDK for C++	예	최적화되지 않은 매개변수: <code>stsRegionalEndpoints</code> , <code>s3UsEast1RegionalEndpoints</code> , <code>tlsNegotiationTimeoutInMillis</code> .
SDK for Go V2 (1.x)	예	최적화되지 않은 매개변수: <code>stsRegionalEndpoints</code> , <code>s3UsEast1RegionalEndpoints</code> , <code>retryMode</code> .
SDK for Go 1.x (V1)	아니요	
SDK for Java 2.x	예	최적화되지 않은 매개변수: <code>stsRegionalEndpoints</code> .
SDK for Java 1.x	아니요	
3.x용 SDK JavaScript	예	최적화되지 않은 매개변수: <code>stsRegionalEndpoints</code> , <code>s3UsEast1RegionalEndpoints</code> , <code>tlsNegotiationTimeoutInMillis</code> , <code>connectTi</code> .

SDK	지원	참고 또는 추가 정보
		<code>meoutInMillis</code> 가 <code>connectionTimeout</code> 호출됩니다.
2.x용 SDK JavaScript	아니요	
SDK for Kotlin	아니요	
SDK for .NET 3.x	예	최적화되지 않은 매개변수: <code>connectTimeoutInMillis</code> , <code>tlsNegotiationTimeoutInMillis</code> .
SDK for PHP 3.x	예	최적화되지 않은 매개변수: <code>tlsNegotiationTimeoutInMillis</code> .
SDK for Python (Boto3)	예	최적화되지 않은 매개변수: <code>tlsNegotiationTimeoutInMillis</code> .
SDK for Ruby 3.x	예	
SDK for Rust	아니요	
다음에 위한 도구 PowerShell	예	최적화되지 않은 매개변수: <code>connectTimeoutInMillis</code> , <code>tlsNegotiationTimeoutInMillis</code> .

AWS공용 런타임 (CRT) 라이브러리

AWS공용 런타임 (CRT) 라이브러리는 SDK의 기본 라이브러리입니다. CRT는 C로 작성된 독립 패키지의 모듈식 제품군으로, 각 패키지는 우수한 성능을 제공하고 다양한 필수 기능을 위한 최소한의 설치 공간을 제공합니다. 이러한 기능은 모든 SDK에서 공통적이며 공유되므로 코드 재사용, 최적화 및 정확성이 향상됩니다. 패키지는 다음과 같습니다.

- [awslabs/aws-c-auth](#): AWS 클라이언트 측 인증 (표준 보안 인증 공급자 및 서명 (sigv4))
- [awslabs/aws-c-cal](#): 암호화 프리미티브 유형, 해시 (MD5, SHA256, SHA256 HMAC), 서명자, AES
- [awslabs/aws-c-common](#): 기본 데이터 구조, 스레딩/동기화 프리미티브 유형, 버퍼 관리, stdlib 관련 함수
- [awslabs/aws-c-compression](#): 압축 알고리즘 (허프만 인코딩/디코딩)
- [awslabs/aws-c-event-stream](#): 이벤트 스트림 메시지 처리 (헤더, 프렐루드, 페이로드, crc/trailer), 이벤트 스트림을 통한 원격 프로시저 호출 (RPC) 구현
- [awslabs/aws-c-http](#): HTTP/1.1 및 HTTP/2 사양의 C99 구현
- [awslabs/aws-c-io](#): 소켓 (TCP, UDP), DNS, 파이프, 이벤트 루프, 채널, SSL/TLS
- [awslabs/aws-c-iot](#): 디바이스와의 AWS IoT 클라우드 서비스 통합의 C99 구현
- [awslabs/aws-c-mqtt](#): 사물 인터넷 (IoT) 을 위한 간단한 표준 메시징 프로토콜
- [awslabs/aws-c-s3](#): Amazon S3 서비스와의 통신을 위한 C99 라이브러리 구현으로, 고대역폭 Amazon EC2 인스턴스의 처리량을 극대화하도록 설계되었습니다.
- [awslabs/aws-c-sdkutils](#): AWS 프로필 파싱 및 관리를 위한 유틸리티 라이브러리
- [awslabs/aws-checksums](#): 효율적인 소프트웨어 구현으로 대체되는 크로스 플랫폼 하드웨어 가속 CRC32c 및 CRC32
- [awslabs/aws-lc](#): Google BoringSSL 프로젝트 및 OpenSSL 프로젝트의 코드를 기반으로 AWS 암호화 팀에서 AWS 및 고객을 위해 유지 관리하는 범용 암호화 라이브러리
- [awslabs/s2n](#): TLS/SSL 프로토콜의 C99 구현, 보안을 최우선으로 하여 작고 빠르도록 설계되었습니다.

CRT는 Go를 제외한 모든 SDK를 통해 사용할 수 있습니다.

CRT 종속성

CRT 라이브러리는 복잡한 관계와 종속성 네트워크를 형성합니다. 소스에서 직접 CRT를 구축해야 하는 경우 이러한 관계를 아는 것이 도움이 됩니다. 하지만 대부분의 사용자는 자신의 언어 SDK (예: C++용 SDK 또는 Java용 AWS SDK) 또는 해당 언어의 IoT 장치 SDK AWS (예: AWSC++용 IoT SDK 또는 Java용 AWSIoT SDK) 를 통해 CRT 기능에 액세스합니다. 다음 다이어그램에서 언어 CRT 바인딩 상자는 특정 언어 SDK의 CRT 라이브러리를 래핑하는 패키지를 나타냅니다. 이 패키지는 다음과 같은 형식의 `aws-crt-*` 패키지 모음입니다. 여기서 '*'는 SDK 언어 (예: [aws-crt-cpp](#) 혹은 [aws-crt-java](#)) 입니다.

다음은 CRT 라이브러리의 계층적 종속성을 보여줍니다.

AWS 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 매트릭스](#)

AWS SDK 및 툴 버전 지원 매트릭스

아래 표에는 사용 가능한 AWS 소프트웨어 개발 키트 (SDK) 메이저 버전 목록과 해당 버전이 유지 관리 수명 주기 내 어느 단계에 속하는지 관련 일정과 함께 나와 있습니다. 주요 버전의 AWS SDK 및 도구의 수명 주기와 기본 종속성에 대한 자세한 내용은 [유지 관리 정책](#) 을 참조하십시오.

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	
SDK for C++	1.x	정식 출시	2015년 9월 2일	
SDK for Go V2	V2 1.x	정식 출시	2021년 1월 19일	
SDK for Go	1.x	유지 관리 공지	2015년 11월 19일	세부 정보 및 날짜는 공지를 참조 하십시오.
SDK for Java	1.x	유지 관리 공지	2010년 3월 25일	세부 정보 및 날짜는 공지를 참조 하십시오.
SDK for Java	2.x	정식 출시	11/20/2018	
SDK는 다음과 같습니다. JavaScript	1.x	지원 종료	2013년 5월 6일	
SDK에 대한 JavaScript	2.x	유지 관리 공지	2014년 6월 19일	세부 정보 및 날짜는 공지를 참조 하십시오.
SDK는 다음과 같습니다. JavaScript	3.x	정식 출시	2020년 12월 15일	

SDK	메이저 버전	현재 단계	일반 출시 날짜	주석
SDK for Kotlin	1.x	정식 출시	2023년 11월 27일	
SDK for .NET	1.x	지원 종료	2009년 11월	
SDK for .NET	2.x	지원 종료	2013년 11월 8일	
SDK for .NET	3.x	정식 출시	2015년 7월 28일	
SDK for PHP	2.x	지원 종료	2012년 11월 2일	
SDK for PHP	3.x	정식 출시	2015년 5월 27일	
SDK for Python (Boto2)	1.x	지원 종료	2011년 7월 13일	
SDK for Python (Boto3)	1.x	정식 출시	2015년 6월 22일	
SDK for Python (Botocore)	1.x	정식 출시	2015년 6월 22일	
SDK for Ruby	1.x	지원 종료	2011년 7월 14일	
SDK for Ruby	2.x	지원 종료	2015년 2월 15일	
SDK for Ruby	3.x	정식 출시	2017년 8월 29일	
SDK for Rust	1.x	정식 출시	2023년 11월 27일	
SDK for Swift	1.x	개발자 미리 보기		
다음에 위한 도구 PowerShell	2.x	지원 종료	2013년 11월 8일	
다음에 위한 도구 PowerShell	3.x	지원 종료	2015년 7월 29일	

SDK	메이저 버전	현재 단계	일반 출시 날짜	주석
다음에 위한 도구 PowerShell	4.x	정식 출시	2019년 11월 21일	

AWS SDK 및 도구용 문서 기록 참조 가이드

다음 표에는 AWS SDK 및 도구 참조 가이드의 중요한 추가 및 업데이트가 설명되어 있습니다. 이 설명서에 대한 업데이트 알림을 받으려면 RSS 피드에 가입하면 됩니다.

변경 사항	설명	날짜
Java 1.x 시스템 속성용 SDK	1.x에서 지원되는 JVM 시스템 구성 설정에 대한 세부 정보를 추가합니다. AWS SDK for Java	2024년 5월 30일
설정 업데이트	JVM 시스템 구성 설정을 추가합니다.	2024년 3월 27일
호환성 표 업데이트	SDK 지원 호환성 업데이트, IAM ID 센터 절차 업데이트.	2024년 2월 20일
컨테이너 보안 인증 업데이트. IMDS 업데이트.	Amazon EKS 지원 추가. IMDSv1 폴백을 비활성화하기 위한 설정 추가.	2023년 12월 29일
요청 압축	요청 압축 기능에 대한 설정 추가.	2023년 12월 27일
호환성 테이블	SDK 및 도구 기능에 대한 호환성 테이블이 SDK for Kotlin, SDK for Rust, AWS Tools for PowerShell등을 포함하도록 업데이트되었습니다.	2023년 12월 10일
인증 업데이트	SDK 및 도구에 지원되는 인증 방법이 업데이트되었습니다.	2023년 7월 1일
IAM 모범 사례 업데이트	IAM 모범 사례에 따라 가이드가 업데이트되었습니다. 자세한 내용은 IAM의 보안 모범 사례 를 참조하십시오.	2023년 2월 27일

SSO 업데이트	새 SSO 토큰 구성을 위한 SSO 보안 인증 업데이트.	2022년 11월 19일
설정 업데이트	일반 구성 및 Amazon S3 다중 리전 액세스 포인트 지원 표 업데이트.	2022년 11월 17일
설정 업데이트	IMDS 클라이언트 및 IMDS 보안 인증의 명확성 업데이트. 환경 변수 업데이트.	2022년 11월 4일
시작 페이지 업데이트	아마존 CodeWhisperer 발표.	2022년 9월 22일
싱글 사인은 서비스 이름 변경	이제 AWS SSO를 다음과 같이 지칭한다는 점을 반영하여 업데이트되었습니다. 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.