



Panduan Referensi

AWS SDKs dan Alat



AWS SDKsdan Alat: Panduan Referensi

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

Merek dagang dan tampilan dagang Amazon tidak boleh digunakan sehubungan dengan produk atau layanan apa pun yang bukan milik Amazon, dengan cara apa pun yang dapat menyebabkan kebingungan di antara pelanggan, atau dengan cara apa pun yang merendahkan atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan hak milik masing-masing pemiliknya, yang mungkin atau tidak terafiliasi, terkait dengan, atau disponsori oleh Amazon.

Table of Contents

AWS SDKsdan Panduan Referensi Alat	1
Sumber daya pengembang	2
Pemberitahuan telemetri toolkit	3
Konfigurasi	4
Berbagi config dan credentials file	5
Profil	5
Format file konfigurasi	7
Format file kredensial	10
Lokasi file bersama	11
Resolusi direktori rumah	11
Ubah lokasi default file-file ini	12
Variabel-variabel lingkungan	13
Cara mengatur variabel lingkungan	13
Pengaturan variabel lingkungan tanpa server	14
Properti sistem JVM	15
Cara mengatur properti sistem JVM	15
Otentikasi dan akses	18
ID AWS Builder	20
IAMOtentikasi Pusat Identitas	20
Konfigurasi akses terprogram menggunakan IAM Identity Center	21
Memahami autentikasi Pusat Identitas IAM	24
Peran IAM Di Mana Saja	28
Langkah 1: Konfigurasi Peran IAM Di Mana Saja	28
Langkah 2: Gunakan Peran IAM Di Mana Saja	29
Asumsikan peran	30
Asumsikan IAM peran	30
Asumsikan peran (web)	32
Bersekutu dengan identitas web atau OpenID Connect	32
AWS kunci akses	34
Gunakan kredensial jangka pendek	34
Gunakan kredensi jangka panjang	34
Kredensi jangka pendek	36
Kredensi jangka panjang	37
IAMperan untuk EC2 contoh	41

Buat IAM peran	41
Luncurkan EC2 instans Amazon dan tentukan IAM peran Anda	41
Connect ke EC2 instance	42
Jalankan aplikasi Anda pada EC2 instance	42
Referensi pengaturan	44
Membuat klien layanan	44
Prioritas pengaturan	44
Halaman pengaturan	46
Configdaftar pengaturan file	47
Credentialsdaftar pengaturan file	51
Daftar variabel lingkungan	51
JVMdaftar properti sistem	55
Penyedia kredensi standar	58
Memahami rantai penyedia kredensi	59
SDK-rantai penyedia kredensi khusus dan khusus alat	60
AWS kunci akses	61
Asumsikan penyedia peran	64
Penyedia kontainer	71
IAMPenyedia Pusat Identitas	74
IMDSpenyedia	80
Penyedia proses	85
Fitur standar	89
Titik akhir berbasis akun	90
ID Aplikasi	92
EC2Metadata contoh Amazon	94
Titik akses Amazon S3	96
Titik Akses Multi-Wilayah Amazon S3	98
Wilayah AWS	100
AWS STS Titik akhir regional	103
Tumpukan ganda dan titik akhir FIPS	107
Penemuan titik akhir	110
Konfigurasi umum	112
IMDSklien	115
Coba lagi perilaku	118
Minta kompresi	124
Titik akhir khusus layanan	126

Default konfigurasi cerdas	170
Runtime Umum	175
CRTdependensi	176
Kebijakan pemeliharaan	177
Gambaran Umum	177
Penentuan Versi	177
Siklus hidup versi utama SDK	177
Siklus hidup ketergantungan	178
Metode komunikasi	179
Dukungan versi	180
Riwayat dokumen	183
AWSGlosarium	186
.....	clxxxvii

AWS SDKs dan Panduan Referensi Alat

Banyak SDKs dan alat berbagi beberapa fungsi umum, baik melalui spesifikasi desain bersama atau melalui perpustakaan bersama.

Panduan ini mencakup informasi mengenai:

- [Konfigurasi](#)— Cara menggunakan variabel bersama `config` dan `credentials` file atau lingkungan untuk mengkonfigurasi Anda AWS SDKs dan alat.
- [Otentikasi dan akses](#)— Tetapkan bagaimana kode atau alat Anda mengotentikasi AWS saat Anda mengembangkan. Layanan AWS
- [Referensi pengaturan](#)— Referensi untuk semua pengaturan standar yang tersedia untuk otentikasi dan konfigurasi.
- [AWS Pustaka Runtime \(CRT\) umum](#)— Ikhtisar pustaka AWS Common Runtime (CRT) bersama yang tersedia untuk hampir semua. SDKs
- [AWS Kebijakan pemeliharaan SDK dan Alat](#) mencakup kebijakan pemeliharaan dan pembuatan versi untuk Kit Pengembangan AWS Perangkat Lunak (SDKs) dan alat, termasuk Mobile dan Internet of Things (IoT) SDKs, dan dependensi yang mendasarinya.

Panduan Referensi Alat AWS SDKs dan Alat ini dimaksudkan untuk menjadi basis informasi yang berlaku untuk beberapa SDKs dan alat. Panduan khusus untuk SDK atau alat yang Anda gunakan harus digunakan selain informasi yang disajikan di sini. Berikut ini adalah SDK dan alat yang memiliki bagian materi yang relevan dalam panduan ini:

Jika Anda menggunakan:	Bagian yang relevan dari panduan ini untuk Anda adalah:
<ul style="list-style-type: none"> • Apa saja SDK atau alat 	AWS Kebijakan pemeliharaan SDK dan Alat
<ul style="list-style-type: none"> • AWS Cloud9 • AWS CDK • AWS Toolkit for Azure DevOps • AWS Toolkit for JetBrains • AWS Toolkit for Visual Studio • AWS Toolkit for Visual Studio Code 	Konfigurasi Otentikasi dan akses AWS Kebijakan pemeliharaan SDK dan Alat

Jika Anda menggunakan:	Bagian yang relevan dari panduan ini untuk Anda adalah:
<ul style="list-style-type: none"> • AWS Serverless Application Model • AWS CodeArtifact • AWS CodeBuild • Amazon CodeCatalyst • AWS CodeCommit • AWS CodeDeploy • AWS CodePipeline 	
<ul style="list-style-type: none"> • AWS CLI • AWS SDK for C++ • AWS SDK for Go • AWS SDK for Java • AWS SDK for JavaScript • AWS SDK for Kotlin • AWS SDK for .NET • AWS SDK for PHP • AWS SDK for Python (Boto3) • AWS SDK for Ruby • AWS SDK for Rust • AWS SDK for Swift • AWS Tools for Windows PowerShell 	<ul style="list-style-type: none"> • Konfigurasi • Otentikasi dan akses • Referensi pengaturan • AWS Pustaka Runtime (CRT) umum • AWS Kebijakan pemeliharaan SDK dan Alat • AWS SDKsdan dukungan versi Tools

Sumber daya pengembang

Untuk ikhtisar alat yang dapat membantu Anda mengembangkan aplikasi AWS, lihat [Alat untuk Dibangun AWS](#). Untuk informasi tentang dukungan, lihat [Pusat AWS Pengetahuan](#).

Amazon Q Developer adalah asisten percakapan bertenaga AI generatif yang dapat membantu Anda memahami, membangun, memperluas, dan mengoperasikan aplikasi. AWS Untuk mempercepat pembangunan Anda AWS, model yang mendukung Amazon Q ditambah dengan AWS konten

berkualitas tinggi untuk menghasilkan jawaban yang lebih lengkap, dapat ditindaklanjuti, dan direferensikan. Untuk informasi selengkapnya, lihat [Apa itu Pengembang Amazon Q?](#) di Panduan Pengguna Pengembang Amazon Q.

Pemberitahuan telemetri toolkit

AWS Integrated Development Environment (IDE) Toolkit adalah plugin dan ekstensi yang memungkinkan akses ke AWS layanan di Anda. IDE IDEPlugin dan ekstensi Amazon Q memungkinkan bantuan AI generatif di Anda. IDE Untuk informasi rinci tentang masing-masing IDE Toolkit, lihat Panduan Pengguna Toolkit di tabel sebelumnya. Untuk mempelajari lebih lanjut tentang menggunakan Amazon Q di Anda IDE, lihat [Menggunakan Amazon Q dalam IDE](#) topik di panduan pengembang Amazon Q.

AWS IDE Toolkit dan Amazon Q dapat mengumpulkan dan menyimpan data telemetri sisi klien untuk menginformasikan keputusan mengenai rilis Toolkit AWS dan Amazon Q di masa mendatang. Data yang dikumpulkan mengukur penggunaan AWS Toolkit dan Amazon Q.

Untuk mempelajari lebih lanjut tentang data telemetri yang dikumpulkan di semua AWS IDE Toolkit dan Amazon Q, lihat dokumen.json [commonDefinitions](#) di repositori Github. `aws-toolkit-common`

Untuk informasi terperinci tentang data telemetri yang dikumpulkan oleh masing-masing AWS IDE Toolkit dan ekstensi Amazon Q, rujuk dokumen sumber daya di repositori Toolkit berikut AWS :
GitHub

- [AWS Kit Alat Visual Studio dengan Amazon Q](#)
- [AWS Toolkit for Visual Studio Code dan ekstensi Amazon Q untuk VS Code](#)
- [AWS Toolkit for JetBrains dan plugin Amazon Q untuk JetBrains](#)
- [Amazon Q untuk Eclipse](#)

AWS Layanan tertentu yang dapat diakses di AWS Toolkit dapat mengumpulkan data telemetri sisi klien tambahan. Untuk informasi rinci tentang jenis data yang dikumpulkan oleh masing-masing AWS layanan, lihat topik [AWS Dokumentasi](#) untuk layanan tertentu yang Anda minati.

Konfigurasi

Dengan AWS SDK dan alat AWS pengembang lainnya, seperti AWS Command Line Interface (AWS CLI), Anda dapat berinteraksi dengan API AWS layanan. Namun, sebelum mencobanya, Anda harus mengonfigurasi SDK atau alat dengan informasi yang diperlukan untuk melakukan operasi yang diminta.

Informasi ini mencakup item-item berikut:

- Informasi kredensial yang mengidentifikasi siapa yang memanggil API. Kredensialnya digunakan untuk mengenkripsi permintaan ke server. AWS Dengan menggunakan informasi ini, AWS mengonfirmasi identitas Anda dan dapat mengambil kebijakan izin yang terkait dengannya. Kemudian dapat menentukan tindakan apa yang diizinkan untuk Anda lakukan.
- Detail konfigurasi lain yang Anda gunakan untuk memberi tahu AWS CLI atau SDK cara memproses permintaan, ke mana harus mengirim permintaan (ke titik akhir AWS layanan mana), dan cara menafsirkan atau menampilkan respons.

Setiap SDK atau alat mendukung beberapa sumber yang dapat Anda gunakan untuk menyediakan informasi kredensi dan konfigurasi yang diperlukan. Beberapa sumber unik untuk SDK atau alat, dan Anda harus merujuk ke dokumentasi untuk alat atau SDK tersebut untuk detail tentang cara menggunakan metode tersebut.

Namun, sebagian besar AWS SDK dan alat mendukung pengaturan umum dari dua sumber utama (di luar kode itu sendiri):

- File [AWS konfigurasi dan kredensial bersama — File](#) bersama `config` dan `credentials` file adalah cara paling umum untuk menentukan otentikasi dan konfigurasi ke SDK atau AWS alat. Gunakan file-file ini untuk menyimpan pengaturan yang dapat digunakan alat dan aplikasi Anda. Pengaturan dalam `credentials` file bersama `config` dan dikaitkan dengan profil tertentu. Dengan beberapa profil, Anda dapat membuat konfigurasi pengaturan yang berbeda untuk diterapkan dalam skenario yang berbeda. Saat Anda menggunakan AWS alat untuk menjalankan perintah atau menggunakan SDK untuk menjalankan AWS API, Anda dapat menentukan profil mana, dan dengan demikian pengaturan konfigurasi mana, yang akan digunakan untuk tindakan itu. Salah satu profil ditetapkan sebagai default profil dan digunakan secara otomatis ketika Anda tidak secara eksplisit menentukan profil yang akan digunakan. Pengaturan yang dapat Anda simpan dalam file-file ini didokumentasikan dalam panduan referensi ini.

- [Variabel lingkungan](#) — Beberapa pengaturan dapat disimpan dalam variabel lingkungan sistem operasi Anda. Meskipun Anda hanya dapat memiliki satu set variabel lingkungan yang berlaku pada satu waktu, mereka mudah dimodifikasi secara dinamis saat program Anda berjalan dan kebutuhan Anda berubah.

Topik tambahan di bagian ini

- [Berbagi config dan credentials file](#)
- [Lokasi yang dibagikan config dan credentials file](#)
- [Dukungan variabel lingkungan](#)
- [Dukungan properti sistem JVM](#)

Berbagi **config** dan **credentials** file

`credentialsFile` yang dibagikan AWS `config` dan berisi satu set profil. Profil adalah seperangkat pengaturan konfigurasi, dalam pasangan kunci-nilai, yang digunakan oleh AWS Command Line Interface (AWS CLI), the AWS SDKs, dan alat lainnya. Nilai konfigurasi dilampirkan ke profil untuk mengonfigurasi beberapa aspek SDK /tool saat profil itu digunakan. File-file ini “dibagikan” karena nilainya memengaruhi aplikasi, proses, atau SDKs lingkungan lokal apa pun bagi pengguna.

Baik file bersama `config` maupun `credentials` file adalah file teks biasa yang hanya berisi ASCII karakter (UTF-8 dikodekan). Mereka mengambil bentuk apa yang umumnya disebut sebagai [INIfile](#).

Profil

Pengaturan dalam `credentials` file bersama `config` dan dikaitkan dengan profil tertentu. Beberapa profil dapat didefinisikan dalam file untuk membuat konfigurasi pengaturan yang berbeda untuk diterapkan di lingkungan pengembangan yang berbeda.

[default] Profil berisi nilai-nilai yang digunakan oleh operasi SDK atau alat jika profil bernama tertentu tidak ditentukan. Anda juga dapat membuat profil terpisah yang dapat Anda referensikan secara eksplisit berdasarkan nama. Setiap profil dapat menggunakan pengaturan dan nilai yang berbeda sesuai kebutuhan aplikasi dan skenario Anda.

Note

[default] hanyalah profil yang tidak disebutkan namanya. Profil ini dinamai `default` karena merupakan profil default yang digunakan oleh SDK jika pengguna tidak menentukan

profil. Itu tidak memberikan nilai default yang diwariskan ke profil lain. Jika Anda menyetel sesuatu di [default] profil dan Anda tidak mengaturnya di profil bernama, maka nilainya tidak disetel saat Anda menggunakan profil bernama.

Menetapkan profil bernama

[default] Profil dan beberapa profil bernama dapat ada dalam file yang sama. Gunakan pengaturan berikut untuk memilih pengaturan profil mana yang digunakan oleh Anda SDK atau alat saat menjalankan kode Anda. Profil juga dapat dipilih dalam kode, atau per-perintah saat bekerja dengan file. AWS CLI

Konfigurasi fungsionalitas ini dengan mengatur salah satu dari berikut ini:

AWS_PROFILE- variabel lingkungan

Ketika variabel lingkungan ini diatur ke profil bernama atau "default", semua SDK kode dan AWS CLI perintah menggunakan pengaturan di profil itu.

Linux/macOS contoh pengaturan variabel lingkungan melalui baris perintah:

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

Contoh Windows pengaturan variabel lingkungan melalui baris perintah:

```
setx AWS_PROFILE "my_default_profile_name"
```

aws.profile- properti JVM sistem

Untuk SDK untuk Kotlin pada JVM dan SDK untuk Java 2.x, Anda dapat [mengatur properti aws.profile sistem](#). Saat SDK membuat klien layanan, ia menggunakan pengaturan di profil bernama kecuali pengaturan diganti dalam kode. SDK untuk Java 1.x tidak mendukung properti sistem ini.

Note

Jika aplikasi Anda berada di server yang menjalankan beberapa aplikasi, kami sarankan Anda selalu menggunakan profil bernama daripada profil default. Profil default secara

otomatis diambil oleh AWS aplikasi apa pun di lingkungan dan dibagikan di antara mereka. Jadi, jika orang lain memperbarui profil default untuk aplikasi mereka, itu dapat secara tidak sengaja memengaruhi yang lain. Untuk melindungi terhadap hal ini, tentukan profil bernama dalam config file bersama dan kemudian gunakan profil bernama itu di aplikasi Anda dengan menyetel profil bernama dalam kode Anda. Anda dapat menggunakan variabel lingkungan atau properti JVM sistem untuk mengatur profil bernama jika Anda tahu bahwa cakupannya hanya memengaruhi aplikasi Anda.

Format file konfigurasi

configFile ini disusun menjadi beberapa bagian. Bagian adalah kumpulan pengaturan bernama, dan berlanjut sampai garis definisi bagian lain ditemukan.

configFile tersebut adalah file plaintext yang menggunakan format berikut:

- Semua entri dalam suatu bagian mengambil bentuk umum dari `setting-name=value`
- Baris dapat dikomentari dengan memulai baris dengan karakter hashtag (`#`).

Jenis bagian

Definisi bagian adalah garis yang menerapkan nama ke kumpulan pengaturan. Garis definisi bagian dimulai dan diakhiri dengan tanda kurung siku (`[]`). Di dalam tanda kurung, ada pengidentifikasi tipe bagian dan nama khusus untuk bagian tersebut. Anda dapat menggunakan huruf, angka, tanda hubung (`-`), dan garis bawah (`_`), tetapi tidak ada spasi.

Jenis bagian: **default**

Contoh baris definisi bagian: `[default]`

`[default]` adalah satu-satunya profil yang tidak memerlukan pengenalan profil bagian.

Contoh berikut menunjukkan config file dasar dengan `[default]` profil. Ini mengatur [region](#) pengaturan. Semua pengaturan yang mengikuti baris ini, hingga definisi bagian lain ditemukan, adalah bagian dari profil ini.

```
[default]
#Full line comment, this text is ignored.
```

```
region = us-east-2
```

Jenis bagian: **profile**

Contoh baris definisi bagian: `[profile dev]`

Garis definisi `profile` bagian adalah pengelompokan konfigurasi bernama yang dapat Anda terapkan untuk skenario pengembangan yang berbeda. Untuk lebih memahami profil bernama, lihat bagian sebelumnya di Profil.

Contoh berikut menunjukkan config file dengan garis definisi `profile` bagian dan profil bernama bernama `foo`. Semua pengaturan yang mengikuti baris ini, hingga definisi bagian lain ditemukan, adalah bagian dari profil bernama ini.

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

Beberapa pengaturan memiliki grup subpengaturan bersarang sendiri, seperti pengaturan dan `s3` subpengaturan dalam contoh berikut. Kaitkan subpengaturan dengan grup dengan membuat indentasi dengan satu spasi atau lebih.

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

Jenis bagian: **sso-session**

Contoh baris definisi bagian: `[sso-session my-sso]`

Baris definisi `sso-session` bagian menamai sekelompok pengaturan yang Anda gunakan untuk mengonfigurasi profil untuk menyelesaikan AWS kredensial yang digunakan. AWS IAM Identity Center Untuk informasi selengkapnya tentang mengonfigurasi autentikasi masuk tunggal, lihat [IAM Autentikasi Pusat Identitas untuk alat SDK atau Anda](#) Profil ditautkan ke `sso-session` bagian oleh pasangan kunci-nilai di mana `sso-session` kunci dan nama `sso-session` bagian Anda adalah nilainya, seperti. `sso-session = <name-of-sso-session-section>`

Contoh berikut mengonfigurasi profil yang akan mendapatkan AWS kredensial jangka pendek untuk IAM peran "" di akun "SampleRole111122223333" menggunakan token dari "my-sso". Bagian "my-

sso" sso-session direferensikan di profile bagian dengan nama menggunakan kunci. sso-session

```
[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
```

Jenis bagian: **services**

Contoh baris definisi bagian: [services dev]

Note

servicesBagian ini mendukung penyesuaian titik akhir khusus layanan dan hanya tersedia di SDKs dan alat yang menyertakan fitur ini. Untuk melihat apakah fitur ini tersedia untuk Anda SDK, lihat titik [Kompatibilitas dengan AWS SDKs](#) akhir khusus layanan.

Baris definisi services bagian menamai sekelompok pengaturan yang mengonfigurasi titik akhir kustom untuk Layanan AWS permintaan. Profil ditautkan ke services bagian oleh pasangan kunci-nilai di mana services kunci dan nama services bagian Anda adalah nilainya, seperti. services = <name-of-services-section>

servicesBagian ini selanjutnya dipisahkan menjadi subbagian dengan <SERVICE> = baris, di mana <SERVICE> adalah kunci Layanan AWS pengenalan. Layanan AWS Pengidentifikasi didasarkan pada API model serviceId dengan mengganti semua spasi dengan garis bawah dan huruf kecil semua huruf. Untuk daftar semua kunci pengenalan layanan yang akan digunakan di services bagian ini, lihat [Pengidentifikasi untuk titik akhir khusus layanan](#). Kunci pengenalan layanan diikuti oleh pengaturan bersarang dengan masing-masing pada barisnya sendiri dan menjorok oleh dua spasi.

Contoh berikut menggunakan services definisi untuk mengonfigurasi titik akhir yang akan digunakan untuk permintaan yang dibuat hanya untuk Amazon DynamoDB layanan. "local-dynamodb" servicesBagian ini direferensikan di profile bagian dengan nama menggunakan services kunci. Kunci Layanan AWS pengenalan adalah dynamodb. Subbagian Amazon DynamoDB

Lokasi yang dibagikan **config** dan **credentials** file

File bersama AWS config dan credentials file adalah file teks biasa yang menyimpan informasi konfigurasi untuk AWS SDK dan alat. File berada secara lokal di lingkungan Anda dan digunakan secara otomatis oleh kode SDK atau dengan AWS CLI perintah yang Anda jalankan di lingkungan itu. Misalnya, di komputer Anda sendiri atau saat mengembangkan instans Amazon Elastic Compute Cloud.

Ketika SDK atau alat berjalan, ia memeriksa file-file ini dan memuat pengaturan konfigurasi yang tersedia. Jika file belum ada, maka file dasar secara otomatis dibuat oleh SDK atau alat.

Secara default, file berada dalam folder bernama `.aws` yang ditempatkan di folder Anda home atau pengguna.

Sistem operasi	Lokasi default dan nama file
Linux dan macOS	<code>~/.aws/config</code> <code>~/.aws/credentials</code>
Windows	<code>%USERPROFILE%\aws\config</code> <code>%USERPROFILE%\aws\credentials</code>

Resolusi direktori rumah

`~` hanya digunakan untuk resolusi direktori home ketika:

- Memulai jalan
- Diikuti segera oleh `/` atau pemisah khusus platform. Di windows, `~/` dan `~\` keduanya menyelesaikan ke direktori home.

Saat menentukan direktori home, variabel-variabel berikut diperiksa:

- (Semua platform) Variabel HOME lingkungan
- (Platform Windows) Variabel USERPROFILE lingkungan
- (Platform Windows) Penggabungan HOMEDRIVE dan variabel HOMEPATH lingkungan (`$HOMEDRIVE$HOMEPATH`)

- (Opsional per SDK atau alat) SDK atau fungsi resolusi home path khusus alat atau variabel

Jika memungkinkan, jika direktori home pengguna ditentukan di awal jalur (misalnya, ~username/), itu diselesaikan ke direktori home nama pengguna yang diminta (misalnya, /home/username/.aws/config).

Ubah lokasi default file-file ini

Anda dapat menggunakan salah satu dari berikut ini untuk mengganti dari mana file-file ini dimuat oleh SDK atau alat.

Menggunakan variabel lingkungan

Variabel lingkungan berikut dapat diatur untuk mengubah lokasi atau nama file-file ini dari default ke nilai kustom:

- configvariabel lingkungan file: **AWS_CONFIG_FILE**
- credentialsvariabel lingkungan file: **AWS_SHARED_CREDENTIALS_FILE**

Linux/macOS

Anda dapat menentukan lokasi alternatif dengan menjalankan perintah [ekspor](#) berikut di Linux atau macOS.

```
$ 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

Anda dapat menentukan lokasi alternatif dengan menjalankan perintah [setx](#) berikut di Windows.

```
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
```

Untuk informasi selengkapnya tentang mengonfigurasi sistem Anda menggunakan variabel lingkungan, lihat [Dukungan variabel lingkungan](#).

Gunakan properti sistem JVM

Untuk SDK untuk Kotlin yang berjalan di JVM dan SDK for Java 2.x, Anda dapat mengatur properti sistem JVM berikut untuk mengubah lokasi atau nama file ini dari default ke nilai kustom:

- configfile properti sistem JVM: **aws.configFile**
- credentials variabel lingkungan file: **aws.sharedCredentialsFile**

Untuk petunjuk tentang cara mengatur properti sistem JVM, lihat [the section called “Cara mengatur properti sistem JVM”](#) SDK for Java 1.x tidak mendukung properti sistem ini.

Dukungan variabel lingkungan

Variabel lingkungan menyediakan cara lain untuk menentukan opsi konfigurasi dan kredensial, dan dapat berguna untuk membuat skrip atau mengatur sementara profil bernama sebagai default. Untuk daftar variabel lingkungan yang didukung oleh sebagian besar SDKs, lihat [Daftar variabel lingkungan](#).

Prioritas opsi

- Jika Anda menentukan setelan dengan menggunakan variabel lingkungannya, itu akan mengganti nilai apa pun yang dimuat dari profil di file bersama AWS config dan credentials file.
- Jika Anda menentukan pengaturan dengan menggunakan parameter pada baris AWS CLI perintah, itu akan mengganti nilai apa pun dari variabel lingkungan yang sesuai atau profil dalam file konfigurasi.

Cara mengatur variabel lingkungan

Contoh berikut menunjukkan bagaimana Anda dapat mengkonfigurasi variabel lingkungan untuk pengguna default.

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
```

Menyetel variabel lingkungan mengubah nilai yang digunakan hingga akhir sesi shell Anda, atau sampai Anda menyetel variabel ke nilai yang berbeda. Anda dapat membuat variabel persisten di seluruh sesi masa depan dengan menyetelnya di skrip startup shell Anda.

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
```

Menggunakan [set](#) untuk mengatur variabel lingkungan mengubah nilai yang digunakan sampai akhir sesi Command Prompt saat ini, atau sampai Anda mengatur variabel ke nilai yang berbeda. Menggunakan [setx](#) untuk mengatur variabel lingkungan mengubah nilai yang digunakan dalam sesi Command Prompt saat ini dan semua sesi Command Prompt yang Anda buat setelah menjalankan perintah. Itu tidak mempengaruhi shell perintah lain yang sudah berjalan pada saat Anda menjalankan perintah.

PowerShell

```
PS C:\> $Env:AWS_ACCESS_KEY_ID="AKIAIOSFODNN7EXAMPLE"
PS C:\> $Env:AWS_SECRET_ACCESS_KEY="wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY"
PS C:\>
\> $Env:AWS_SESSION_TOKEN="AQoEXAMPLEH4aoAH0gNCAPy...truncated...zrkuWJ0gQs8IZZaIv2BXIa2R40lgk"
PS C:\> $Env:AWS_REGION="us-west-2"
```

Jika Anda menetapkan variabel lingkungan pada PowerShell prompt seperti yang ditunjukkan pada contoh sebelumnya, itu menyimpan nilai hanya untuk durasi sesi saat ini. Untuk membuat pengaturan variabel lingkungan persisten di semua sesi PowerShell Command Prompt, simpan dengan menggunakan aplikasi Sistem di Control Panel. Atau, Anda dapat mengatur variabel untuk semua PowerShell sesi future dengan menambahkannya ke PowerShell profil Anda. Lihat [PowerShell dokumentasi](#) untuk informasi selengkapnya tentang menyimpan variabel lingkungan atau mempertahankannya di seluruh sesi.

Pengaturan variabel lingkungan tanpa server

Jika Anda menggunakan arsitektur tanpa server untuk pengembangan, Anda memiliki opsi lain untuk mengatur variabel lingkungan. Bergantung pada container Anda, Anda dapat menggunakan strategi

berbeda untuk kode yang berjalan di container tersebut untuk melihat dan mengakses variabel lingkungan, mirip dengan lingkungan non-cloud.

Misalnya, dengan AWS Lambda, Anda dapat langsung mengatur variabel lingkungan. Untuk detailnya, lihat [Menggunakan variabel AWS Lambda lingkungan](#) di Panduan AWS Lambda Pengembang.

Dalam Kerangka Tanpa Server, Anda sering dapat mengatur variabel SDK lingkungan dalam `serverless.yml` file di bawah kunci penyedia di bawah pengaturan lingkungan. Untuk informasi tentang `serverless.yml` file, lihat [Pengaturan fungsi umum](#) dalam dokumentasi Kerangka Tanpa Server.

Terlepas dari mekanisme mana yang Anda gunakan untuk mengatur variabel lingkungan kontainer, ada beberapa yang dicadangkan oleh kontainer, seperti yang didokumentasikan untuk Lambda pada variabel lingkungan [runtime yang ditentukan](#). Selalu konsultasikan dokumentasi resmi untuk wadah yang Anda gunakan untuk menentukan bagaimana variabel lingkungan diperlakukan dan apakah ada batasan.

Dukungan properti sistem JVM

[Properti sistem JVM](#) menyediakan cara lain untuk menentukan opsi konfigurasi dan kredensial untuk SDK yang berjalan di JVM seperti dan. AWS SDK for Java AWS SDK for Kotlin [Untuk daftar properti sistem JVM yang didukung oleh SDK, lihat Referensi pengaturan.](#)

Prioritas opsi

- Jika Anda menentukan setelan dengan menggunakan properti sistem JVM-nya, itu akan mengganti nilai apa pun yang ditemukan dalam variabel lingkungan atau dimuat dari profil di AWS dan file bersama. `config credentials`
- Jika Anda menentukan setelan dengan menggunakan variabel lingkungannya, itu akan mengganti nilai apa pun yang dimuat dari profil di AWS `config` dan `credentials` file bersama.

Cara mengatur properti sistem JVM

Anda dapat mengatur properti sistem JVM beberapa cara.

Pada baris perintah

Atur properti sistem JVM pada baris perintah saat menjalankan perintah dengan menggunakan sakelar `java -D`. Perintah berikut mengonfigurasi secara Wilayah AWS global untuk semua klien layanan kecuali Anda secara eksplisit mengganti nilai dalam kode.

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

Jika Anda perlu mengatur beberapa properti sistem JVM, tentukan `-D` sakelar beberapa kali.

Dengan variabel lingkungan

Jika Anda tidak dapat mengakses baris perintah untuk memanggil JVM untuk menjalankan aplikasi Anda, Anda dapat menggunakan variabel `JAVA_TOOL_OPTIONS` lingkungan untuk mengkonfigurasi opsi baris perintah. Pendekatan ini berguna dalam situasi seperti menjalankan AWS Lambda fungsi pada runtime Java atau menjalankan kode dalam JVM tertanam.

Contoh berikut mengonfigurasi Wilayah AWS secara global untuk semua klien layanan kecuali Anda secara eksplisit mengganti nilai dalam kode.

Linux, macOS, or Unix

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

Menyetel variabel lingkungan mengubah nilai yang digunakan hingga akhir sesi shell Anda, atau sampai Anda menyetel variabel ke nilai yang berbeda. Anda dapat membuat variabel persisten di seluruh sesi masa depan dengan menyetelnya di skrip startup shell Anda.

Windows Command Prompt

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

Menggunakan [set](#) untuk mengatur variabel lingkungan mengubah nilai yang digunakan sampai akhir sesi Command Prompt saat ini, atau sampai Anda mengatur variabel ke nilai yang berbeda. Menggunakan [setx](#) untuk mengatur variabel lingkungan mengubah nilai yang digunakan dalam sesi Command Prompt saat ini dan semua sesi Command Prompt yang Anda buat setelah menjalankan perintah. Itu tidak mempengaruhi shell perintah lain yang sudah berjalan pada saat Anda menjalankan perintah.

Saat runtime

Anda juga dapat mengatur properti sistem JVM saat runtime dalam kode dengan menggunakan `System.setProperty` metode seperti yang ditunjukkan pada contoh berikut.

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

Important

Tetapkan properti sistem JVM apa pun sebelum Anda menginisialisasi klien layanan SDK, jika tidak, klien layanan dapat menggunakan nilai lain.

Otentikasi dan akses

Anda harus menetapkan bagaimana kode Anda mengotentikasi dengan AWS ketika Anda mengembangkan dengan Layanan AWS. Anda dapat mengonfigurasi akses terprogram ke AWS sumber daya dengan cara yang berbeda, tergantung pada lingkungan dan AWS akses yang tersedia untuk Anda.

Opsi otentikasi untuk kode yang berjalan secara lokal (tidak masuk) AWS

- [IAM Autentikasi Pusat Identitas untuk alat SDK atau Anda](#)— Sebagai praktik keamanan terbaik, kami sarankan menggunakan AWS Organizations dengan IAM Identity Center untuk mengelola akses di semua Akun AWS. Anda dapat membuat pengguna di AWS IAM Identity Center, menggunakan Microsoft Active Directory, menggunakan penyedia identitas SAML 2.0 (iDP), atau secara individual menggabungkan IDP Anda ke Akun AWS. Untuk memeriksa apakah Wilayah Anda mendukung Pusat IAM Identitas, lihat [AWS IAM Identity Center titik akhir dan kuota](#) di Referensi Umum Amazon Web
- [Peran IAM Di Mana Saja](#)— Anda dapat menggunakan IAM Peran Di Mana Saja untuk mendapatkan kredensi keamanan sementara IAM untuk beban kerja seperti server, kontainer, dan aplikasi yang berjalan di luar. AWS Untuk menggunakan IAM Peran Di Mana Saja, beban kerja Anda harus menggunakan sertifikat X.509.
- [Asumsikan peran dengan AWS kredensial](#)— Anda dapat mengambil IAM peran untuk mengakses AWS sumber daya sementara yang mungkin tidak dapat Anda akses sebaliknya.
- [AWS kunci akses](#)— Pilihan lain yang mungkin kurang nyaman atau mungkin meningkatkan risiko keamanan untuk AWS sumber daya Anda.

Opsi otentikasi untuk kode yang berjalan dalam lingkungan AWS

Jika kode Anda berjalan AWS, kredensi dapat dibuat secara otomatis tersedia untuk aplikasi Anda. Misalnya, jika aplikasi Anda di-host di Amazon Elastic Compute Cloud, dan ada IAM peran yang terkait dengan sumber daya tersebut, kredensialnya secara otomatis tersedia untuk aplikasi Anda. Demikian juga, jika Anda menggunakan EKS kontainer Amazon ECS atau Amazon, kredensi yang ditetapkan untuk IAM peran tersebut dapat diperoleh secara otomatis oleh kode yang berjalan di dalam container melalui rantai penyedia SDK kredensi.

- [Menggunakan IAM peran untuk EC2 instans Amazon](#)— Gunakan IAM peran untuk menjalankan aplikasi Anda dengan aman di EC2 instans Amazon.

- Anda dapat berinteraksi secara terprogram dengan AWS menggunakan IAM Identity Center dengan cara-cara berikut:
 - Gunakan [AWS CloudShell](#) untuk menjalankan AWS CLI perintah dari konsol.
 - [Untuk mencoba ruang kolaborasi berbasis cloud untuk tim pengembangan perangkat lunak, pertimbangkan untuk menggunakan Amazon. CodeCatalyst](#)

Otentikasi melalui penyedia identitas berbasis web - Aplikasi web seluler atau berbasis klien

Jika Anda membuat aplikasi seluler atau aplikasi web berbasis klien yang memerlukan akses AWS, buat aplikasi Anda sehingga meminta kredensi AWS keamanan sementara secara dinamis dengan menggunakan federasi identitas web.

Dengan federasi identitas web, Anda tidak perlu membuat kode masuk khusus atau mengelola identitas pengguna Anda sendiri. Sebagai gantinya, pengguna aplikasi dapat masuk menggunakan penyedia identitas eksternal (IDP) yang terkenal, seperti Login with Amazon, Facebook, Google, atau iDP lain yang kompatibel dengan OpenID Connect (OIDC). Mereka dapat menerima token otentikasi, dan kemudian menukar token itu dengan kredensi keamanan sementara di peta AWS itu ke IAM peran dengan izin untuk menggunakan sumber daya di peta Anda. Akun AWS

Untuk mempelajari cara mengonfigurasi ini untuk alat SDK atau alat Anda, lihat [Mengambil peran dengan identitas web atau OpenID Connect](#).

Untuk aplikasi seluler, pertimbangkan untuk menggunakan Amazon Cognito. Amazon Cognito bertindak sebagai pialang identitas dan melakukan banyak pekerjaan federasi untuk Anda. Untuk informasi selengkapnya, lihat [Menggunakan Amazon Cognito untuk aplikasi seluler](#) di IAMPanduan Pengguna.

Informasi lebih lanjut tentang manajemen akses

Panduan IAM Pengguna memiliki informasi berikut tentang mengontrol akses ke AWS sumber daya secara aman:

- [IAM Identitas \(pengguna, grup pengguna, dan peran\)](#) — Memahami dasar-dasar identitas di AWS
- [Praktik terbaik keamanan dalam IAM](#) — Rekomendasi keamanan untuk diikuti ketika mengembangkan AWS aplikasi sesuai dengan model [tanggung jawab bersama](#).

Ini Referensi Umum Amazon Web memiliki dasar-dasar dasar sebagai berikut:

- [Memahami dan mendapatkan AWS kredensial Anda](#) — Akses opsi kunci dan praktik manajemen untuk akses konsol dan program.

ID AWS Builder

Anda ID AWS Builder melengkapi apa pun yang mungkin sudah Akun AWS Anda miliki atau ingin buat. Sementara Akun AWS bertindak sebagai wadah untuk AWS sumber daya yang Anda buat dan menyediakan batas keamanan untuk sumber daya tersebut, Anda ID AWS Builder mewakili Anda sebagai individu. Anda dapat masuk dengan Anda ID AWS Builder untuk mengakses alat dan layanan pengembang seperti Amazon CodeWhisperer dan Amazon CodeCatalyst.

- [ID AWS Builder Masuk dengan](#) Panduan AWS Sign-In Pengguna — Pelajari cara membuat dan menggunakan ID AWS Builder dan mempelajari apa yang disediakan Builder ID.
- [Mengautentikasi dengan CodeWhisperer dan AWS Toolkit - Builder ID](#) di Panduan CodeWhisperer Pengguna - Pelajari cara CodeWhisperer menggunakan file ID AWS Builder.
- [CodeCatalyst konsep - ID AWS Builder](#) di Panduan CodeCatalyst Pengguna Amazon - Pelajari cara CodeCatalyst menggunakan file ID AWS Builder.

IAM Autentikasi Pusat Identitas untuk alat SDK atau Anda

AWS IAM Identity Center adalah metode yang direkomendasikan untuk memberikan AWS kredensial saat mengembangkan pada layanan AWS non-komputasi. Misalnya, ini akan menjadi sesuatu seperti lingkungan pengembangan lokal Anda. Jika Anda mengembangkan AWS sumber daya, seperti Amazon Elastic Compute Cloud (AmazonEC2) atau AWS Cloud9, sebaiknya Anda mendapatkan kredensi dari layanan tersebut.

Dalam tutorial ini, Anda membuat akses Pusat IAM Identitas dan akan mengkonfigurasinya untuk Anda SDK atau alat dengan menggunakan portal AWS akses dan AWS CLI.

- Portal AWS akses adalah lokasi web tempat Anda masuk secara manual ke Pusat IAM Identitas. Format URL adalah `d-xxxxxxxxx.awsapps.com/start` atau `your_subdomain.awsapps.com/start`. Saat masuk ke portal AWS akses, Anda dapat melihat Akun AWS dan peran yang telah dikonfigurasi untuk pengguna tersebut. Prosedur ini menggunakan portal AWS akses untuk mendapatkan nilai konfigurasi yang Anda butuhkan untuk proses otentikasi SDK /tool.

- AWS CLI ini digunakan untuk mengonfigurasi SDK atau alat Anda untuk menggunakan otentikasi Pusat IAM Identitas untuk API panggilan yang dilakukan oleh kode Anda. Proses satu kali ini memperbarui AWS config file bersama Anda, yang kemudian digunakan oleh alat SDK atau Anda ketika Anda menjalankan kode Anda.

Konfigurasi akses terprogram menggunakan IAM Identity Center

Langkah 1: Tetapkan akses dan pilih set izin yang sesuai

Jika Anda belum mengaktifkan Pusat IAM Identitas, lihat [Mengaktifkan Pusat IAM Identitas](#) di Panduan AWS IAM Identity Center Pengguna.

Pilih salah satu metode berikut untuk mengakses AWS kredensial Anda.

Saya tidak memiliki akses melalui Pusat IAM Identitas

1. Tambahkan pengguna dan tambahkan izin administratif dengan mengikuti [Konfigurasi akses pengguna dengan prosedur direktori Pusat IAM Identitas default](#) di Panduan AWS IAM Identity Center Pengguna.
2. Set `AdministratorAccess` izin tidak boleh digunakan untuk pengembangan reguler. Sebagai gantinya, sebaiknya gunakan set `PowerUserAccess` izin yang telah ditentukan sebelumnya, kecuali majikan Anda telah membuat set izin khusus untuk tujuan ini.

Ikuti hal yang sama [Konfigurasi akses pengguna dengan prosedur direktori Pusat IAM Identitas default](#) lagi, tetapi kali ini:

- Alih-alih membuat *Admin team* grup, buat *Dev team* grup, dan ganti ini setelahnya dalam instruksi.
- Anda dapat menggunakan pengguna yang ada, tetapi pengguna harus ditambahkan ke *Dev team* grup baru.
- Alih-alih membuat set *AdministratorAccess* izin, buat set *PowerUserAccess* izin, dan ganti setelahnya dalam instruksi.

Setelah selesai, Anda harus memiliki yang berikut:

- Sebuah `Dev team` kelompok.
- `PowerUserAccess` izin terlampir yang disetel ke `Dev team` grup.

- Pengguna Anda ditambahkan ke Dev team grup.
3. Keluar dari portal dan masuk lagi untuk melihat opsi Anda Akun AWS dan untuk Administrator atau PowerUserAccess. Pilih PowerUserAccess saat bekerja dengan alat Anda/SDK.

Saya sudah memiliki akses AWS melalui penyedia identitas federasi yang dikelola oleh majikan saya (seperti Microsoft Entra atau Okta)

Masuk AWS melalui portal penyedia identitas Anda. Jika Administrator Cloud Anda telah memberi Anda izin PowerUserAccess (pengembang), Anda akan melihat Akun AWS bahwa Anda memiliki akses ke dan izin Anda ditetapkan. Di samping nama set izin Anda, Anda melihat opsi untuk mengakses akun secara manual atau terprogram menggunakan set izin tersebut.

Implementasi kustom dapat menghasilkan pengalaman yang berbeda, seperti nama set izin yang berbeda. Jika Anda tidak yakin izin mana yang disetel untuk digunakan, hubungi tim TI Anda untuk mendapatkan bantuan.

Saya sudah memiliki akses AWS melalui portal AWS akses yang dikelola oleh majikan saya

Masuk AWS melalui portal AWS akses. Jika Administrator Cloud Anda telah memberi Anda izin PowerUserAccess (pengembang), Anda akan melihat Akun AWS bahwa Anda memiliki akses ke dan izin Anda ditetapkan. Di samping nama set izin Anda, Anda melihat opsi untuk mengakses akun secara manual atau terprogram menggunakan set izin tersebut.

Saya sudah memiliki akses AWS melalui penyedia identitas kustom federasi yang dikelola oleh majikan saya

Hubungi tim TI Anda untuk bantuan.

Langkah 2: Konfigurasi SDKs dan alat untuk menggunakan Pusat IAM Identitas

1. Pada mesin pengembangan Anda, instal yang terbaru AWS CLI.
 - a. Lihat [Menginstal atau memperbarui versi terbaru AWS CLI dari](#) Panduan AWS Command Line Interface Pengguna.
 - b. (Opsional) Untuk memverifikasi AWS CLI bahwa berfungsi, buka prompt perintah dan jalankan `aws --version` perintah.

2. Masuk ke portal AWS akses. Majikan Anda dapat memberikan ini URL atau Anda mungkin mendapatkannya dalam email berikut Langkah 1: Tetapkan akses. Jika tidak, temukan portal AWS akses Anda URL di Dasbor <https://console.aws.amazon.com/singlesignon/>.
 - a. Di portal AWS akses, di tab Akun, pilih akun individual untuk dikelola. Peran untuk pengguna Anda ditampilkan. Pilih tombol Access untuk mendapatkan kredensial untuk baris perintah atau akses terprogram untuk set izin yang sesuai. Gunakan set `PowerUserAccess` izin yang telah ditentukan sebelumnya, atau izin mana pun yang telah Anda atau perusahaan Anda buat untuk menerapkan izin hak istimewa paling sedikit untuk pengembangan.
 - b. Di kotak dialog Dapatkan kredensi, pilih macOS dan Linux atau Windows, tergantung pada sistem operasi Anda.
 - c. Pilih metode kredensial Pusat IAM Identitas untuk mendapatkan nilai `Issuer URL` dan `SSO Region` nilai yang Anda butuhkan untuk langkah berikutnya. Catatan: `SSO Start URL` dapat digunakan secara bergantian dengan `Issuer URL`
3. Di AWS CLI command prompt, jalankan `aws configure sso` perintah. Saat diminta, masukkan nilai konfigurasi yang Anda kumpulkan di langkah sebelumnya. Untuk detail tentang AWS CLI perintah ini, lihat [Mengkonfigurasi profil Anda dengan `aws configure sso wizard`](#).
 - a. Untuk prompt `SSO Start URL`, masukkan nilai yang Anda peroleh `Issuer URL`.
 - b. Untuk nama CLI profil, kami sarankan untuk memasukkan `default` saat Anda memulai. Untuk informasi tentang cara menyetel profil non-default (bernama) dan variabel lingkungan terkaitnya, lihat [Profil](#).
4. (Opsional) Pada prompt AWS CLI perintah, konfirmasi identitas sesi aktif dengan menjalankan `aws sts get-caller-identity` perintah. Respons harus menunjukkan set izin Pusat IAM Identitas yang Anda konfigurasi.
5. Jika Anda menggunakan AWS SDK, buat aplikasi untuk Anda SDK di lingkungan pengembangan Anda.
 - a. Untuk beberapa SDKs, paket tambahan seperti `SSO` dan `SSO0IDC` harus ditambahkan ke aplikasi Anda sebelum Anda dapat menggunakan otentikasi Pusat IAM Identitas. Untuk detailnya, lihat spesifik Anda SDK.
 - b. Jika sebelumnya Anda mengonfigurasi akses ke AWS, tinjau `AWS credentials` file bersama Anda untuk apa pun [AWS kunci akses](#). Anda harus menghapus kredensi statis sebelum alat SDK atau akan menggunakan kredensi Pusat IAM Identitas karena diutamakan. [Memahami rantai penyedia kredensi](#)

Untuk mempelajari lebih dalam bagaimana alat SDKs dan menggunakan dan menyegarkan kredensial menggunakan konfigurasi ini, lihat. [Memahami autentikasi Pusat Identitas IAM](#)

Bergantung pada panjang sesi yang dikonfigurasi, akses Anda pada akhirnya akan kedaluwarsa dan alat SDK atau akan mengalami kesalahan otentikasi. Untuk me-refresh sesi portal akses lagi bila diperlukan, gunakan AWS CLI untuk menjalankan `aws sso login` perintah.

Anda dapat memperpanjang durasi sesi portal akses Pusat IAM Identitas dan durasi sesi yang ditetapkan izin. Ini memperpanjang jumlah waktu Anda dapat menjalankan kode sebelum Anda perlu masuk lagi secara manual dengan kode. AWS CLI Untuk informasi selengkapnya, lihat topik berikut di Panduan Pengguna AWS IAM Identity Center :

- IAM Durasi sesi Pusat Identitas - [Konfigurasi durasi sesi portal AWS akses pengguna Anda](#)
- Izin mengatur durasi sesi - [Mengatur durasi sesi](#)

Untuk detail tentang semua pengaturan SDKs dan alat penyedia Pusat IAM Identitas, lihat [IAM Penyedia kredensi Pusat Identitas](#) di panduan ini.

Memahami autentikasi Pusat Identitas IAM

Ketentuan Pusat Identitas IAM yang relevan

Istilah berikut membantu Anda memahami proses dan konfigurasi di belakang AWS IAM Identity Center. Dokumentasi untuk API AWS SDK menggunakan nama yang berbeda dari IAM Identity Center untuk beberapa konsep otentikasi ini. Sangat membantu untuk mengetahui kedua nama tersebut.

Tabel berikut menunjukkan bagaimana nama-nama alternatif berhubungan satu sama lain.

Nama Pusat Identitas IAM	Nama API SDK	Deskripsi
Pusat Identitas	sso	Meskipun AWS Single Sign-On diganti namanya, namespace sso API akan mempertahankan nama aslinya untuk tujuan kompatibilitas mundur. Untuk informasi selengkapnya, lihat ganti

Nama Pusat Identitas IAM	Nama API SDK	Deskripsi
		nama Pusat Identitas IAM di AWS IAM Identity Center Panduan Pengguna.
Konsol Pusat Identitas IAM Konsol administratif		Konsol yang Anda gunakan untuk mengonfigurasi sistem masuk tunggal.
AWSURL portal akses		URL yang unik untuk akun Pusat Identitas IAM Anda, seperti <code>https://xxx.awsapps.com/start</code> . Anda masuk ke portal ini menggunakan kredensial masuk Pusat Identitas IAM Anda.
Sesi Portal Akses Pusat Identitas IAM	Sesi otentikasi	Menyediakan token akses pembawa ke penelepon.
Sesi set izin		Sesi IAM yang digunakan SDK secara internal untuk melakukan panggilan. Layanan AWS Dalam diskusi informal, Anda mungkin melihat ini salah disebut sebagai “sesi peran.”
Izin menetapkan kredensial	Kredensial AWS kredensial sigv4	Kredensial yang sebenarnya digunakan SDK untuk sebagian besar Layanan AWS panggilan (khususnya, semua panggilan Layanan AWS sigv4). Dalam diskusi informal, Anda mungkin melihat ini salah disebut sebagai “kredensial peran.”

Nama Pusat Identitas IAM	Nama API SDK	Deskripsi
Penyedia kredensi Pusat Identitas IAM	Penyedia kredensi SSO	Bagaimana Anda mendapatkan kredensialnya, seperti kelas atau modul yang menyediakan fungsionalitas.

Memahami resolusi kredensi SDK untuk Layanan AWS

API Pusat Identitas IAM menukar kredensial token pembawa untuk kredensial sigv4. Sebagian besar Layanan AWS adalah API sigv4, dengan beberapa pengecualian seperti dan. Amazon CodeWhisperer Amazon CodeCatalyst Berikut ini menjelaskan proses resolusi kredensial untuk mendukung sebagian besar Layanan AWS panggilan untuk kode aplikasi Anda. AWS IAM Identity Center

Memulai sesi portal AWS akses

- Mulai proses dengan masuk ke sesi dengan kredensi Anda.
 - Gunakan `aws sso login` perintah di AWS Command Line Interface (AWS CLI). Ini memulai sesi Pusat Identitas IAM baru jika Anda belum memiliki sesi aktif.
- Saat memulai sesi baru, Anda menerima token penyegaran dan token akses dari IAM Identity Center. Ini AWS CLI juga memperbarui file JSON cache SSO dengan token akses baru dan token penyegaran dan membuatnya tersedia untuk digunakan oleh SDK.
- Jika Anda sudah memiliki sesi aktif, AWS CLI perintah menggunakan kembali sesi yang ada dan akan kedaluwarsa setiap kali sesi yang ada berakhir. Untuk mempelajari cara mengatur panjang sesi Pusat Identitas IAM, lihat [Mengonfigurasi durasi sesi portal AWS akses pengguna Anda](#) di AWS IAM Identity Center Panduan Pengguna.
 - Panjang sesi maksimum telah diperpanjang hingga 90 hari untuk mengurangi kebutuhan untuk sering masuk.

Bagaimana SDK mendapatkan kredensial untuk panggilan Layanan AWS

SDK menyediakan akses Layanan AWS saat Anda membuat instance objek klien per layanan. Ketika profil yang dipilih dari `AWS config` file bersama dikonfigurasi untuk resolusi kredensial Pusat Identitas IAM, Pusat Identitas IAM digunakan untuk menyelesaikan kredensial untuk aplikasi Anda.

- [Proses resolusi kredensial](#) selesai selama runtime ketika klien dibuat.

Untuk mengambil kredensial API sigv4 menggunakan sistem masuk tunggal Pusat Identitas IAM, SDK menggunakan token akses Pusat Identitas IAM untuk mendapatkan sesi IAM. Sesi IAM ini disebut sesi set izin, dan menyediakan AWS akses ke SDK dengan mengasumsikan peran IAM.

- Durasi sesi yang ditetapkan izin diatur secara independen dari durasi sesi Pusat Identitas IAM.
 - Untuk mempelajari cara menyetel durasi sesi yang ditetapkan izin, lihat [Mengatur durasi sesi](#) di Panduan AWS IAM Identity Center Pengguna.
- Ketahuilah bahwa kredensial set izin juga disebut sebagai kredensial dan AWS kredensial sigv4 di sebagian besar dokumentasi SDK API. AWS

Kredensial set izin dikembalikan dari panggilan ke API Pusat [getRoleCredentials](#) Identitas IAM ke SDK. Objek klien SDK menggunakan peran IAM yang diasumsikan untuk melakukan panggilan ke Layanan AWS, seperti meminta Amazon S3 untuk mencantumkan bucket di akun Anda. Objek klien dapat terus beroperasi menggunakan kredensial set izin tersebut hingga sesi set izin berakhir.

Kedaluwarsa sesi dan penyegaran

Saat menggunakan token akses per jam yang diperoleh dari IAM Identity Center secara otomatis di-refresh menggunakan token penyegaran. [SSO konfigurasi penyedia token](#)

- Jika token akses kedaluwarsa saat SDK mencoba menggunakannya, SDK menggunakan token penyegaran untuk mencoba mendapatkan token akses baru. Pusat Identitas IAM membandingkan token penyegaran dengan durasi sesi portal akses Pusat Identitas IAM Anda. Jika token penyegaran tidak kedaluwarsa, Pusat Identitas IAM merespons dengan token akses lain.
- Token akses ini dapat digunakan untuk menyegarkan sesi set izin klien yang ada, atau untuk menyelesaikan kredensial untuk klien baru.

Namun, jika sesi portal akses IAM Identity Center kedaluwarsa, maka tidak ada token akses baru yang diberikan. Oleh karena itu, durasi yang ditetapkan izin tidak dapat diperpanjang. Ini akan kedaluwarsa (dan akses akan hilang) setiap kali izin cache menetapkan waktu lama sesi habis untuk klien yang ada.

Kode apa pun yang membuat klien baru akan gagal otentikasi segera setelah sesi IAM Identity Center berakhir. Ini karena kredensial set izin tidak di-cache. Kode Anda tidak akan dapat membuat klien baru dan menyelesaikan proses resolusi kredensial sampai Anda memiliki token akses yang valid.

Untuk rekap, saat SDK memerlukan kredensial set izin baru, SDK terlebih dahulu memeriksa kredensial yang valid dan sudah ada dan menggunakannya. Ini berlaku apakah kredensialnya untuk klien baru atau untuk klien yang sudah ada dengan kredensi kedaluwarsa. Jika kredensialnya tidak ditemukan atau tidak valid, maka SDK akan memanggil API Pusat Identitas IAM untuk mendapatkan kredensi baru. Untuk memanggil API, diperlukan token akses. Jika token akses kedaluwarsa, SDK menggunakan token penyegaran untuk mencoba mendapatkan token akses baru dari layanan Pusat Identitas IAM. Token ini diberikan jika sesi portal akses Pusat Identitas IAM Anda tidak kedaluwarsa.

Peran IAM Di Mana Saja

Anda dapat menggunakan IAM Roles Anywhere untuk mendapatkan kredensial keamanan sementara di IAM untuk beban kerja seperti server, kontainer, dan aplikasi yang berjalan di luar. AWS Untuk menggunakan Peran IAM Di Mana Saja, beban kerja Anda harus menggunakan sertifikat X.509. Administrator Cloud Anda harus menyediakan sertifikat dan kunci pribadi yang diperlukan untuk mengonfigurasi Peran IAM Anywhere sebagai penyedia kredensial Anda.

Langkah 1: Konfigurasi Peran IAM Di Mana Saja

IAM Roles Anywhere menyediakan cara untuk mendapatkan kredensial sementara untuk beban kerja atau proses yang berjalan di luar. AWS Jangkar kepercayaan didirikan dengan otoritas sertifikat untuk mendapatkan kredensial sementara untuk peran IAM terkait. Peran menetapkan izin yang akan dimiliki beban kerja Anda saat kode Anda diautentikasi dengan IAM Roles Anywhere.

Untuk langkah-langkah menyiapkan profil jangkar kepercayaan, peran IAM, dan Peran IAM Di Mana Saja, lihat [Membuat jangkar kepercayaan dan profil di Peran Di Mana Saja di Panduan Pengguna IAM AWS Identity and Access Management Roles Anywhere](#).

Note

Profil dalam Panduan Pengguna IAM Roles Anywhere mengacu pada konsep unik dalam layanan IAM Roles Anywhere. Ini tidak terkait dengan profil dalam AWS config file bersama.

Langkah 2: Gunakan Peran IAM Di Mana Saja

Untuk mendapatkan kredensial keamanan sementara dari IAM Roles Anywhere, gunakan alat bantu kredensial yang disediakan oleh IAM Roles Anywhere. Alat kredensial mengimplementasikan proses penandatanganan untuk IAM Roles Anywhere.

Untuk petunjuk mengunduh alat bantu kredensial, lihat [Memperoleh kredensial keamanan sementara dari AWS Identity and Access Management Peran Di Mana Saja di Panduan Pengguna IAM Roles Anywhere](#).

Untuk menggunakan kredensial keamanan sementara dari IAM Roles Anywhere dengan AWS SDK dan SDK AWS CLI, Anda dapat mengonfigurasi `credential_process` pengaturan dalam file bersama. AWS config SDK dan AWS CLI mendukung penyedia kredensial proses yang digunakan `credential_process` untuk mengautentikasi. Berikut ini menunjukkan struktur umum yang akan ditetapkan `credential_process`.

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

`credential-process` Perintah alat pembantu mengembalikan kredensial sementara dalam format JSON standar yang kompatibel dengan pengaturan. `credential_process` Perhatikan bahwa nama perintah berisi tanda hubung tetapi nama pengaturan berisi garis bawah. Perintah ini membutuhkan parameter berikut:

- `private-key`— Jalur ke kunci pribadi yang menandatangani permintaan.
- `certificate`— Jalan menuju sertifikat.
- `role-arn`— ARN dari peran untuk mendapatkan kredensial sementara untuk.
- `profile-arn`— ARN profil yang menyediakan pemetaan untuk peran yang ditentukan.
- `trust-anchor-arn`— ARN dari jangkar kepercayaan yang digunakan untuk mengotentikasi.

Administrator Cloud Anda harus menyediakan sertifikat dan kunci pribadi. Ketiga nilai ARN dapat disalin dari file. AWS Management Console Contoh berikut menunjukkan config file bersama yang mengonfigurasi pengambilan kredensial sementara dari alat pembantu.

```
[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
```

Untuk parameter opsional dan detail alat pembantu tambahan, lihat [IAM Roles Anywhere Credential Helper](#) aktif. GitHub

Untuk detail tentang pengaturan konfigurasi SDK itu sendiri dan penyedia kredensi proses, lihat [Penyedia kredensi proses](#) di panduan ini.

Asumsikan peran dengan AWS kredensial

Dengan asumsi peran melibatkan penggunaan seperangkat kredensial keamanan sementara untuk mengakses AWS sumber daya yang mungkin tidak dapat Anda akses sebaliknya. Kredensial sementara ini terdiri dari access key ID, secret access key, dan token keamanan. Untuk mempelajari lebih lanjut tentang AWS Security Token Service (AWS STS) API permintaan, lihat [Tindakan](#) dalam AWS Security Token Service API Referensi.

Untuk mengatur SDK atau alat Anda untuk mengambil peran, Anda harus terlebih dahulu membuat atau mengidentifikasi peran tertentu diasumsikan. IAM peran diidentifikasi secara unik oleh peran Amazon Resource Name (ARN). Peran membangun hubungan kepercayaan dengan entitas lain. Entitas tepercaya yang menggunakan peran tersebut mungkin satu Layanan AWS atau lainnya Akun AWS. Untuk mempelajari lebih lanjut tentang IAM peran, lihat [Menggunakan IAM peran](#) di Panduan IAM Pengguna.

Setelah IAM peran diidentifikasi, jika Anda dipercaya oleh peran tersebut, Anda dapat mengonfigurasi SDK atau alat untuk menggunakan izin yang diberikan oleh peran tersebut.

Note

Ini adalah praktik AWS terbaik untuk menggunakan titik akhir Regional bila memungkinkan dan untuk mengkonfigurasi Anda [Wilayah AWS](#).

Asumsikan IAM peran

Saat mengasumsikan peran, AWS STS mengembalikan satu set kredensial keamanan sementara. Kredensial ini bersumber dari profil lain atau dari instance atau wadah tempat kode Anda berjalan. Paling umum jenis asumsi peran ini digunakan ketika Anda memiliki AWS kredensial untuk satu akun, tetapi aplikasi Anda memerlukan akses ke sumber daya di akun lain.

Langkah 1: Siapkan IAM peran

Untuk mengatur SDK atau alat Anda untuk mengambil peran, Anda harus terlebih dahulu membuat atau mengidentifikasi peran tertentu untuk diasumsikan. IAM peran diidentifikasi secara unik menggunakan peran. [ARN](#) Peran membangun hubungan kepercayaan dengan entitas lain, biasanya di dalam akun Anda atau untuk akses lintas akun. Untuk mengatur ini, lihat [Membuat IAM peran](#) di Panduan IAM Pengguna.

Langkah 2: Konfigurasi alat SDK atau

Konfigurasi alat SDK atau untuk sumber kredensial dari `credential_source` atau `source_profile`

Gunakan `credential_source` untuk sumber kredensial dari ECS wadah Amazon, EC2 instans Amazon, atau dari variabel lingkungan.

Gunakan `source_profile` untuk sumber kredensial dari profil lain. `source_profile` juga mendukung rantai peran, yang merupakan hierarki profil di mana peran yang diasumsikan kemudian digunakan untuk mengambil peran lain.

Saat Anda menentukan ini di profil, alat SDK atau secara otomatis membuat AWS STS [AssumeRole](#) API panggilan yang sesuai untuk Anda. Untuk mengambil dan menggunakan kredensial sementara dengan mengasumsikan peran, tentukan nilai konfigurasi berikut dalam file bersama. AWS config Untuk detail selengkapnya tentang masing-masing pengaturan ini, lihat [Asumsikan pengaturan penyedia kredensi peran](#) bagian.

- `role_arn`- Dari IAM peran yang Anda buat di Langkah 1
- Konfigurasi salah satu `source_profile` atau `credential_source`
- (Opsional) `duration_seconds`
- (Opsional) `external_id`
- (Opsional) `mfa_serial`
- (Opsional) `role_session_name`

Contoh berikut menunjukkan konfigurasi kedua opsi peran asumsi dalam config file bersama:

```
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
```

Untuk detail tentang semua pengaturan penyedia kredensi peran asumsi, lihat [Asumsikan penyedia kredensi peran](#) di panduan ini.

Mengambil peran dengan identitas web atau OpenID Connect

Dengan asumsi peran melibatkan penggunaan seperangkat kredensial keamanan sementara untuk mengakses AWS sumber daya yang mungkin tidak dapat Anda akses sebaliknya. Kredensial sementara ini terdiri dari access key ID, secret access key, dan token keamanan. Untuk mempelajari lebih lanjut tentang AWS Security Token Service (AWS STS) API permintaan, lihat [Tindakan](#) dalam AWS Security Token Service API Referensi.

Untuk mengatur SDK atau alat Anda untuk mengambil peran, Anda harus terlebih dahulu membuat atau mengidentifikasi peran tertentu untuk diasumsikan. IAM peran diidentifikasi secara unik oleh peran Amazon Resource Name (ARN). Peran membangun hubungan kepercayaan dengan entitas lain. Entitas tepercaya yang menggunakan peran mungkin penyedia identitas web atau OpenID Connect (OIDC), atau SAML federasi. Untuk mempelajari lebih lanjut tentang IAM peran, lihat [Metode untuk mengambil peran](#) dalam Panduan IAM Pengguna.

Setelah IAM peran dikonfigurasi dalam peran Anda SDK, jika peran tersebut dikonfigurasi untuk mempercayai penyedia identitas Anda, Anda dapat mengonfigurasi lebih lanjut SDK untuk mengambil peran tersebut untuk mendapatkan AWS kredensial sementara.

Note

Ini adalah praktik AWS terbaik untuk menggunakan titik akhir Regional bila memungkinkan dan untuk mengkonfigurasi Anda [Wilayah AWS](#).

Bersekutu dengan identitas web atau OpenID Connect

Anda dapat menggunakan Token JSON Web (JWTs) dari penyedia identitas publik, seperti Login Dengan Amazon, Facebook, Google untuk mendapatkan AWS kredensial sementara menggunakan `AssumeRoleWithWebIdentity`. Tergantung pada bagaimana mereka digunakan, ini JWTs dapat disebut token ID atau token akses. Anda juga dapat menggunakan yang JWTs diterbitkan dari

penyedia identitas (IdPs) yang kompatibel dengan OIDC protokol penemuan, seperti EntraID atau PingFederate.

Jika Anda menggunakan Amazon Elastic Kubernetes Service, fitur ini menyediakan kemampuan untuk menentukan peran yang IAM berbeda untuk setiap akun layanan Anda di kluster Amazon. EKS Fitur Kubernetes ini mendistribusikan JWTs ke pod Anda yang kemudian digunakan oleh penyedia kredensi ini untuk mendapatkan kredensi sementara. AWS Untuk informasi selengkapnya tentang EKS konfigurasi Amazon ini, lihat [IAM peran untuk akun layanan](#) di Panduan EKS Pengguna Amazon. Namun, untuk opsi yang lebih sederhana, kami sarankan Anda menggunakan [Amazon EKS Pod Identities](#) sebagai gantinya jika Anda [SDK mendukungnya](#).

Langkah 1: Siapkan penyedia identitas dan IAM peran

Untuk mengkonfigurasi federasi dengan iDP eksternal, gunakan penyedia IAM identitas untuk menginformasikan AWS tentang iDP eksternal dan konfigurasinya. Ini membangun kepercayaan antara IDP Anda Akun AWS dan eksternal. Sebelum mengonfigurasi SDK untuk menggunakan Token JSON Web (JWT) untuk otentikasi, Anda harus terlebih dahulu mengatur penyedia identitas (iDP) dan IAM peran yang digunakan untuk mengaksesnya. Untuk mengaturnya, lihat [Membuat peran untuk identitas web atau Federasi OpenID Connect \(konsol\)](#) di IAMPanduan Pengguna.

Langkah 2: Konfigurasi alat SDK atau

Konfigurasi alat SDK atau untuk menggunakan Token JSON Web (JWT) dari AWS STS untuk otentikasi.

Saat Anda menentukan ini di profil, alat SDK atau secara otomatis membuat AWS STS [AssumeRoleWithWebIdentity](#) API panggilan yang sesuai untuk Anda. Untuk mengambil dan menggunakan kredensial sementara menggunakan federasi identitas web, tentukan nilai konfigurasi berikut dalam file bersama. AWS config Untuk detail selengkapnya tentang masing-masing pengaturan ini, lihat [Asumsikan pengaturan penyedia kredensi peran](#) bagian.

- `role_arn`- Dari IAM peran yang Anda buat di Langkah 1
- `web_identity_token_file`- Dari iDP eksternal
- (Opsional) `duration_seconds`
- (Opsional) `role_session_name`

Berikut ini adalah contoh konfigurasi `config` file bersama untuk mengambil peran dengan identitas web:

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

Note

Untuk aplikasi seluler, pertimbangkan untuk menggunakan Amazon Cognito. Amazon Cognito bertindak sebagai pialang identitas dan melakukan banyak pekerjaan federasi untuk Anda. Namun, penyedia identitas Amazon Cognito tidak disertakan dalam pustaka inti SDKs dan alat seperti penyedia identitas lainnya. Untuk mengakses Amazon Cognito API, sertakan klien layanan Amazon Cognito di build atau pustaka untuk alat atau Anda. SDK Untuk penggunaan dengan AWS SDKs, lihat [Contoh Kode](#) di Panduan Pengembang Amazon Cognito.

Untuk detail tentang semua pengaturan penyedia kredensi peran asumsi, lihat [Asumsikan penyedia kredensi peran](#) di panduan ini.

AWS kunci akses

Gunakan kredensial jangka pendek

Sebaiknya konfigurasi SDK atau alat yang akan digunakan [IAM Autentikasi Pusat Identitas untuk alat SDK atau Anda](#) untuk menggunakan opsi durasi sesi yang diperpanjang.

Namun, untuk mengatur SDK atau kredensial sementara alat secara langsung, lihat [Otentikasi menggunakan kredensial jangka pendek](#)

Gunakan kredensial jangka panjang

Warning

Untuk menghindari risiko keamanan, jangan gunakan pengguna IAM untuk otentikasi saat mengembangkan perangkat lunak yang dibuat khusus atau bekerja dengan data nyata. Sebaliknya, gunakan federasi dengan penyedia identitas seperti [AWS IAM Identity Center](#).

Mengelola akses di seluruh Akun AWS

Sebagai praktik keamanan terbaik, kami sarankan menggunakan AWS Organizations dengan IAM Identity Center untuk mengelola akses di semua Akun AWS. Untuk informasi selengkapnya tentang administrator, lihat [Praktik terbaik keamanan di IAM](#) di dalam Panduan Pengguna IAM.

Anda dapat membuat pengguna di Pusat Identitas IAM, menggunakan Microsoft Active Directory, menggunakan penyedia identitas SAMP 2.0 (iDP), atau menggabungkan IDP Anda secara individual. Akun AWS Dengan menggunakan salah satu pendekatan ini, Anda dapat memberikan pengalaman masuk tunggal untuk pengguna Anda. Anda juga dapat menerapkan otentikasi multi-faktor (MFA) dan menggunakan kredensi sementara untuk akses. Akun AWS Ini berbeda dari pengguna IAM, yang merupakan kredensi jangka panjang yang dapat dibagikan dan yang dapat meningkatkan risiko keamanan terhadap sumber daya Anda AWS .

Buat pengguna IAM hanya untuk lingkungan kotak pasir

Jika Anda baru AWS, Anda dapat membuat pengguna IAM uji dan kemudian menggunakannya untuk menjalankan tutorial dan menjelajahi apa yang AWS ditawarkan. Tidak apa-apa untuk menggunakan jenis kredensi ini saat Anda belajar, tetapi kami sarankan Anda menghindari menggunakannya di luar lingkungan kotak pasir.

Untuk kasus penggunaan berikut, mungkin masuk akal untuk memulai dengan pengguna IAM di AWS:

- Memulai AWS SDK atau alat Anda dan menjelajah Layanan AWS di lingkungan kotak pasir.
- Menjalankan skrip terjadwal, pekerjaan, dan proses otomatis lainnya yang tidak mendukung proses masuk yang dihadiri manusia sebagai bagian dari pembelajaran Anda.

Jika Anda menggunakan pengguna IAM di luar kasus penggunaan ini, maka transisi ke IAM Identity Center atau federasikan penyedia identitas Anda ke Akun AWS sesegera mungkin. Untuk informasi lebih lanjut, lihat [Federasi identitas di AWS](#).

Kunci akses pengguna IAM yang aman

Anda harus memutar kunci akses pengguna IAM secara teratur. Ikuti panduan dalam [Memutar kunci akses](#) di Panduan Pengguna IAM. Jika Anda yakin bahwa Anda telah secara tidak sengaja membagikan kunci akses pengguna IAM Anda, kemudian putar kunci akses Anda.

Kunci akses pengguna IAM harus disimpan dalam `AWS credentials` file bersama di mesin lokal. Jangan menyimpan kunci akses pengguna IAM dalam kode Anda. Jangan sertakan file konfigurasi yang berisi kunci akses pengguna IAM Anda di dalam perangkat lunak manajemen kode sumber apa pun. Alat eksternal, seperti [git-secret](#) proyek open source, dapat membantu Anda dari secara tidak sengaja melakukan informasi sensitif ke repositori Git. Untuk informasi selengkapnya, lihat [Identitas IAM \(pengguna, grup pengguna, dan peran\)](#) di Panduan Pengguna IAM.

Untuk mengatur pengguna IAM untuk memulai, lihat [Otentikasi menggunakan kredensi jangka panjang](#).

Otentikasi menggunakan kredensi jangka pendek

Sebaiknya konfigurasi alat SDK atau alat Anda untuk digunakan [IAM Autentikasi Pusat Identitas untuk alat SDK atau Anda](#) dengan opsi durasi sesi yang diperpanjang. Namun, Anda dapat menyalin dan menggunakan kredensi sementara yang tersedia di portal akses AWS. Kredensi baru perlu disalin ketika ini kedaluwarsa. Anda dapat menggunakan kredensi sementara di profil atau menggunakannya sebagai nilai untuk properti sistem dan variabel lingkungan.

Praktik terbaik: Alih-alih mengelola kunci akses dan token secara manual dalam file kredensial, kami menyarankan aplikasi Anda menggunakan kredensi sementara yang dikirimkan dari:

- Layanan AWS komputasi, seperti menjalankan aplikasi Anda di Amazon Elastic Compute Cloud atau di AWS Lambda
- Pilihan lain dalam rantai penyedia kredensi, seperti [IAM Autentikasi Pusat Identitas untuk alat SDK atau Anda](#).
- Atau gunakan [Penyedia kredensi proses](#) untuk mengambil kredensi sementara.

Siapkan file kredensial menggunakan kredensial jangka pendek yang diambil dari portal akses AWS

1. [Buat file kredensial bersama](#).
2. Dalam file kredensial, rekatkan teks placeholder berikut hingga Anda menempelkan kredensi sementara yang berfungsi.

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


Jika Anda menggunakan IAM pengguna untuk menjalankan kode Anda, maka alat SDK atau di lingkungan pengembangan Anda mengautentikasi dengan menggunakan kredensi IAM pengguna jangka panjang dalam file bersama. `AWS credentials` Tinjau [praktik terbaik Keamanan dalam IAM](#) topik dan transisi ke Pusat IAM Identitas atau kredensial sementara lainnya sesegera mungkin.

Peringatan penting dan panduan untuk kredensial

Peringatan untuk kredensial

- **NOT**Gunakan kredensi root akun Anda untuk mengakses AWS sumber daya. Kredensi ini menyediakan akses akun yang tidak terbatas dan sulit dicabut.
- **NOT**Masukkan kunci akses literal atau informasi kredensi dalam file aplikasi Anda. Jika Anda melakukannya, Anda membuat risiko secara tidak sengaja mengekspos kredensialnya jika, misalnya, Anda mengunggah proyek ke repositori publik.
- **NOT**Sertakan file yang berisi kredensial di area proyek Anda.
- Ketahuilah bahwa kredensial apa pun yang disimpan dalam `AWS credentials` file bersama disimpan dalam teks biasa.

Panduan tambahan untuk mengelola kredensial dengan aman

Untuk diskusi umum tentang cara mengelola AWS kredensial dengan aman, lihat [Praktik terbaik untuk mengelola kunci AWS akses](#) di [Referensi Umum AWS](#) Selain diskusi itu, pertimbangkan hal-hal berikut:

- Gunakan [IAM peran untuk tugas untuk tugas](#) Amazon Elastic Container Service (Amazon ECS).
- Gunakan [IAM peran](#) untuk aplikasi yang berjalan di EC2 instans Amazon.

Prasyarat: Buat akun AWS

Untuk menggunakan IAM pengguna untuk mengakses AWS layanan, Anda memerlukan AWS akun dan AWS kredensial.

1. Buat akun.

Untuk membuat AWS akun, lihat [Memulai: Apakah Anda AWS pengguna pertama kali?](#) dalam Panduan AWS Account Management Referensi.

2. Buat pengguna administratif.

Hindari menggunakan akun pengguna root Anda (akun awal yang Anda buat) untuk mengakses konsol manajemen dan layanan. Sebagai gantinya, buat akun pengguna administratif, seperti yang dijelaskan dalam [Buat pengguna administratif](#) di Panduan IAM Pengguna.

Setelah Anda membuat akun pengguna administratif dan mencatat detail login, pastikan untuk keluar dari akun pengguna root Anda dan masuk kembali menggunakan akun administratif.

Tak satu pun dari akun ini sesuai untuk melakukan pengembangan pada AWS atau untuk menjalankan aplikasi di AWS. Sebagai praktik terbaik, Anda perlu membuat pengguna, set izin, atau peran layanan yang sesuai untuk tugas-tugas ini. Untuk informasi selengkapnya, lihat [Menerapkan izin hak istimewa terkecil](#) di Panduan Pengguna. IAM


Langkah 1: Buat IAM pengguna Anda

- Buat IAM pengguna Anda dengan mengikuti prosedur [Membuat IAM pengguna \(konsol\)](#) di Panduan IAM Pengguna. Saat membuat IAM pengguna Anda:
 - Kami menyarankan Anda memilih Menyediakan akses pengguna ke AWS Management Console. Ini memungkinkan Anda untuk melihat Layanan AWS terkait dengan kode yang Anda jalankan di lingkungan visual, seperti memeriksa log AWS CloudTrail diagnostik atau mengunggah file ke Amazon Simple Storage Service, yang sangat membantu saat men-debug kode Anda.
 - Untuk Setel izin - Opsi izin, pilih Lampirkan kebijakan secara langsung untuk mengetahui cara Anda ingin menetapkan izin kepada pengguna ini.
 - Sebagian besar SDK tutorial “Memulai” menggunakan layanan Amazon S3 sebagai contoh. Untuk menyediakan aplikasi Anda dengan akses penuh ke Amazon S3, pilih `AmazonS3FullAccess` kebijakan untuk melampirkan ke pengguna ini.
 - Anda dapat mengabaikan langkah-langkah opsional dari prosedur tersebut mengenai pengaturan batas izin atau tag.

Langkah 2: Dapatkan kunci akses Anda

1. Di panel navigasi IAM konsol, pilih Pengguna dan kemudian pilih pengguna **User name** yang Anda buat sebelumnya.
2. Pada halaman pengguna, pilih halaman Security credentials. Kemudian, di bawah tombol Access, pilih Create Access Key.

3. Untuk Buat kunci akses Langkah 1, pilih Command Line Interface (CLI) atau kode Lokal. Kedua opsi menghasilkan jenis kunci yang sama untuk digunakan dengan kedua AWS CLI dan SDKs.
4. Untuk Buat tombol akses Langkah 2, masukkan tag opsional dan pilih Berikutnya.
5. Untuk Buat kunci akses Langkah 3, pilih Unduh file.csv untuk menyimpan .csv file dengan kunci akses IAM pengguna dan kunci akses rahasia. Anda memerlukan informasi ini untuk nanti.

 Warning

Gunakan langkah-langkah keamanan yang tepat untuk menjaga kredensial ini tetap aman.

6. Pilih Selesai.

Langkah 3: Perbarui **credentials** file bersama

1. Buat atau buka AWS `credentials` file bersama. File ini ada `~/.aws/credentials` di sistem Linux dan macOS, dan `%USERPROFILE%\.aws\credentials` di Windows. Untuk informasi selengkapnya, lihat [Lokasi File Kredensial](#).
2. Tambahkan teks berikut ke `credentials` file bersama. Ganti nilai ID contoh dan nilai kunci contoh dengan nilai dalam .csv file yang Anda download sebelumnya.

```
[default]
aws_access_key_id = AKIAIOSFODNN7EXAMPLE
aws_secret_access_key = wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
```

3. Simpan file tersebut.

`credentials` file bersama adalah cara paling umum untuk menyimpan kredensial. Ini juga dapat ditetapkan sebagai variabel lingkungan, lihat [AWS kunci akses](#) untuk nama variabel lingkungan. Ini adalah cara untuk membantu Anda memulai, tetapi kami sarankan Anda beralih ke Pusat IAM Identitas atau kredensi sementara lainnya sesegera mungkin. Setelah Anda beralih dari penggunaan kredensi jangka panjang, ingatlah untuk menghapus kredensial ini dari file bersama `credentials`.

Menggunakan IAM peran untuk EC2 instans Amazon

Contoh ini mencakup pengaturan AWS Identity and Access Management peran dengan akses Amazon S3 untuk digunakan dalam aplikasi yang diterapkan ke instans Amazon. EC2

Untuk menjalankan AWS SDK aplikasi Anda di instans Amazon Elastic Compute Cloud, buat IAM peran, lalu berikan akses EC2 instans Amazon Anda ke peran tersebut. Untuk informasi selengkapnya, lihat [IAM Peran untuk Amazon EC2](#) di Panduan EC2 Pengguna Amazon.

Buat IAM peran

AWS SDK Aplikasi yang Anda kembangkan kemungkinan mengakses setidaknya satu Layanan AWS untuk melakukan tindakan. Buat IAM peran yang memberikan izin yang diperlukan untuk menjalankan aplikasi Anda.

Prosedur ini membuat peran yang memberikan akses hanya-baca ke Amazon S3 sebagai contoh. Banyak AWS SDK panduan memiliki tutorial “memulai” yang dibaca dari Amazon S3.

1. Masuk ke AWS Management Console dan buka IAM konsol di <https://console.aws.amazon.com/iam/>.
2. Di panel navigasi, pilih Peran, lalu pilih Buat peran.
3. Untuk Pilih entitas tepercaya, di bawah Jenis entitas tepercaya, pilih Layanan AWS.
4. Di bawah Kasus penggunaan, pilih Amazon EC2, lalu pilih Berikutnya.
5. Untuk Menambahkan izin, pilih kotak centang untuk Amazon S3 Read Only Access dari daftar kebijakan, lalu pilih Berikutnya.
6. Masukkan nama untuk peran tersebut, lalu pilih Buat peran. Ingat nama ini karena Anda akan membutuhkannya saat membuat EC2 instance Amazon Anda.

Luncurkan EC2 instans Amazon dan tentukan IAM peran Anda

Anda dapat membuat dan meluncurkan EC2 instans Amazon menggunakan IAM peran Anda dengan melakukan hal berikut:

- Ikuti [Luncurkan instance dengan cepat](#) di Panduan EC2 Pengguna Amazon. Namun, sebelum langkah pengajuan akhir, lakukan juga hal berikut:
 - Di bawah Detail lanjutan, untuk profil IAM Instans, pilih peran yang Anda buat di langkah sebelumnya.

Dengan EC2 pengaturan ini IAM dan Amazon, Anda dapat menerapkan aplikasi Anda ke EC2 instans Amazon dan aplikasi Anda akan memiliki akses baca ke layanan Amazon S3.

Connect ke EC2 instance

Connect ke EC2 instans Amazon sehingga Anda dapat mentransfer aplikasi Anda ke sana dan kemudian menjalankan aplikasi. Anda akan memerlukan file yang berisi bagian pribadi dari key pair yang Anda gunakan di bawah Key pair (login) ketika Anda membuat instance Anda; yaitu, PEM file.

Anda dapat melakukannya dengan mengikuti panduan untuk jenis instans Anda: [Connect to your Linux instance](#) or [Connect to your Windows instance](#). Ketika Anda terhubung, lakukan sedemikian rupa sehingga Anda dapat mentransfer file dari mesin pengembangan Anda ke instans Anda.

Note

Di terminal Linux atau macOS, Anda dapat menggunakan perintah salin aman untuk menyalin aplikasi Anda. Untuk menggunakan scp dengan key pair, Anda dapat menggunakan perintah berikut: `scp -i path/to/key file/to/copy ec2-user@ec2-xx-xx-xxx-xxx.compute.amazonaws.com:~`.

Untuk informasi selengkapnya untuk Windows, lihat [Mentransfer file ke instance Windows](#).

Jika Anda menggunakan AWS Toolkit, Anda sering juga dapat terhubung ke instance dengan menggunakan Toolkit. Untuk informasi selengkapnya, lihat panduan pengguna khusus untuk Toolkit yang Anda gunakan.

Jalankan aplikasi Anda pada EC2 instance

1. Salin file aplikasi Anda dari drive lokal Anda ke EC2 instans Amazon Anda.
2. Mulai aplikasi dan verifikasi bahwa itu berjalan dengan hasil yang sama seperti pada mesin pengembangan Anda.
3. (Opsional) Verifikasi bahwa aplikasi menggunakan kredensial yang disediakan oleh peran IAM
 - a. Masuk ke AWS Management Console dan buka EC2 konsol Amazon di <https://console.aws.amazon.com/ec2/>.
 - b. Pilih instans.
 - c. Pilih Tindakan, Keamanan, lalu pilih Ubah IAM peran.

- d. Untuk IAM peran, lepaskan IAM peran dengan memilih No IAM Role.
- e. Pilih Perbarui IAM peran.
- f. Jalankan aplikasi lagi dan konfirmasi bahwa ia mengembalikan kesalahan otorisasi.

Referensi pengaturan

SDKs menyediakan bahasa khusus APIs untuk. Layanan AWS Mereka menangani beberapa pekerjaan berat yang diperlukan dalam berhasil melakukan API panggilan, termasuk otentikasi, perilaku coba lagi, dan banyak lagi. Untuk melakukan ini, SDKs memiliki strategi yang fleksibel untuk mendapatkan kredensial untuk digunakan untuk permintaan Anda, untuk mempertahankan pengaturan untuk digunakan dengan setiap layanan, dan untuk mendapatkan nilai yang akan digunakan untuk pengaturan global.

Anda dapat menemukan informasi terperinci tentang pengaturan konfigurasi di bagian berikut:

- [AWS SDKs dan Penyedia kredensi standar Alat](#)— Penyedia kredensi umum distandarisasi di beberapa. SDKs
- [AWS SDKs dan fitur standar Alat](#)— Fitur umum distandarisasi di beberapa. SDKs

Membuat klien layanan

Untuk mengakses secara terprogram Layanan AWS, SDKs gunakan kelas/objek klien untuk masing-masing. Layanan AWS Misalnya, jika aplikasi Anda perlu mengakses AmazonEC2, aplikasi Anda akan membuat objek EC2 klien Amazon untuk berinteraksi dengan layanan tersebut. Anda kemudian menggunakan klien layanan untuk membuat permintaan untuk itu Layanan AWS. Sebagian besar SDKs, objek klien layanan tidak dapat diubah, jadi Anda harus membuat klien baru untuk setiap layanan yang Anda minta dan untuk membuat permintaan ke layanan yang sama menggunakan konfigurasi yang berbeda.

Prioritas pengaturan

Pengaturan global mengonfigurasi fitur, penyedia kredensi, dan fungsionalitas lain yang didukung oleh sebagian besar SDKs dan memiliki dampak luas. Layanan AWS Semua SDKs memiliki serangkaian tempat (atau sumber) yang mereka periksa untuk menemukan nilai untuk pengaturan global. Berikut ini adalah prioritas pencarian pengaturan:

1. Pengaturan eksplisit apa pun yang disetel dalam kode atau pada klien layanan itu sendiri lebih diutamakan daripada yang lain.
 - Beberapa pengaturan dapat diatur berdasarkan per-operasi, dan dapat diubah sesuai kebutuhan untuk setiap operasi yang Anda panggil. Untuk AWS CLI atau AWS Tools for PowerShell,

ini mengambil bentuk parameter per operasi yang Anda masukkan pada baris perintah.

Untuk SDK, tugas eksplisit dapat mengambil bentuk parameter yang Anda tetapkan ketika Anda membuat instance Layanan AWS klien atau objek konfigurasi, atau kadang-kadang ketika Anda memanggil individu. API

2. Hanya Java/Kotlin: Properti JVM sistem untuk pengaturan dicentang. Jika disetel, nilai itu digunakan untuk mengkonfigurasi klien.
3. Variabel lingkungan diperiksa. Jika disetel, nilai itu digunakan untuk mengkonfigurasi klien.
4. SDK memeriksa `credentials` file bersama untuk pengaturan. Jika sudah diatur, klien menggunakannya.
5. `configFile` bersama untuk pengaturan. Jika pengaturan hadir, SDK menggunakannya.
 - Variabel `AWS_PROFILE` lingkungan atau properti `aws.profile` JVM sistem dapat digunakan untuk menentukan profil mana yang SDK dimuat.
6. Setiap nilai default yang disediakan oleh kode SDK sumber itu sendiri digunakan terakhir.

Note

Beberapa SDKs dan alat mungkin memeriksa dalam urutan yang berbeda. Juga, beberapa SDKs dan alat mendukung metode lain untuk menyimpan dan mengambil parameter. Misalnya, AWS SDK for .NET mendukung sumber tambahan yang disebut [SDKStore](#). Untuk informasi selengkapnya tentang penyedia yang unik untuk alat SDK atau, lihat panduan khusus untuk SDK atau alat yang Anda gunakan.

Urutan menentukan metode mana yang diutamakan dan mengesampingkan yang lain. Misalnya, jika Anda mengatur profil di `config` file bersama, itu hanya ditemukan dan digunakan setelah alat SDK atau memeriksa tempat lain terlebih dahulu. Ini berarti bahwa jika Anda meletakkan pengaturan dalam `credentials` file, itu digunakan sebagai pengganti yang ditemukan dalam `config` file. Jika Anda mengonfigurasi variabel lingkungan dengan pengaturan dan nilai, itu akan menimpa pengaturan itu di `config` file `credentials` dan file. Dan akhirnya, pengaturan pada operasi individu (parameter AWS CLI baris perintah atau API parameter) atau dalam kode akan mengesampingkan semua nilai lain untuk satu perintah itu.

Halaman pengaturan

Halaman-halaman dalam bagian referensi Pengaturan dari panduan ini merinci pengaturan yang tersedia yang dapat diatur melalui berbagai mekanisme. Tabel berikut mencantumkan pengaturan file konfigurasi dan kredensi, variabel lingkungan, dan (untuk Java dan Kotlin SDKs) JVM setelan yang dapat digunakan di luar kode Anda untuk mengonfigurasi fitur. Setiap topik tertaut di setiap daftar membawa Anda ke halaman pengaturan yang sesuai.

- [Config](#) daftar pengaturan file
- [Credentials](#) daftar pengaturan file
- [Daftar variabel lingkungan](#)
- [JVM](#) daftar properti sistem

Setiap penyedia kredensi atau fitur memiliki halaman di mana pengaturan yang digunakan untuk mengonfigurasi fungsionalitas tersebut terdaftar. Untuk setiap pengaturan, Anda sering dapat mengatur nilai baik dengan menambahkan pengaturan ke file konfigurasi, atau dengan menyetel variabel lingkungan, atau (hanya untuk Java dan Kotlin) dengan menyetel properti JVM sistem. Setiap pengaturan mencantumkan semua metode yang didukung untuk mengatur nilai dalam blok di atas detail deskripsi. Meskipun [prioritas](#) bervariasi, fungsionalitas yang dihasilkan adalah sama terlepas dari bagaimana Anda mengaturnya.

Deskripsi akan mencakup nilai default, jika ada, yang berlaku jika Anda tidak melakukan apa-apa. Ini juga mendefinisikan apa nilai yang valid untuk pengaturan itu.

Sebagai contoh, mari kita lihat pengaturan dari halaman [Minta kompresi](#) fitur.

Informasi pengaturan `disable_request_compression` contoh mengkomunikasikan hal-hal berikut:

- Ada tiga cara yang setara untuk mengontrol kompresi permintaan di luar basis kode Anda. Anda dapat:
 - Setel di file konfigurasi Anda menggunakan `disable_request_compression`
 - Tetapkan sebagai variabel lingkungan menggunakan `AWS_DISABLE_REQUEST_COMPRESSION`
 - Atau, jika Anda menggunakan Java atau Kotlin SDK, atur sebagai properti JVM sistem menggunakan `aws.disableRequestCompression`

Note

Mungkin juga ada cara untuk mengonfigurasi fungsionalitas yang sama secara langsung dalam kode Anda, tetapi Referensi ini tidak mencakup ini karena unik untuk masing-masing SDK. Jika Anda ingin mengatur konfigurasi Anda dalam kode itu sendiri, lihat SDK panduan atau API referensi khusus Anda.

- Jika Anda tidak melakukan apa-apa, nilainya akan default ke `false`.
- Satu-satunya nilai yang valid untuk pengaturan Boolean ini adalah `true` dan `false`.

Di bagian bawah setiap halaman fitur ada Kompatibilitas dengan AWS SDKs tabel.

Tabel ini menunjukkan apakah Anda SDK mendukung pengaturan yang tercantum pada halaman. `Supported` Kolom menunjukkan tingkat dukungan dengan nilai-nilai berikut:

- **Yes**— Pengaturan sepenuhnya didukung oleh SDK seperti yang tertulis.
- **Partial**— Beberapa pengaturan didukung atau perilaku menyimpang dari deskripsi. Untuk `Partial`, catatan tambahan menunjukkan penyimpangan.
- **No**— Tidak ada pengaturan yang didukung. Ini tidak membuat klaim apakah fungsionalitas yang sama dapat dicapai dalam kode; itu hanya menunjukkan bahwa pengaturan konfigurasi eksternal yang terdaftar tidak didukung.

Configdaftar pengaturan file

Pengaturan yang tercantum dalam tabel berikut dapat ditetapkan dalam AWS config file bersama. Mereka global dan mempengaruhi semua Layanan AWS. SDK dan alat juga dapat mendukung pengaturan unik dan variabel lingkungan. Untuk melihat pengaturan dan variabel lingkungan yang hanya didukung oleh individu SDK atau alat, lihat panduan khusus SDK atau alat tersebut.

Nama pengaturan	Detail
account_id_endpoint_mode	Titik akhir berbasis akun

Nama pengaturan	Detail
api_versions	Pengaturan konfigurasi umum
aws_access_key_id	AWS kunci akses
aws_account_id	Titik akhir berbasis akun
aws_secret_access_key	AWS kunci akses
aws_session_token	AWS kunci akses
ca_bundle	Pengaturan konfigurasi umum
credential_process	Penyedia kredensi proses
credential_source	Asumsikan penyedia kredensi peran
defaults_mode	Default konfigurasi cerdas
disable_request_compression	Minta kompresi
duration_seconds	Asumsikan penyedia kredensi peran
ec2_metadata_service_endpoint	IMDS penyedia kredensi
ec2_metadata_service_endpoint_mode	IMDS penyedia kredensi

Nama pengaturan	Detail
ec2_metadata_v1_disabled	IMDS penyedia kredensi
endpoint_discovery_enabled	Penemuan titik akhir
endpoint_url	Titik akhir khusus layanan
external_id	Asumsikan penyedia kredensi peran
ignore_configured_endpoint_urls	Titik akhir khusus layanan
max_attempts	Coba lagi perilaku
metadata_service_number_attempts	EC2 Metadata contoh Amazon
metadata_service_timeout	EC2 Metadata contoh Amazon
mfa_serial	Asumsikan penyedia kredensi peran
output	Pengaturan konfigurasi umum
parameter_validation	Pengaturan konfigurasi umum
region	Wilayah AWS

Nama pengaturan	Detail
request_m in_compre ssion_siz e_bytes	Minta kompresi
retry_mode	Coba lagi perilaku
role_arn	Asumsikan penyedia kredensi peran
role_sess ion_name	Asumsikan penyedia kredensi peran
s3_disabl e_multire gion_acce ss_points	Titik Akses Multi-Wilayah Amazon S3
s3_use_ar n_region	Titik akses Amazon S3
sdk_ua_app_id	ID Aplikasi
source_profile	Asumsikan penyedia kredensi peran
sso_account_id	IAM Penyedia kredensi Pusat Identitas
sso_region	IAM Penyedia kredensi Pusat Identitas
sso_regis tration_scopes	IAM Penyedia kredensi Pusat Identitas
sso_role_name	IAM Penyedia kredensi Pusat Identitas
sso_start_url	IAM Penyedia kredensi Pusat Identitas
sts_regio nal_endpoints	AWS STS Titik akhir regional

Nama pengaturan	Detail
use_duals tack_endpoint	Tumpukan ganda dan titik akhir FIPS
use_fips_ endpoint	Tumpukan ganda dan titik akhir FIPS
web_ident ity_token_file	Asumsikan penyedia kredensi peran

Credentialsdaftar pengaturan file

Pengaturan yang tercantum dalam tabel berikut dapat ditetapkan dalam AWS `credentials` file bersama. Mereka global dan mempengaruhi semua Layanan AWS. SDKsdan alat juga dapat mendukung pengaturan unik dan variabel lingkungan. Untuk melihat pengaturan dan variabel lingkungan yang hanya didukung oleh individu SDK atau alat, lihat panduan khusus SDK atau alat tersebut.

Nama pengaturan	Detail
aws_acces s_key_id	AWS kunci akses
aws_secre t_access_key	AWS kunci akses
aws_sessi on_token	AWS kunci akses

Daftar variabel lingkungan

Variabel lingkungan yang didukung oleh sebagian besar SDKs tercantum dalam tabel berikut. Mereka global dan mempengaruhi semua Layanan AWS. SDKsdan alat juga dapat mendukung pengaturan unik dan variabel lingkungan. Untuk melihat pengaturan dan variabel lingkungan yang hanya didukung oleh individu SDK atau alat, lihat panduan khusus SDK atau alat tersebut.

Nama pengaturan	Detail
AWS_ACCESS_KEY_ID	AWS kunci akses
AWS_ACCOUNT_ID	Titik akhir berbasis akun
AWS_ACCOUNT_ID_ENDPOINT_MODE	Titik akhir berbasis akun
AWS_CA_BUNDLE	Pengaturan konfigurasi umum
AWS_CONFIG_FILE	Lokasi yang dibagikan config dan credentials file
AWS_CONTAINER_AUTHORIZATION_TOKEN	Penyedia kredensi kontainer
AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE	Penyedia kredensi kontainer
AWS_CONTAINER_CREDENTIALS_FULL_URI	Penyedia kredensi kontainer
AWS_CONTAINER_CREDENTIALS_RELATIVE_URI	Penyedia kredensi kontainer
AWS_DEFAULTS_MODE	Default konfigurasi cerdas

Nama pengaturan	Detail
AWS_DISABLE_REQUEST_COMPRESSION	Minta kompresi
AWS_EC2_METADATA_DISABLED	IMDS penyedia kredensi
AWS_EC2_METADATA_SERVICE_ENDPOINT	IMDS penyedia kredensi
AWS_EC2_METADATA_SERVICE_ENDPOINT_MODE	IMDS penyedia kredensi
AWS_EC2_METADATA_V1_DISABLED	IMDS penyedia kredensi
AWS_ENABLE_ENDPOINT_DISCOVERY	Penemuan titik akhir
AWS_ENDPOINT_URL	Titik akhir khusus layanan
AWS_ENDPOINT_URL_<SERVICE>	Titik akhir khusus layanan

Nama pengaturan	Detail
AWS_IGNORE_CONFIGURED_ENDPOINT_URLS	Titik akhir khusus layanan
AWS_MAX_AVAILABLE_THREADS	Coba lagi perilaku
AWS_METADATA_SERVICE_NUM_ATTEMPTS	EC2Metadata contoh Amazon
AWS_METADATA_SERVICE_TIMEOUT	EC2Metadata contoh Amazon
AWS_PROFILE	Berbagi config dan credentials file
AWS_REGION	Wilayah AWS
AWS_REQUEST_MIN_COMPRESSION_SIZE_BYTES	Minta kompresi
AWS_RETRY_MODE	Coba lagi perilaku
AWS_ROLE_ARN	Asumsikan penyedia kredensi peran
AWS_ROLE_SESSION_NAME	Asumsikan penyedia kredensi peran
AWS_S3_MULTIREGION_ACCESS_POINTS	Titik Akses Multi-Wilayah Amazon S3

Nama pengaturan	Detail
AWS_S3_US E_ARN_REGION	Titik akses Amazon S3
AWS_SDK_U A_APP_ID	ID Aplikasi
AWS_SECRE T_ACCESS_KEY	AWS kunci akses
AWS_SESSI ON_TOKEN	AWS kunci akses
AWS_SHARE D_CREDENT IALS_FILE	Lokasi yang dibagikan config dan credentials file
AWS_STS_R EGIONAL_E NDPOINTS	AWS STS Titik akhir regional
AWS_USE_D UALSTACK_ ENDPOINT	Tumpukan ganda dan titik akhir FIPS
AWS_USE_F IPS_ENDPOINT	Tumpukan ganda dan titik akhir FIPS
AWS_WEB_I DENTITY_T OKEN_FILE	Asumsikan penyedia kredensi peran

JVMdaftar properti sistem

Anda dapat menggunakan properti JVM sistem berikut untuk AWS SDK for Java dan AWS SDK for Kotlin (menargetkan JVM). Lihat [the section called “Cara mengatur properti sistem JVM”](#) petunjuk tentang cara mengatur properti JVM sistem.

Nama pengaturan	Detail
<code>aws.accessKeyId</code>	AWS kunci akses
<code>aws.accountId</code>	Titik akhir berbasis akun
<code>aws.accountIdEndpointMode</code>	Titik akhir berbasis akun
<code>aws.configFile</code>	Lokasi yang dibagikan config dan credentials file
<code>aws.defaultsMode</code>	Default konfigurasi cerdas
<code>aws.disableEc2MetadataV1</code>	IMDS penyedia kredensi
<code>aws.disableRequestCompression</code>	Minta kompresi
<code>aws.ec2MetadataServiceEndpoint</code>	IMDS penyedia kredensi
<code>aws.ec2MetadataServiceEndpointMode</code>	IMDS penyedia kredensi
<code>aws.endpointDiscoveryEnabled</code>	Penemuan titik akhir
<code>aws.endpointUrl</code>	Titik akhir khusus layanan

Nama pengaturan	Detail
<code>aws.endpointUrl<ServiceName></code>	Titik akhir khusus layanan
<code>aws.ignoreConfiguredEndpointUrls</code>	Titik akhir khusus layanan
<code>aws.maxAttempts</code>	Coba lagi perilaku
<code>aws.profile</code>	Berbagi config dan credentials file
<code>aws.region</code>	Wilayah AWS
<code>aws.requestMinCompressionSizeBytes</code>	Minta kompresi
<code>aws.retryMode</code>	Coba lagi perilaku
<code>aws.roleArn</code>	Asumsikan penyedia kredensi peran
<code>aws.roleSessionName</code>	Asumsikan penyedia kredensi peran
<code>aws.s3DisableMultiRegionAccessPoints</code>	Titik Akses Multi-Wilayah Amazon S3
<code>aws.s3UseArnRegion</code>	Titik akses Amazon S3
<code>aws.secretAccessKey</code>	AWS kunci akses

Nama pengaturan	Detail
<code>aws.sessionToken</code>	AWS kunci akses
<code>aws.sharedCredentialsFile</code>	Lokasi yang dibagikan config dan credentials file
<code>aws.useDualstackEndpoint</code>	Tumpukan ganda dan titik akhir FIPS
<code>aws.useFipsEndpoint</code>	Tumpukan ganda dan titik akhir FIPS
<code>aws.userAgentAppId</code>	ID Aplikasi
<code>aws.webIdentityTokenFile</code>	Asumsikan penyedia kredensi peran

AWS SDKsdan Penyedia kredensi standar Alat

Banyak penyedia kredensi telah distandarisasi untuk default yang konsisten dan bekerja dengan cara yang sama di banyak. SDKs Konsistensi ini meningkatkan produktivitas dan kejelasan saat pengkodean di beberapa SDKs. Semua pengaturan dapat diganti dalam kode. Untuk detailnya, lihat spesifik Anda SDKAPI.

Important

Tidak semua SDKs mendukung semua penyedia, atau bahkan semua aspek dalam penyedia.

Topik

- [Memahami rantai penyedia kredensi](#)

- [SDK-rantai penyedia kredensi khusus dan khusus alat](#)
- [AWS kunci akses](#)
- [Asumsikan penyedia kredensi peran](#)
- [Penyedia kredensi kontainer](#)
- [IAM Penyedia kredensi Pusat Identitas](#)
- [IMDS penyedia kredensi](#)
- [Penyedia kredensi proses](#)

Memahami rantai penyedia kredensi

Semua SDKs memiliki serangkaian tempat (atau sumber) yang mereka periksa untuk menemukan kredensial yang valid untuk digunakan untuk membuat permintaan ke file. Layanan AWS Setelah kredensial yang valid ditemukan, pencarian dihentikan. Pencarian sistematis ini disebut rantai penyedia kredensi.

Saat menggunakan salah satu penyedia kredensi standar, AWS SDKs selalu berusaha untuk memperbarui kredensial secara otomatis ketika mereka kedaluwarsa. Rantai penyedia kredensi bawaan memberi aplikasi Anda kemampuan untuk menyegarkan kredensial Anda terlepas dari penyedia mana yang Anda gunakan dalam rantai tersebut. Tidak ada kode tambahan yang diperlukan SDK untuk melakukan ini.

Meskipun rantai berbeda yang digunakan oleh masing-masing SDK bervariasi, mereka paling sering mencakup sumber-sumber seperti berikut:

Penyedia kredensi	Deskripsi
AWS kunci akses	AWS kunci akses untuk IAM pengguna (seperti <code>AWS_ACCESS_KEY_ID</code> , dan <code>AWS_SECRET_ACCESS_KEY</code>).
Bersekutu dengan identitas web atau OpenID Connect - Asumsikan penyedia kredensi peran	Masuk menggunakan penyedia identitas eksternal (IDP) yang terkenal, seperti Login with Amazon, Facebook, Google, atau IDP lain yang kompatibel dengan OpenID Connect (OIDC). Asumsikan izin IAM peran menggunakan JSON Web Token (JWT) from AWS Security Token Service (AWS STS).

Penyedia kredensi	Deskripsi
IAM Penyedia kredensi Pusat Identitas	Dapatkan kredensial dari AWS IAM Identity Center
Asumsikan penyedia kredensi peran	Dapatkan akses ke sumber daya lain dengan mengasumsikan izin peran IAM. (Ambil dan kemudian gunakan kredensi sementara untuk peran).
Penyedia kredensi kontainer	Amazon Elastic Container Service (Amazon ECS) dan kredensi Amazon Elastic Kubernetes Service (Amazon EKS) Penyedia kredensi kontainer mengambil kredensi untuk aplikasi kontainer pelanggan.
Penyedia kredensi proses	Penyedia kredensi khusus. Dapatkan kredensial Anda dari sumber atau proses eksternal, termasuk IAM Peran Di Mana Saja.
IMDS penyedia kredensi	Kredensi profil instans Amazon Elastic Compute Cloud (Amazon EC2). Kaitkan IAM peran dengan masing-masing EC2 instance Anda. Kredensi sementara untuk peran itu tersedia untuk kode yang berjalan dalam instance. Kredensialnya dikirimkan melalui layanan EC2 metadata Amazon.

Untuk setiap langkah dalam rantai, ada beberapa cara untuk menetapkan nilai pengaturan. Menetapkan nilai yang ditentukan dalam kode selalu diutamakan. Namun, ada juga [Variabel-variabel lingkungan](#) dan [Berbagi config dan credentials file](#). Untuk informasi selengkapnya, lihat [Prioritas pengaturan](#).

SDK-rantai penyedia kredensi khusus dan khusus alat

Untuk langsung menuju ke detail rantai penyedia kredensi khusus atau alat Anda SDK, pilih SDK atau alat Anda dari yang berikut ini:

- [AWS CLI](#)
- [SDK untuk C++](#)

- [SDK untuk Go](#)
- [SDK untuk Java](#)
- [SDK untuk JavaScript](#)
- [SDK untuk Kotlin](#)
- [SDK untuk .NET](#)
- [SDK untuk PHP](#)
- [SDK untuk Python \(Boto3\)](#)
- [SDK untuk Ruby](#)
- [SDK untuk Rust](#)
- [SDK untuk Swift](#)
- [Alat untuk PowerShell](#)

AWS kunci akses

Warning

Untuk menghindari risiko keamanan, jangan gunakan IAM pengguna untuk otentikasi saat mengembangkan perangkat lunak yang dibuat khusus atau bekerja dengan data nyata. Sebaliknya, gunakan federasi dengan penyedia identitas seperti [AWS IAM Identity Center](#).

AWS kunci akses untuk IAM pengguna dapat digunakan sebagai AWS kredensial Anda. AWS SDK secara otomatis menggunakan AWS kredensial ini untuk menandatangani API permintaan AWS, sehingga beban kerja Anda dapat mengakses AWS sumber daya dan data Anda dengan aman dan nyaman. Disarankan untuk selalu menggunakan `aws_session_token` sehingga kredensialnya bersifat sementara dan tidak lagi berlaku setelah kedaluwarsa. Menggunakan kredensial jangka panjang tidak disarankan.

Note

Jika AWS tidak dapat menyegarkan kredensial sementara ini, AWS dapat memperpanjang validitas kredensialnya sehingga beban kerja Anda tidak terpengaruh.

AWS `credentials` file bersama adalah lokasi yang disarankan untuk menyimpan informasi kredensial karena aman di luar direktori sumber aplikasi dan terpisah dari pengaturan SDK -spesifik file bersama. `config`

Untuk mempelajari lebih lanjut tentang AWS kredensial dan menggunakan kunci akses, lihat [kredensial AWS keamanan](#) dan [Mengelola kunci akses untuk IAM pengguna di Panduan Pengguna](#). IAM

Konfigurasi fungsi ini dengan menggunakan yang berikut:

aws_access_key_id- Pengaturan AWS `config` file bersama, **aws_access_key_id**- Pengaturan AWS `credentials` file bersama (metode yang disarankan), **AWS_ACCESS_KEY_ID**- variabel lingkungan, **aws.accessKeyId**- properti JVM sistem: Hanya Java/Kotlin

Menentukan kunci AWS akses yang digunakan sebagai bagian dari kredensial untuk mengautentikasi pengguna.

aws_secret_access_key- Pengaturan AWS `config` file bersama, **aws_secret_access_key**- Pengaturan AWS `credentials` file bersama (metode yang disarankan),

AWS_SECRET_ACCESS_KEY- variabel lingkungan, **aws.secretAccessKey**- properti JVM sistem: Hanya Java/Kotlin

Menentukan kunci AWS rahasia yang digunakan sebagai bagian dari kredensial untuk mengautentikasi pengguna.

aws_session_token- Pengaturan AWS `config` file bersama, **aws_session_token**- Pengaturan AWS `credentials` file bersama (metode yang disarankan), **AWS_SESSION_TOKEN**- variabel lingkungan, **aws.sessionToken**- properti JVM sistem: Hanya Java/Kotlin

Menentukan token AWS sesi yang digunakan sebagai bagian dari kredensial untuk mengautentikasi pengguna. Anda menerima nilai ini sebagai bagian dari kredensial sementara yang dikembalikan oleh permintaan yang berhasil untuk mengambil peran. Token sesi hanya diperlukan jika Anda secara manual menentukan kredensial keamanan sementara. Namun, kami menyarankan Anda untuk selalu menggunakan kredensial keamanan sementara alih-alih kredensial jangka panjang. Untuk rekomendasi keamanan, lihat [Praktik terbaik keamanan di IAM](#).

Untuk petunjuk tentang cara mendapatkan nilai-nilai ini, lihat [Otentikasi menggunakan kredensial jangka pendek](#).

Contoh pengaturan nilai-nilai yang diperlukan ini dalam `credentials` file `config` atau:

```
[default]
```

```
aws_access_key_id = AKIAIOSFODNN7EXAMPLE
aws_secret_access_key = wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
aws_session_token = AQoEXAMPLEH4aoAH0gNCAPy...truncated...zrkuWJ0gQs8IZZaIv2BXIa2R40lgk
```

Linux/macOS contoh pengaturan variabel lingkungan melalui baris perintah:

```
export AWS_ACCESS_KEY_ID=AKIAIOSFODNN7EXAMPLE
export AWS_SECRET_ACCESS_KEY=wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
export
  AWS_SESSION_TOKEN=AQoEXAMPLEH4aoAH0gNCAPy...truncated...zrkuWJ0gQs8IZZaIv2BXIa2R40lgk
```

Contoh Windows pengaturan variabel lingkungan melalui baris perintah:

```
setx AWS_ACCESS_KEY_ID AKIAIOSFODNN7EXAMPLE
setx AWS_SECRET_ACCESS_KEY wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
setx
  AWS_SESSION_TOKEN AQoEXAMPLEH4aoAH0gNCAPy...truncated...zrkuWJ0gQs8IZZaIv2BXIa2R40lgk
```

Kompatibilitas dengan AWS SDKs

Berikut ini SDKs mendukung fitur dan pengaturan yang dijelaskan dalam topik ini. Setiap pengecualian sebagian dicatat. Setiap pengaturan properti JVM sistem didukung oleh AWS SDK for Java dan AWS SDK for Kotlin satu-satunya.

SDK	Di Catatan atau informasi lebih lanjut
AWS CLI v2	Ya
SDK untuk C ++	Ya configfile bersama tidak didukung.
SDK untuk Go V2 (1.x)	Ya
SDK untuk Go 1.x (V1)	Ya Untuk menggunakan pengaturan config file bersama, Anda harus mengaktifkan pemuatan dari file konfigurasi; lihat Sesi .
SDK untuk Java 2.x	Ya
SDK untuk Java 1.x	Ya
SDK untuk JavaScript 3.x	Ya

SDK	Di	Catatan atau informasi lebih lanjut
SDK untuk JavaScript 2.x	Ya	
SDK untuk Kotlin	Ya	
SDK untuk .NET 3.x	Ya	Variabel lingkungan tidak didukung.
SDK untuk PHP 3.x	Ya	
SDK untuk Python (Boto3)	Ya	
SDK untuk Ruby 3.x	Ya	
SDK untuk Rust	Ya	
SDK untuk Swift	Ya	
Alat untuk PowerShell	Ya	Variabel lingkungan tidak didukung.

Asumsikan penyedia kredensi peran

Dengan asumsi peran melibatkan penggunaan seperangkat kredensial keamanan sementara untuk mengakses AWS sumber daya yang mungkin tidak dapat Anda akses sebaliknya. Kredensial sementara ini terdiri dari access key ID, secret access key, dan token keamanan.

Untuk mengatur SDK atau alat Anda untuk mengambil peran, Anda harus terlebih dahulu membuat atau mengidentifikasi peran tertentu untuk diasumsikan. IAM peran diidentifikasi secara unik oleh peran Amazon Resource Name (ARN). Peran membangun hubungan kepercayaan dengan entitas lain. Entitas tepercaya yang menggunakan peran mungkin Layanan AWS, yang lain Akun AWS, penyedia identitas web atau OIDC, atau SAML federasi.

Setelah IAM peran diidentifikasi, jika Anda dipercaya oleh peran tersebut, Anda dapat mengonfigurasi SDK atau alat untuk menggunakan izin yang diberikan oleh peran tersebut. Untuk melakukan ini, gunakan pengaturan berikut.

Untuk panduan tentang mulai menggunakan pengaturan ini, lihat [Asumsikan peran dengan AWS kredensial](#) di panduan ini.

Asumsikan pengaturan penyedia kredensi peran

Konfigurasi fungsi ini dengan menggunakan yang berikut:

credential_source- Pengaturan AWS **config** file bersama

Digunakan dalam EC2 instans Amazon atau penampung Amazon Elastic Container Service untuk menentukan di mana alat SDK atau dapat menemukan kredensial yang memiliki izin untuk mengambil peran yang Anda tentukan dengan parameter. `role_arn`

Nilai default: Tidak ada

Nilai yang valid:

- Lingkungan - [Menentukan bahwa SDK atau alat adalah untuk mengambil kredensial sumber dari variabel lingkungan dan. `AWS_ACCESS_KEY_ID` `AWS_SECRET_ACCESS_KEY`](#)
- Ec2 InstanceMetadata - Menentukan bahwa SDK atau alat adalah untuk menggunakan [IAMperan yang dilampirkan ke profil EC2 instance untuk](#) mendapatkan kredensial sumber.
- EcsContainer— Menentukan bahwa SDK atau alat adalah untuk menggunakan [IAMperan yang dilampirkan ke ECS wadah untuk](#) mendapatkan kredensial sumber.

Anda tidak dapat menentukan keduanya `credential_source` dan `source_profile` di profil yang sama.

Contoh pengaturan ini dalam `config` file untuk menunjukkan bahwa kredensial harus bersumber dari Amazon: EC2

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

duration_seconds- Pengaturan AWS **config** file bersama

Menentukan durasi maksimum sesi peran, dalam hitungan detik.

Pengaturan ini hanya berlaku ketika profil menentukan untuk mengambil peran.

Nilai default: 3600 detik (satu jam)

Nilai yang valid: Nilai dapat berkisar dari 900 detik (15 menit) hingga pengaturan durasi sesi maksimum yang dikonfigurasi untuk peran (yang bisa maksimal 43200 detik, atau 12 jam). Untuk

informasi selengkapnya, [lihat Melihat Pengaturan Durasi Sesi Maksimum untuk Peran](#) di Panduan IAM Pengguna.

Contoh pengaturan ini dalam config file:

```
duration_seconds = 43200
```

external_id- Pengaturan AWS **config** file bersama

Menentukan pengenalan unik yang digunakan oleh pihak ketiga untuk mengambil peran dalam akun pelanggan mereka.

Setelan ini hanya berlaku jika profil menetapkan untuk mengambil peran dan kebijakan kepercayaan untuk peran tersebut memerlukan nilai untuk `ExternalId`. Nilai memetakan ke `ExternalId` parameter yang diteruskan ke `AssumeRole` operasi saat profil menentukan peran.

Nilai default: Tidak ada.

Nilai yang valid: Lihat [Cara menggunakan ID Eksternal Saat Memberikan Akses ke AWS Sumber Daya Anda kepada Pihak Ketiga](#) dalam Panduan IAM Pengguna.

Contoh pengaturan ini dalam config file:

```
external_id = unique_value_assigned_by_3rd_party
```

mfa_serial- Pengaturan AWS **config** file bersama

Menentukan identifikasi atau nomor seri perangkat otentikasi multi-faktor (MFA) yang harus digunakan pengguna saat mengambil peran.

Diperlukan saat mengasumsikan peran di mana kebijakan kepercayaan untuk peran tersebut mencakup kondisi yang memerlukan MFA otentikasi. Untuk informasi selengkapnya MFA, lihat [Autentikasi AWS multi-faktor IAM di IAM](#) Panduan Pengguna.

Nilai default: Tidak ada.

Nilai yang valid: Nilai dapat berupa nomor seri untuk perangkat keras (seperti `GAHT12345678`), atau Amazon Resource Name (ARN) untuk MFA perangkat virtual. Formatnya ARN adalah: `arn:aws:iam::account-id:mfa/mfa-device-name`

Contoh pengaturan ini dalam config file:

Contoh ini mengasumsikan MFA perangkat virtual, yang disebut `MyMFADevice`, yang telah dibuat untuk akun dan diaktifkan untuk pengguna.

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

role_arn- Pengaturan AWS **config** file bersama, **AWS_ROLE_ARN**- variabel lingkungan, **aws.roleArn**- properti JVM sistem: Hanya Java/Kotlin

Menentukan Amazon Resource Name (ARN) IAM peran yang ingin Anda gunakan untuk melakukan operasi yang diminta menggunakan profil ini.

Nilai default: Tidak ada.

Nilai yang valid: Nilai harus ARN IAM peran, diformat sebagai berikut:

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

Selain itu, Anda juga harus menentukan salah satu pengaturan berikut:

- **source_profile**— Untuk mengidentifikasi profil lain yang akan digunakan untuk menemukan kredensial yang memiliki izin untuk mengambil peran dalam profil ini.
- **credential_source**— Untuk menggunakan kredensial yang diidentifikasi oleh variabel lingkungan saat ini atau kredensial yang dilampirkan ke profil EC2 instans Amazon, atau instance penampung Amazon. ECS
- **web_identity_token_file** Untuk menggunakan penyedia identitas publik atau penyedia identitas yang kompatibel dengan OpenID Connect (OIDC) untuk pengguna yang telah diautentikasi dalam aplikasi seluler atau web.

role_session_name- Pengaturan AWS **config** file bersama, **AWS_ROLE_SESSION_NAME**- variabel lingkungan, **aws.roleSessionName**- properti JVM sistem: Hanya Java/Kotlin

Menentukan nama untuk melampirkan ke sesi peran. Nama ini muncul di AWS CloudTrail log untuk entri yang terkait dengan sesi ini, yang dapat berguna saat mengaudit. Untuk detailnya, lihat [CloudTrail userIdentity elemen](#) di Panduan AWS CloudTrail Pengguna.

Nilai default: Parameter opsional. Jika Anda tidak memberikan nilai ini, nama sesi akan dibuat secara otomatis jika profil mengambil peran.

Nilai yang valid: Disediakan untuk `RoleSessionName` parameter ketika AWS CLI atau AWS API memanggil `AssumeRole` operasi (atau operasi seperti `AssumeRoleWithWebIdentity` operasi) atas nama Anda. Nilai menjadi bagian dari pengguna peran yang diasumsikan Amazon Resource

Name (ARN) yang dapat Anda kueri, dan muncul sebagai bagian dari entri CloudTrail log untuk operasi yang dipanggil oleh profil ini.

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

Contoh pengaturan ini dalam config file:

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

source_profile- Pengaturan AWS **config** file bersama

Menentukan profil lain yang kredensialnya digunakan untuk mengambil peran yang ditentukan oleh `role_arn` pengaturan di profil asli. Untuk memahami bagaimana profil digunakan dalam file bersama AWS config dan credentials file, lihat [Berbagi config dan credentials file](#).

Jika Anda menentukan profil yang juga merupakan profil peran asumsi, setiap peran akan diasumsikan secara berurutan untuk menyelesaikan kredensialnya sepenuhnya. Rantai ini dihentikan ketika SDK menemukan profil dengan kredensial. Rantai peran membatasi sesi AWS CLI atau AWS API peran Anda hingga maksimal satu jam dan tidak dapat ditingkatkan. Untuk informasi selengkapnya, lihat [Istilah dan konsep peran](#) di Panduan IAM Pengguna.

Nilai default: Tidak ada.

Nilai yang valid: String teks yang terdiri dari nama profil yang ditentukan dalam credentials file config dan. Anda juga harus menentukan nilai untuk `role_arn` di profil saat ini.

Anda tidak dapat menentukan keduanya `credential_source` dan `source_profile` di profil yang sama.

Contoh pengaturan ini dalam file konfigurasi:

```
[profile A]  
source_profile = B  
role_arn = arn:aws:iam::123456789012:role/RoleA  
role_session_name = ProfileARoleSession  
  
[profile B]  
credential_process = ./aws_signing_helper credential-process --certificate /  
path/to/certificate --private-key /path/to/private-key --trust-anchor-  
arn arn:aws:rolesanywhere:region:account:trust-anchor/TA_ID --profile-  
arn arn:aws:rolesanywhere:region:account:profile/PROFILE_ID --role-arn  
arn:aws:iam::account:role/ROLE_ID
```

Pada contoh sebelumnya, A profil memberi tahu alat SDK atau untuk secara otomatis mencari kredensial untuk profil yang B ditautkan. Dalam hal ini, B profil menggunakan alat pembantu kredensial yang disediakan oleh [Peran IAM Di Mana Saja](#) untuk mendapatkan kredensial untuk file. AWS SDK Kredensi sementara tersebut kemudian digunakan oleh kode Anda untuk mengakses AWS sumber daya. Peran yang ditentukan harus memiliki kebijakan IAM izin terlampir yang memungkinkan kode yang diminta berjalan, seperti perintah Layanan AWS, atau API metode. Setiap tindakan yang diambil oleh profil A memiliki nama sesi peran yang disertakan dalam CloudTrail log.

Untuk contoh rantai peran kedua, konfigurasi berikut dapat digunakan jika Anda memiliki aplikasi di instans Amazon Elastic Compute Cloud, dan Anda ingin aplikasi tersebut mengambil peran lain.

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

[profile B]
credential_source=Ec2InstanceMetadata
```

Profil A akan menggunakan kredensial dari EC2 instans Amazon untuk mengambil peran yang ditentukan dan akan memperbarui kredensialnya secara otomatis.

web_identity_token_file- Pengaturan AWS **config** file bersama,

AWS_WEB_IDENTITY_TOKEN_FILE- variabel lingkungan, **aws.webIdentityTokenFile**- properti JVM sistem: Hanya Java/Kotlin

Menentukan path ke file yang berisi token akses dari [penyedia OAuth 2.0 yang didukung atau penyedia identitas OpenID Connect ID](#).

Pengaturan ini memungkinkan otentikasi dengan menggunakan penyedia federasi identitas web, seperti [Google](#), [Facebook](#), dan [Amazon](#), di antara banyak lainnya. Alat pengembang SDK atau memuat konten file ini dan meneruskannya sebagai `WebIdentityToken` argumen saat memanggil `AssumeRoleWithWebIdentity` operasi atas nama Anda.

Nilai default: Tidak ada.

Nilai yang valid: Nilai ini harus berupa jalur dan nama file. File harus berisi token akses OAuth 2.0 atau token OpenID Connect yang diberikan kepada Anda oleh penyedia identitas. Jalur relatif diperlakukan sebagai relatif terhadap direktori kerja proses.

Kompatibilitas dengan AWS SDKs

Berikut ini SDKs mendukung fitur dan pengaturan yang dijelaskan dalam topik ini. Setiap pengecualian sebagian dicatat. Setiap pengaturan properti JVM sistem didukung oleh AWS SDK for Java dan AWS SDK for Kotlin satu-satunya.

SDK	Di	Catatan atau informasi lebih lanjut
AWS CLI v2	Ya	
SDK untuk C++	Parsi	<code>credential_source</code> tidak didukung. <code>duration_seconds</code> tidak didukung. <code>mfa_serial</code> tidak didukung.
SDK untuk Go V2 (1.x)	Ya	
SDK untuk Go 1.x (V1)	Ya	Untuk menggunakan pengaturan config file bersama, Anda harus mengaktifkan pemuatan dari file konfigurasi; lihat Sesi .
SDK untuk Java 2.x	Parsi	<code>mfa_serial</code> tidak didukung. <code>duration_seconds</code> tidak didukung.
SDK untuk Java 1.x	Parsi	<code>credential_source</code> tidak didukung. <code>mfa_serial</code> tidak didukung. JVM properti sistem tidak didukung.
SDK untuk JavaScript 3.x	Ya	
SDK untuk JavaScript 2.x	Parsi	<code>credential_source</code> tidak didukung.
SDK untuk Kotlin	Ya	
SDK untuk .NET 3.x	Ya	
SDK untuk PHP 3.x	Ya	
SDK untuk Python (Boto3)	Ya	
SDK untuk Ruby 3.x	Ya	
SDK untuk Rust	Ya	
SDK untuk Swift	Ya	

SDK	Di Catatan atau informasi lebih lanjut
Alat untuk PowerShell	Ya

Penyedia kredensi kontainer

Penyedia kredensi kontainer mengambil kredensi untuk aplikasi kontainer pelanggan. Penyedia kredensi ini berguna untuk pelanggan Amazon Elastic Container Service (Amazon ECS) dan Amazon Elastic Kubernetes Service (Amazon EKS). SDKs mencoba memuat kredensi dari HTTP titik akhir yang ditentukan melalui permintaan GET.

Jika Anda menggunakan Amazon ECS, sebaiknya gunakan IAM Peran tugas untuk meningkatkan isolasi kredensi, otorisasi, dan auditabilitas. Saat dikonfigurasi, Amazon ECS menetapkan variabel `AWS_CONTAINER_CREDENTIALS_RELATIVE_URI` lingkungan yang digunakan SDKs dan alat untuk mendapatkan kredensi. Untuk mengonfigurasi Amazon ECS untuk fungsionalitas ini, lihat [IAM Peran tugas](#) di Panduan Pengembang Layanan Kontainer Elastis Amazon.

Jika Anda menggunakan Amazon EKS, kami sarankan Anda menggunakan Amazon EKS Pod Identity untuk meningkatkan isolasi kredensi, hak istimewa terkecil, auditabilitas, operasi independen, usabilitas ulang, dan skalabilitas. Baik Pod dan IAM peran Anda dikaitkan dengan akun layanan Kubernetes untuk mengelola kredensi aplikasi Anda. Untuk mempelajari lebih lanjut tentang Identitas EKS Pod [Amazon](#), lihat [Identitas EKS Pod Amazon](#) di Panduan EKS Pengguna Amazon. Saat dikonfigurasi, Amazon EKS menetapkan variabel `AWS_CONTAINER_CREDENTIALS_FULL_URI` dan `AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE` lingkungan yang digunakan alat SDKs dan alat untuk mendapatkan kredensi. Untuk informasi penyiapan, lihat [Menyiapkan Agen Identitas EKS Pod Amazon](#) di Panduan EKS Pengguna Amazon atau [Amazon EKS Pod Identity menyederhanakan IAM izin untuk aplikasi di EKS kluster Amazon di situs web](#) Blog. AWS

Konfigurasi fungsi ini dengan menggunakan yang berikut:

`AWS_CONTAINER_CREDENTIALS_FULL_URI`- variabel lingkungan

Menentukan HTTP URL endpoint penuh untuk digunakan ketika membuat permintaan SDK untuk kredensial. Ini termasuk skema dan tuan rumah.

Nilai default: Tidak ada.

Nilai yang valid: ValidURI.

Catatan: Pengaturan ini merupakan alternatif `AWS_CONTAINER_CREDENTIALS_RELATIVE_URI` dan hanya akan digunakan jika tidak `AWS_CONTAINER_CREDENTIALS_RELATIVE_URI` disetel.

Linux/macOS contoh pengaturan variabel lingkungan melalui baris perintah:

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

atau

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

AWS_CONTAINER_CREDENTIALS_RELATIVE_URI- variabel lingkungan

Menentukan HTTP URL endpoint relatif untuk digunakan ketika membuat permintaan SDK untuk kredensial. Nilai ditambahkan ke ECS nama host Amazon default dari `169.254.170.2`

Nilai default: Tidak ada.

Nilai yang valid: Relatif yang validURI.

Linux/macOS contoh pengaturan variabel lingkungan melalui baris perintah:

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

AWS_CONTAINER_AUTHORIZATION_TOKEN- variabel lingkungan

Menentukan token otorisasi dalam teks biasa. Jika variabel ini diatur, SDK akan mengatur header Otorisasi pada HTTP permintaan dengan nilai variabel lingkungan.

Nilai default: Tidak ada.

Nilai yang valid: String.

Catatan: Pengaturan ini merupakan alternatif `AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE` dan hanya akan digunakan jika tidak `AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE` disetel.

Linux/macOS contoh pengaturan variabel lingkungan melalui baris perintah:

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

```
export AWS_CONTAINER_AUTHORIZATION_TOKEN=Basic abcd
```

AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE- variabel lingkungan

Menentukan path file absolut ke file yang berisi token otorisasi dalam teks biasa.

Nilai default: Tidak ada.

Nilai yang valid: String.

Linux/macOS contoh pengaturan variabel lingkungan melalui baris perintah:

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

Kompatibilitas dengan AWS SDKs

Berikut ini SDKs mendukung fitur dan pengaturan yang dijelaskan dalam topik ini. Setiap pengecualian sebagian dicatat. Setiap pengaturan properti JVM sistem didukung oleh AWS SDK for Java dan AWS SDK for Kotlin satu-satunya.

SDK	Di	Catatan atau informasi lebih lanjut
AWS CLI v2	Ya	
SDK untuk C ++	Ya	
SDK untuk Go V2 (1.x)	Ya	
SDK untuk Go 1.x (V1)	Ya	
SDK untuk Java 2.x	Ya	AWS_CONTAINER_CREDENTIALS_FULL_URI dan juga digunakan untuk Lambda SnapStart untuk Java .
SDK untuk Java 1.x	Ya	AWS_CONTAINER_CREDENTIALS_FULL_URI dan juga digunakan untuk Lambda SnapStart untuk Java .
SDK untuk JavaScript 3.x	Ya	

SDK	Di	Catatan atau informasi lebih lanjut
SDK untuk JavaScript 2.x	Ya	
SDK untuk Kotlin	Ya	
SDK untuk .NET 3.x	Ya	
SDK untuk PHP 3.x	Ya	
SDK untuk Python (Boto3)	Ya	
SDK untuk Ruby 3.x	Ya	
SDK untuk Rust	Ya	
SDK untuk Swift	Ya	
Alat untuk PowerShell	Ya	

IAM Penyedia kredensi Pusat Identitas

Mekanisme otentikasi ini digunakan AWS IAM Identity Center untuk mendapatkan akses single sign-on (SSO) Layanan AWS untuk kode Anda.

Note

Dalam AWS SDK API dokumentasi, penyedia kredensi Pusat IAM Identitas disebut penyedia SSO kredensi.

Setelah mengaktifkan Pusat IAM Identitas, Anda menentukan profil untuk pengaturannya di AWS config file bersama Anda. Profil ini digunakan untuk terhubung ke portal akses Pusat IAM Identitas. Ketika pengguna berhasil mengautentikasi dengan Pusat IAM Identitas, portal mengembalikan kredensi jangka pendek untuk IAM peran yang terkait dengan pengguna tersebut. Untuk mempelajari cara SDK mendapatkan kredensi sementara dari konfigurasi dan menggunakannya untuk Layanan AWS permintaan, lihat [Memahami autentikasi Pusat Identitas IAM](#)

Ada dua cara untuk mengkonfigurasi Pusat IAM Identitas melalui config file:

- (Disarankan) konfigurasi penyedia SSO token — Durasi sesi yang diperpanjang. Termasuk dukungan untuk durasi sesi kustom.
- Konfigurasi lama yang tidak dapat disegarkan - Menggunakan sesi delapan jam yang tetap.

Di kedua konfigurasi, Anda harus masuk lagi saat sesi Anda kedaluwarsa.

Dua panduan berikut berisi informasi tambahan tentang Pusat IAM Identitas:

- [AWS IAM Identity Center Panduan Pengguna](#)
- [AWS IAM Identity Center API Referensi Portal](#)

Untuk menyelam lebih dalam tentang cara SDKs dan alat menggunakan dan menyegarkan kredensial menggunakan konfigurasi ini, lihat [Memahami autentikasi Pusat Identitas IAM](#)

Prasyarat

Anda harus mengaktifkan Pusat IAM Identitas terlebih dahulu. Untuk detail tentang mengaktifkan autentikasi Pusat IAM Identitas, lihat [Mengaktifkan AWS IAM Identity Center di Panduan Pengguna AWS IAM Identity Center](#)

Note

Atau, untuk prasyarat lengkap dan konfigurasi config file bersama yang diperlukan yang dirinci di halaman ini, lihat petunjuk yang dipandu untuk pengaturan. [IAM Autentikasi Pusat Identitas untuk alat SDK atau Anda](#)

SSO konfigurasi penyedia token

Saat Anda menggunakan konfigurasi penyedia SSO token, alat AWS SDK atau alat Anda secara otomatis menyegarkan sesi hingga periode sesi yang diperpanjang. Untuk informasi selengkapnya tentang durasi sesi dan durasi maksimum, lihat [Mengonfigurasi durasi sesi portal AWS akses dan aplikasi terintegrasi Pusat IAM Identitas](#) di Panduan AWS IAM Identity Center Pengguna.

`sso-session` Bagian config file digunakan untuk mengelompokkan variabel konfigurasi untuk memperoleh token SSO akses, yang kemudian dapat digunakan untuk memperoleh AWS kredensial. Untuk detail selengkapnya tentang bagian ini dalam sebuah config file, lihat [Format file konfigurasi](#).

Contoh config file bersama berikut mengonfigurasi alat SDK atau menggunakan dev profil untuk meminta kredensi Pusat IAM Identitas.

```
[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
```

Contoh sebelumnya menunjukkan bahwa Anda menentukan `sso-session` bagian dan mengaitkannya ke profil. Biasanya, `sso_account_id` dan `sso_role_name` harus diatur di `profile` bagian sehingga SDK dapat meminta AWS kredensial. `sso_region`, `sso_start_url`, dan `sso_registration_scopes` harus diatur dalam `sso-session` bagian.

`sso_account_id` dan `sso_role_name` tidak diperlukan untuk semua skenario konfigurasi SSO token. Jika aplikasi Anda hanya menggunakan otentikasi pembawa dukungan Layanan AWS itu, maka AWS kredensial tradisional tidak diperlukan. Otentikasi pembawa adalah skema HTTP otentikasi yang menggunakan token keamanan yang disebut token pembawa. Dalam skenario ini, `sso_account_id` dan `sso_role_name` tidak diperlukan. Lihat Layanan AWS panduan individual untuk menentukan apakah layanan mendukung otorisasi token pembawa.

Lingkup pendaftaran dikonfigurasi sebagai bagian dari `flessso-session`. Cakupan adalah mekanisme di OAuth 2.0 untuk membatasi akses aplikasi ke akun pengguna. Contoh sebelumnya ditetapkan `sso_registration_scopes` untuk menyediakan akses yang diperlukan untuk daftar akun dan peran.

Contoh berikut menunjukkan bagaimana Anda dapat menggunakan kembali `sso-session` konfigurasi yang sama di beberapa profil.

```
[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
```

Token otentikasi di-cache ke disk di bawah `~/.aws/sso/cache` direktori dengan nama file berdasarkan nama sesi.

Konfigurasi lama yang tidak dapat disegarkan

Penyegaran token otomatis tidak didukung menggunakan konfigurasi lama yang tidak dapat disegarkan. Kami merekomendasikan menggunakan [SSO konfigurasi penyedia token](#) sebagai gantinya.

Untuk menggunakan konfigurasi lama yang tidak dapat disegarkan, Anda harus menentukan pengaturan berikut di dalam profil Anda:

- `sso_start_url`
- `sso_region`
- `sso_account_id`
- `sso_role_name`

Anda menentukan portal pengguna untuk profil dengan `sso_start_url` dan `sso_region` pengaturan. Anda menentukan izin dengan `sso_role_name` pengaturan `sso_account_id` dan.

Contoh berikut menetapkan empat nilai yang diperlukan dalam config file.

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

Token otentikasi di-cache ke disk di bawah `~/.aws/sso/cache` direktori dengan nama file berdasarkan file. `sso_start_url`

IAMPengaturan penyedia kredensi Pusat Identitas

Konfigurasi fungsi ini dengan menggunakan yang berikut ini:

sso_start_url- Pengaturan AWS **config** file bersama

URL yang menunjuk ke penerbit Pusat IAM Identitas organisasi Anda URL atau portal URL akses. Untuk informasi selengkapnya, lihat [Menggunakan portal AWS akses](#) di Panduan AWS IAM Identity Center Pengguna.

Untuk menemukan nilai ini, buka [konsol Pusat IAM Identitas](#), lihat Dasbor, temukan portal AWS akses URL.

- Atau, dimulai dengan versi 2.22.0 dari AWS CLI, Anda dapat menggunakan nilai untuk AWS Penerbit. URL

sso_region- Pengaturan AWS **config** file bersama

Yang Wilayah AWS berisi host portal Pusat IAM Identitas Anda; yaitu, Wilayah yang Anda pilih sebelum mengaktifkan Pusat IAM Identitas. Ini independen dari AWS Wilayah default Anda, dan bisa berbeda.

Untuk daftar lengkap Wilayah AWS dan kodenya, lihat [Titik Akhir Regional](#) di Referensi Umum Amazon Web Untuk menemukan nilai ini, buka [konsol Pusat IAM Identitas](#), lihat Dasbor, dan temukan Wilayah.

sso_account_id- Pengaturan AWS **config** file bersama

ID numerik Akun AWS yang ditambahkan melalui AWS Organizations layanan untuk digunakan untuk otentikasi.

Untuk melihat daftar akun yang tersedia, buka [konsol Pusat IAM Identitas](#) dan buka Akun AWS Halaman. Anda juga dapat melihat daftar akun yang tersedia menggunakan [ListAccounts](#) API metode di API Referensi AWS IAM Identity Center Portal. Misalnya, Anda dapat memanggil AWS CLI metode [daftar-akun](#).

sso_role_name- Pengaturan AWS **config** file bersama

Nama set izin disediakan sebagai IAM peran yang menentukan izin yang dihasilkan pengguna. Peran harus ada dalam yang Akun AWS ditentukan oleh `sso_account_id`. Gunakan nama peran, bukan peran Amazon Resource Name (ARN).

Kumpulan izin memiliki IAM kebijakan dan kebijakan izin khusus yang dilampirkan padanya dan menentukan tingkat akses yang dimiliki pengguna ke yang ditetapkan Akun AWS.

Untuk melihat daftar set izin yang tersedia per Akun AWS, buka [konsol Pusat IAM Identitas](#) dan buka Akun AWS IAM. Pilih nama set izin yang benar yang tercantum dalam Akun AWS tabel. Anda juga dapat melihat daftar set izin yang tersedia menggunakan [ListAccountRoles](#) API metode di API Referensi AWS IAM Identity Center Portal. Misalnya, Anda dapat memanggil AWS CLI metode [list-account-roles](#).

sso_registration_scopes- Pengaturan AWS **config** file bersama

Daftar string cakupan valid yang dibatasi koma yang akan diotorisasi untuk `sso-session` Aplikasi dapat meminta satu atau lebih cakupan, dan token akses yang dikeluarkan untuk aplikasi terbatas pada cakupan yang diberikan. Cakupan minimum `sso:account:access` harus diberikan untuk mendapatkan token penyegaran kembali dari layanan Pusat IAM Identitas. Untuk daftar opsi cakupan akses yang tersedia, lihat [Cakupan akses](#) di Panduan AWS IAM Identity Center Pengguna.

Cakupan ini menentukan izin yang diminta untuk diotorisasi untuk OIDC klien terdaftar dan token akses yang diambil oleh klien. Cakupan mengotorisasi akses ke titik akhir resmi token pembawa IAM Identity Center.

Pengaturan ini tidak berlaku untuk konfigurasi lama yang tidak dapat disegarkan. Token yang dikeluarkan menggunakan konfigurasi lama terbatas pada ruang lingkup secara `sso:account:access` implisit.

Kompatibilitas dengan AWS SDKs

Berikut ini SDKs mendukung fitur dan pengaturan yang dijelaskan dalam topik ini. Setiap pengecualian sebagian dicatat. Setiap pengaturan properti JVM sistem didukung oleh AWS SDK for Java dan AWS SDK for Kotlin satu-satunya.

SDK	Di Catatan atau informasi lebih lanjut
AWS CLI v2	Ya
SDK untuk C++	Ya
SDK untuk Go V2 (1.x)	Ya
SDK untuk Go 1.x (V1)	Ya Untuk menggunakan pengaturan <code>config</code> file bersama, Anda harus mengaktifkan pemuatan dari file konfigurasi; lihat Sesi .

SDK	Di	Catatan atau informasi lebih lanjut
SDK untuk Java 2.x	Ya	Nilai konfigurasi juga didukung dalam <code>credentials</code> file.
SDK untuk Java 1.x	Tida	
SDK untuk JavaScript 3.x	Ya	
SDK untuk JavaScript 2.x	Ya	
SDK untuk Kotlin	Ya	
SDK untuk .NET 3.x	Ya	
SDK untuk PHP 3.x	Ya	
SDK untuk Python (Boto3)	Ya	
SDK untuk Ruby 3.x	Ya	
SDK untuk Rust	Parsi	Konfigurasi lama yang tidak dapat disegarkan saja.
SDK untuk Swift	Ya	
Alat untuk PowerShell	Ya	

IMDS penyedia kredensi

Instance Metadata Service (IMDS) menyediakan data tentang instans yang dapat Anda gunakan untuk mengonfigurasi atau mengelola instance yang sedang berjalan. Untuk informasi selengkapnya tentang data yang tersedia, lihat [Bekerja dengan metadata instans](#) di EC2 Panduan Pengguna Amazon. Amazon EC2 menyediakan endpoint lokal yang tersedia untuk instans yang dapat memberikan berbagai bit informasi ke instans. Jika instance memiliki peran yang dilampirkan, itu dapat memberikan serangkaian kredensial yang valid untuk peran itu. Endpoint SDKs dapat menggunakan endpoint tersebut untuk menyelesaikan kredensi sebagai bagian dari rantai penyedia [kredensi default](#) mereka. Instance Metadata Service Version 2 (IMDSv2), versi yang lebih aman IMDS yang menggunakan token sesi, digunakan secara default. Jika gagal karena kondisi yang tidak dapat dicoba ulang (kode HTTP kesalahan 403, 404, 405), IMDSv1 digunakan sebagai fallback.

Konfigurasi fungsi ini dengan menggunakan yang berikut ini:

AWS_EC2_METADATA_DISABLED- variabel lingkungan

Apakah akan mencoba menggunakan Amazon EC2 Instance Metadata Service (IMDS) untuk mendapatkan kredensi atau tidak.

Nilai default: `false`.

Nilai yang valid:

- **true**— Jangan gunakan IMDS untuk mendapatkan kredensi.
- **false**— Gunakan IMDS untuk mendapatkan kredensi.

ec2_metadata_v1_disabled- Pengaturan AWS **config** file bersama, **AWS_EC2_METADATA_V1_DISABLED**- variabel lingkungan, **aws.disableEc2MetadataV1**- properti JVM sistem: Hanya Java/Kotlin

Apakah akan menggunakan Instance Metadata Service Version 1 (IMDSv1) sebagai fallback jika gagal atau tidak. IMDSv2

Note

Baru SDKs tidak mendukung IMDSv1 dan, dengan demikian, tidak mendukung pengaturan ini. Untuk detailnya, lihat tabel [Kompatibilitas dengan AWS SDKs](#).

Nilai default: `false`.

Nilai yang valid:

- **true**— Jangan gunakan IMDSv1 sebagai fallback.
- **false**— Gunakan IMDSv1 sebagai fallback.

ec2_metadata_service_endpoint- Pengaturan AWS **config** file bersama, **AWS_EC2_METADATA_SERVICE_ENDPOINT**- variabel lingkungan, **aws.ec2MetadataServiceEndpoint**- properti JVM sistem: Hanya Java/Kotlin

Titik akhir dari. IMDS Nilai ini mengesampingkan lokasi default yang AWS SDKs dan alat akan mencari metadata EC2 instans Amazon.

Nilai default: Jika `ec2_metadata_service_endpoint_mode` sama `IPv4`, maka titik akhir default adalah. `http://169.254.169.254` Jika `ec2_metadata_service_endpoint_mode` sama `IPv6`, maka titik akhir default adalah. `http://[fd00:ec2::254]`

Nilai yang valid: ValidURI.

ec2_metadata_service_endpoint_mode- Pengaturan AWS **config** file bersama, **AWS_EC2_METADATA_SERVICE_ENDPOINT_MODE**- variabel lingkungan, **aws.ec2MetadataServiceEndpointMode**- properti JVM sistem: Hanya Java/Kotlin

Modus endpoint dari. IMDS

Nilai default:IPv4.

Nilai yang valid:IPv4,IPv6.

Note

Penyedia IMDS kredensi adalah bagian dari. [Memahami rantai penyedia kredensi](#) Namun, penyedia IMDS kredensi hanya diperiksa setelah beberapa penyedia lain yang ada di seri ini. Oleh karena itu, jika Anda ingin program Anda menggunakan kredensi penyedia ini, Anda harus menghapus penyedia kredensi valid lainnya dari konfigurasi Anda atau menggunakan profil yang berbeda. Atau, alih-alih mengandalkan rantai penyedia kredensi untuk secara otomatis menemukan penyedia mana yang mengembalikan kredensi yang valid, tentukan penggunaan penyedia kredensi dalam IMDS kode. Anda dapat menentukan sumber kredensi secara langsung saat Anda membuat klien layanan.

Keamanan untuk IMDS kredensial

Secara default, jika tidak AWS SDK dikonfigurasi dengan kredensi yang valid, SDK akan mencoba menggunakan Amazon EC2 Instance Metadata Service (IMDS) untuk mengambil kredensial untuk peran. AWS Perilaku ini dapat dinonaktifkan dengan menyetel variabel **AWS_EC2_METADATA_DISABLED** lingkungan ke `true`. Hal ini mencegah aktivitas jaringan yang tidak perlu dan meningkatkan keamanan pada jaringan yang tidak tepercaya di mana Layanan Metadata EC2 Instans Amazon dapat ditiru.

Note

AWS SDK klien yang dikonfigurasi dengan kredensi yang valid tidak akan pernah digunakan IMDS untuk mengambil kredensial, terlepas dari pengaturan ini.

Menonaktifkan penggunaan kredensial Amazon EC2 IMDS

Bagaimana Anda mengatur variabel lingkungan ini tergantung pada sistem operasi apa yang digunakan serta apakah Anda ingin perubahan itu persisten atau tidak.

Linux dan macOS

Pelanggan yang menggunakan Linux atau macOS dapat mengatur variabel lingkungan ini dengan perintah berikut:

```
$ export AWS_EC2_METADATA_DISABLED=true
```

Jika Anda ingin pengaturan ini persisten di beberapa sesi shell dan restart sistem, Anda dapat menambahkan perintah di atas ke file profil shell Anda, seperti `.bash_profile`, `.zsh_profile`, atau `.profile`.

Windows

Pelanggan yang menggunakan Windows dapat mengatur variabel lingkungan ini dengan perintah berikut:

```
$ set AWS_EC2_METADATA_DISABLED=true
```

Jika Anda ingin pengaturan ini persisten di beberapa sesi shell dan restart sistem dapat menggunakan perintah berikut sebagai gantinya:

```
$ setx AWS_EC2_METADATA_DISABLED=true
```

Note

setxPerintah tidak menerapkan nilai ke sesi shell saat ini, jadi Anda perlu memuat ulang atau membuka kembali shell agar perubahan diterapkan.

Kompatibilitas dengan AWS SDKs

Berikut ini SDKs mendukung fitur dan pengaturan yang dijelaskan dalam topik ini. Setiap pengecualian sebagian dicatat. Setiap pengaturan properti JVM sistem didukung oleh AWS SDK for Java dan AWS SDK for Kotlin satu-satunya.

SDK	Di	Catatan atau informasi lebih lanjut
AWS CLI v2	Ya	
SDK untuk C++	Ya	
SDK untuk Go V2 (1.x)	Ya	
SDK untuk Go 1.x (V1)	Ya	Untuk menggunakan pengaturan config file bersama, Anda harus mengaktifkan pemuatan dari file konfigurasi; lihat Sesi .
SDK untuk Java 2.x	Ya	
SDK untuk Java 1.x	Parsi	JVM properti sistem: Gunakan <code>com.amazonaws.sdk.disableEc2MetadataV1</code> sebagai pengganti <code>aws.disableEc2MetadataV1</code> ; <code>aws.ec2MetadataServiceEndpoint</code> dan <code>aws.ec2MetadataServiceEndpointMode</code> tidak didukung.
SDK untuk JavaScript 3.x	Ya	
SDK untuk JavaScript 2.x	Ya	
SDK untuk Kotlin	Ya	Tidak menggunakan IMDSv1 fallback.
SDK untuk .NET 3.x	Ya	
SDK untuk PHP 3.x	Ya	
SDK untuk Python (Boto3)	Ya	
SDK untuk Ruby 3.x	Ya	
SDK untuk Rust	Ya	Tidak menggunakan IMDSv1 fallback.
SDK untuk Swift	Ya	
Alat untuk PowerShell	Ya	Anda dapat menonaktifkan IMDSv1 fallback secara eksplisit dalam kode menggunakan <code>[Amazon.Util].EC2In</code>

SDK	Di Catatan atau informasi lebih lanjut
	<code>stanceMetadata]::EC2MetadataV1Disabled = \$true</code>

Penyedia kredensi proses

SDK menyediakan cara untuk memperluas rantai penyedia kredensial untuk kasus penggunaan khusus. Penyedia ini dapat digunakan untuk menyediakan implementasi kustom, seperti mengambil kredensial dari penyimpanan kredensial lokal atau mengintegrasikan dengan penyedia identifikasi lokal Anda.

Misalnya, IAM Roles Anywhere digunakan `credential_process` untuk mendapatkan kredensi sementara atas nama aplikasi Anda. `credential_process` Untuk mengonfigurasi penggunaan ini, lihat [Peran IAM Di Mana Saja](#).

Note

Berikut ini menjelaskan metode sumber kredensial dari proses eksternal dan dapat digunakan jika Anda menjalankan perangkat lunak di luar AWS. Jika Anda membangun sebuah AWS menghitung sumber daya, menggunakan penyedia kredensi lainnya. Jika menggunakan opsi ini, Anda harus memastikan bahwa file konfigurasi dikunci sebanyak mungkin menggunakan praktik terbaik keamanan untuk sistem operasi Anda. Konfirmasikan bahwa alat kredensi kustom Anda tidak menulis informasi rahasia apa pun `StdErr`, karena dan SDKs AWS CLI dapat menangkap dan mencatat informasi tersebut, berpotensi mengeksposnya kepada pengguna yang tidak sah.

Konfigurasi fungsi ini dengan menggunakan yang berikut ini:

`credential_process`- dibagikan AWS **`config`** pengaturan file

Menentukan perintah eksternal yang SDK atau alat berjalan atas nama Anda untuk menghasilkan atau mengambil kredensi otentikasi untuk digunakan. Pengaturan menentukan nama program/perintah yang akan dipanggil. SDK Ketika SDK memanggil proses, ia menunggu proses untuk menulis JSON data. `stdout` Penyedia kustom harus mengembalikan informasi dalam format tertentu. Informasi tersebut berisi kredensial yang dapat digunakan alat SDK atau untuk mengautentikasi Anda.

Note

Penyedia kredensi proses adalah bagian dari [Memahami rantai penyedia kredensi](#). Namun, penyedia kredensi proses hanya diperiksa setelah beberapa penyedia lain yang ada di seri ini. Oleh karena itu, jika Anda ingin program Anda menggunakan kredensi penyedia ini, Anda harus menghapus penyedia kredensi valid lainnya dari konfigurasi Anda atau menggunakan profil yang berbeda. Atau, alih-alih mengandalkan rantai penyedia kredensi untuk secara otomatis menemukan penyedia mana yang mengembalikan kredensi yang valid, tentukan penggunaan penyedia kredensi proses dalam kode. Anda dapat menentukan sumber kredensi secara langsung saat Anda membuat klien layanan.

Menentukan jalur ke program kredensyal

Nilai pengaturan adalah string yang berisi jalur ke program yang dijalankan SDK atau alat pengembangan atas nama Anda:

- Jalur dan nama file hanya dapat terdiri dari karakter-karakter ini: A-Z, a-z, 0-9, tanda hubung (-), garis bawah (_), periode (.), garis miring maju (/), garis miring terbalik (\), dan spasi.
- Jika jalur atau nama file berisi spasi, kelilingi jalur lengkap dan nama file dengan tanda kutip ganda ("").
- Jika nama parameter atau nilai parameter berisi spasi, kelilingi elemen tersebut dengan tanda kutip ganda (""). Kelilingi hanya nama atau nilainya, bukan pasangannya.
- Jangan sertakan variabel lingkungan apa pun dalam string. Misalnya, jangan sertakan \$HOME atau %USERPROFILE%.
- Jangan tentukan folder beranda sebagai ~. * Anda harus menentukan jalur lengkap atau nama file dasar. Jika ada nama file dasar, sistem mencoba untuk menemukan program dalam folder yang ditentukan oleh variabel PATH lingkungan. Jalur bervariasi tergantung pada sistem operasi:

Contoh berikut menunjukkan pengaturan `credential_process` dalam file bersama `config` di Linux/ macOS.

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

Contoh berikut menunjukkan pengaturan `credential_process` dalam file bersama `config` pada Windows.

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

- Dapat ditentukan dalam profil khusus:

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

Output yang valid dari program kredensial

SDK menjalankan perintah seperti yang ditentukan dalam profil dan kemudian membaca data dari aliran output standar. Perintah yang Anda tentukan, apakah skrip atau program biner, harus menghasilkan JSON output STDOUT yang cocok dengan sintaks berikut.

```
{
  "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

Pada tulisan ini, `Version` kuncinya harus diatur ke 1. Ini mungkin meningkat seiring waktu seiring berkembangnya struktur.

`Expiration` kuncinya adalah stempel RFC3339 waktu yang diformat. Jika `Expiration` kunci tidak ada dalam output alat, SDK asumsi bahwa kredensialnya adalah kredensial jangka panjang yang tidak disegarkan. Jika tidak, kredensialnya dianggap sebagai kredensial sementara, dan secara otomatis disegarkan dengan menjalankan kembali perintah sebelum kredensialnya kedaluwarsa.

`credential_process`

Note

SDK itu tidak menyimpan kredensial proses eksternal seperti yang dilakukannya kredensial-peran asumsi. Jika caching diperlukan, Anda harus menerapkannya dalam proses eksternal.

Proses eksternal dapat mengembalikan kode pengembalian bukan nol untuk menunjukkan bahwa kesalahan terjadi saat mengambil kredensial.

Kompatibilitas dengan AWS SDKs

Berikut ini SDKs mendukung fitur dan pengaturan yang dijelaskan dalam topik ini. Setiap pengecualian sebagian dicatat. Pengaturan properti JVM sistem apa pun didukung oleh AWS SDK for Java dan AWS SDK for Kotlin hanya.

SDK	Di	Catatan atau informasi lebih lanjut
AWS CLI v2	Ya	
SDK untuk C++	Ya	
SDK untuk Go V2 (1.x)	Ya	
SDK untuk Go 1.x (V1)	Ya	Untuk menggunakan pengaturan config file bersama, Anda harus mengaktifkan pemuatan dari file konfigurasi; lihat Sesi .
SDK untuk Java 2.x	Ya	
SDK untuk Java 1.x	Ya	
SDK untuk JavaScript 3.x	Ya	
SDK untuk JavaScript 2.x	Ya	
SDK untuk Kotlin	Ya	
SDK untuk .NET 3.x	Ya	
SDK untuk PHP 3.x	Ya	
SDK untuk Python (Boto3)	Ya	

SDK	Di	Catatan atau informasi lebih lanjut
SDK untuk Ruby 3.x	Ya	
SDK untuk Rust	Ya	
SDK untuk Swift	Ya	
Alat untuk PowerShell	Ya	

AWS SDKs dan fitur standar Alat

Banyak fitur telah distandarisasi ke default yang konsisten dan bekerja dengan cara yang sama di banyak fitur. SDKs Konsistensi ini meningkatkan produktivitas dan kejelasan saat pengkodean di beberapa SDKs. Semua pengaturan dapat diganti dalam kode, lihat spesifik SDK API Anda untuk detailnya.

Important

Tidak semua SDKs mendukung semua fitur, atau bahkan semua aspek dalam suatu fitur.

Topik

- [Titik akhir berbasis akun](#)
- [ID Aplikasi](#)
- [EC2 Metadata contoh Amazon](#)
- [Titik akses Amazon S3](#)
- [Titik Akses Multi-Wilayah Amazon S3](#)
- [Wilayah AWS](#)
- [AWS STS Titik akhir regional](#)
- [Tumpukan ganda dan titik akhir FIPS](#)
- [Penemuan titik akhir](#)
- [Pengaturan konfigurasi umum](#)
- [IMDS klien](#)

- [Coba lagi perilaku](#)
- [Minta kompresi](#)
- [Titik akhir khusus layanan](#)
- [Default konfigurasi cerdas](#)

Titik akhir berbasis akun

Titik akhir berbasis akun membantu memastikan kinerja dan skalabilitas tinggi dengan menggunakan Akun AWS ID Anda untuk merampingkan perutean Layanan AWS permintaan layanan yang mendukung fitur ini. Saat Anda menggunakan penyedia kredensi, dan layanan yang mendukung titik akhir berbasis akun, SDK maka secara otomatis akan membangun dan menggunakan titik akhir berbasis akun daripada titik akhir regional. AWS SDK Titik akhir berbasis akun berbentuk `https://<account-id>.ddb.<region>.amazonaws.com`, di mana `<account-id>` diganti dengan Akun AWS ID Anda dan `<region>` diganti dengan Anda Wilayah AWS

Secara default, ID akun dikumpulkan saat permintaan diproses dan digunakan untuk membuat titik akhir. Resolusi kredensial juga terjadi ketika permintaan diproses dan mungkin mengubah metode resolusi titik akhir. Bergantung pada penyedia kredensi apa yang Anda gunakan, ID akun mungkin bersumber secara berbeda.

Konfigurasi fungsi ini dengan menggunakan yang berikut:

aws_account_id- Pengaturan AWS **config** file bersama, **AWS_ACCOUNT_ID**- variabel lingkungan, **aws.accountId**- properti JVM sistem: Hanya Java/Kotlin

Akun AWS ID. Digunakan untuk routing endpoint berbasis akun. Akun AWS ID memiliki format seperti 111122223333.

Perutean endpoint berbasis akun memberikan kinerja permintaan yang lebih baik untuk beberapa layanan.

account_id_endpoint_mode- Pengaturan AWS **config** file bersama, **AWS_ACCOUNT_ID_ENDPOINT_MODE**- variabel lingkungan, **aws.accountIdEndpointMode**- properti JVM sistem: Hanya Java/Kotlin

Pengaturan ini digunakan untuk mematikan perutean endpoint berbasis akun jika perlu, dan melewati aturan berbasis akun.

Nilai default: `preferred`

Nilai yang valid:

- **preferred**— Titik akhir harus menyertakan ID akun jika tersedia.
- **disabled**— Titik akhir yang diselesaikan tidak menyertakan ID akun.
- **required**— Titik akhir harus menyertakan ID akun. Jika ID akun tidak tersedia, SDK kesalahan akan muncul.

Kompatibilitas dengan AWS SDKs

Berikut ini SDKs mendukung fitur dan pengaturan yang dijelaskan dalam topik ini. Setiap pengecualian sebagian dicatat. Setiap pengaturan properti JVM sistem didukung oleh AWS SDK for Java dan AWS SDK for Kotlin satu-satunya.

SDK	Didukung	Dirilis dalam SDK versi	Catatan atau informasi lebih lanjut
AWS CLI v2	Tidak		
SDK untuk C++	Tidak		
SDK untuk Go V2 (1.x)	Ya	v1.35.0	
SDK untuk Go 1.x (V1)	Tidak		
SDK untuk Java 2.x	Ya	v2.28.4	
SDK untuk Java 1.x	Ya	v1.12.771	
SDK untuk JavaScript 3.x	Ya	v3.656.0	
SDK untuk JavaScript 2.x	Tidak		
SDK untuk Kotlin	Ya	v1.3.37	

SDK	Didukung	Dirilis dalam SDK versi	Catatan atau informasi lebih lanjut
SDK untuk .NET 3.x	Tidak		
SDK untuk PHP 3.x	Ya	v3.318.0	
SDK untuk Python (Boto3)	Tidak		
SDK untuk Ruby 3.x	Ya	v1.123.0	
SDK untuk Rust	Tidak		
SDK untuk Swift	Tidak		
Alat untuk PowerShell	Tidak		

ID Aplikasi

Satu Akun AWS dapat digunakan oleh beberapa aplikasi pelanggan untuk melakukan panggilan ke Layanan AWS. ID Aplikasi menyediakan cara bagi pelanggan untuk mengidentifikasi aplikasi sumber mana yang melakukan serangkaian panggilan menggunakan Akun AWS. AWS SDKs dan layanan tidak menggunakan atau menafsirkan nilai ini selain untuk memunculkannya kembali dalam komunikasi pelanggan. Misalnya, nilai ini dapat dimasukkan dalam email operasional atau di AWS Health Dashboard untuk secara unik mengidentifikasi aplikasi mana yang terkait dengan notifikasi.

Konfigurasi fungsi ini dengan menggunakan yang berikut:

sdk_ua_app_id- dibagikan AWS **config** pengaturan file, **AWS_SDK_UA_APP_ID**- variabel lingkungan, **aws.userAgentAppId**- properti JVM sistem: Hanya Java/Kotlin

Pengaturan ini adalah string unik yang Anda tetapkan ke aplikasi Anda untuk mengidentifikasi aplikasi mana dalam aplikasi tertentu Akun AWS membuat panggilan ke AWS.

Nilai default: None

Nilai yang valid: String dengan panjang maksimum 50. Huruf, angka, dan karakter khusus berikut diperbolehkan: !\$,%,&,*,+,-,.,,^,_,`|,~.

Contoh pengaturan nilai ini dalam config file:

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

Linux/macOS contoh pengaturan variabel lingkungan melalui baris perintah:

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

Contoh Windows pengaturan variabel lingkungan melalui baris perintah:

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

Jika Anda menyertakan simbol yang memiliki arti khusus pada shell yang digunakan, lepaskan nilainya sebagaimana mestinya.

Kompatibilitas dengan AWS SDKs

Berikut ini SDKs mendukung fitur dan pengaturan yang dijelaskan dalam topik ini. Setiap pengecualian sebagian dicatat. Pengaturan properti JVM sistem apa pun didukung oleh AWS SDK for Java dan AWS SDK for Kotlin hanya.

SDK	Di Catatan atau informasi lebih lanjut
AWS CLI v2	Ya
SDK untuk C++	Ya configfile bersama tidak didukung.
SDK untuk Go V2 (1.x)	Ya
SDK untuk Go 1.x (V1)	Tida
SDK untuk Java 2.x	Parsi Pengaturan config file bersama tidak didukung; variabel lingkungan tidak didukung.

SDK	Di	Catatan atau informasi lebih lanjut
SDK untuk Java 1.x	Tida	
SDK untuk JavaScript 3.x	Ya	
SDK untuk JavaScript 2.x	Tida	
SDK untuk Kotlin	Ya	
SDK untuk .NET 3.x	Ya	Variabel lingkungan tidak didukung.
SDK untuk PHP 3.x	Ya	
SDK untuk Python (Boto3)	Ya	
SDK untuk Ruby 3.x	Ya	
SDK untuk Rust	Ya	
SDK untuk Swift	Tida	
Alat untuk PowerShell	Tida	

EC2 Metadata contoh Amazon

Amazon EC2 menyediakan layanan pada instans yang disebut Layanan Metadata Instans (IMDS). Untuk mempelajari selengkapnya tentang layanan ini, lihat [Bekerja dengan metadata instans](#) di EC2 Panduan Pengguna Amazon. Saat mencoba mengambil kredensial pada EC2 instans Amazon yang telah dikonfigurasi dengan IAM peran, koneksi ke layanan metadata instance dapat disesuaikan.

Konfigurasi fungsi ini dengan menggunakan yang berikut ini:

metadata_service_num_attempts- dibagikan AWS **config** pengaturan file,

AWS_METADATA_SERVICE_NUM_ATTEMPTS- variabel lingkungan

Pengaturan ini menentukan jumlah total upaya yang harus dilakukan sebelum menyerah ketika mencoba untuk mengambil data dari layanan metadata instance.

Nilai default: 1

Nilai yang valid: Angka lebih besar dari atau sama dengan 1.

metadata_service_timeout- dibagikan AWS **config** pengaturan file,
AWS_METADATA_SERVICE_TIMEOUT- variabel lingkungan

Menentukan jumlah detik sebelum waktu habis ketika mencoba untuk mengambil data dari layanan metadata contoh.

Nilai default: 1

Nilai yang valid: Angka lebih besar dari atau sama dengan 1.

Contoh pengaturan nilai-nilai ini dalam config file:

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

Linux/macOS contoh pengaturan variabel lingkungan melalui baris perintah:

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

Contoh Windows pengaturan variabel lingkungan melalui baris perintah:

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

Kompatibilitas dengan AWS SDKs

Berikut ini SDKs mendukung fitur dan pengaturan yang dijelaskan dalam topik ini. Setiap pengecualian sebagian dicatat. Pengaturan properti JVM sistem apa pun didukung oleh AWS SDK for Java dan AWS SDK for Kotlin hanya.

SDK	Di Catatan atau informasi lebih lanjut
AWS CLI v2	Ya
SDK untuk C ++	Tida
SDK untuk Go V2 (1.x)	Tida

SDK	Di	Catatan atau informasi lebih lanjut
SDK untuk Go 1.x (V1)	Tida	
SDK untuk Java 2.x	Tida	
SDK untuk Java 1.x	Parsi	Hanya AWS_METADATA_SERVICE_TIMEOUT didukung.
SDK untuk JavaScript 3.x	Tida	
SDK untuk JavaScript 2.x	Tida	
SDK untuk Kotlin	Tida	
SDK untuk .NET 3.x	Tida	
SDK untuk PHP 3.x	Ya	
SDK untuk Python (Boto3)	Ya	
SDK untuk Ruby 3.x	Tida	
SDK untuk Rust	Tida	
SDK untuk Swift	Tida	
Alat untuk PowerShell	Tida	

Titik akses Amazon S3

Layanan Amazon S3 menyediakan titik akses sebagai cara alternatif untuk berinteraksi dengan bucket Amazon S3. Titik akses memiliki kebijakan dan konfigurasi unik yang dapat diterapkan padanya, bukan langsung ke bucket. Dengan AWS SDKs, Anda dapat menggunakan titik akses Amazon Resource Names (ARNs) di bidang bucket untuk API operasi alih-alih menentukan nama bucket secara eksplisit. Mereka digunakan untuk operasi tertentu seperti menggunakan titik akses ARN dengan [GetObject](#) untuk mengambil objek dari ember, atau menggunakan titik akses ARN dengan [PutObject](#) untuk menambahkan objek ke ember.

Untuk mempelajari selengkapnya tentang jalur akses Amazon S3 dan ARNs, lihat [Menggunakan titik akses](#) di Panduan Pengguna Amazon S3.

Konfigurasi fungsi ini dengan menggunakan yang berikut ini:

s3_use_arn_region- dibagikan AWS **config** pengaturan file, **AWS_S3_USE_ARN_REGION**- variabel lingkungan, **aws.s3UseArnRegion**- properti JVM sistem: Hanya Java/Kotlin, Untuk mengonfigurasi nilai secara langsung dalam kode, konsultasikan spesifik Anda SDK secara langsung.

Pengaturan ini mengontrol apakah SDK menggunakan titik akses ARN Wilayah AWS untuk membangun titik akhir Regional untuk permintaan tersebut. Yang SDK memvalidasi bahwa ARN Wilayah AWS dilayani oleh yang sama AWS partisi sebagai klien yang dikonfigurasi Wilayah AWS untuk mencegah panggilan lintas partisi yang kemungkinan besar akan gagal. Jika kalikan ditentukan, pengaturan yang dikonfigurasi kode diutamakan, diikuti oleh pengaturan variabel lingkungan.

Nilai default: `false`

Nilai yang valid:

- **true**— SDK Kegunaannya ARN Wilayah AWS saat membangun titik akhir alih-alih dikonfigurasi klien Wilayah AWS. Pengecualian: Jika klien dikonfigurasi Wilayah AWS adalah FIPS Wilayah AWS, maka itu harus cocok ARN dengan Wilayah AWS. Jika tidak, kesalahan akan terjadi.
- **false**— SDK Menggunakan klien yang dikonfigurasi Wilayah AWS saat membangun titik akhir.

Kompatibilitas dengan AWS SDKs

Berikut ini SDKs mendukung fitur dan pengaturan yang dijelaskan dalam topik ini. Setiap pengecualian sebagian dicatat. Pengaturan properti JVM sistem apa pun didukung oleh AWS SDK for Java dan AWS SDK for Kotlin hanya.

SDK	Di Catatan atau informasi lebih lanjut
AWS CLI v2	Ya
SDK untuk C++	Ya
SDK untuk Go V2 (1.x)	Ya

SDK	Di	Catatan atau informasi lebih lanjut
SDK untuk Go 1.x (V1)	Ya	Untuk menggunakan pengaturan config file bersama, Anda harus mengaktifkan pemuatan dari file konfigurasi; lihat Sesi .
SDK untuk Java 2.x	Ya	
SDK untuk Java 1.x	Ya	JVM properti sistem tidak didukung.
SDK untuk JavaScript 3.x	Ya	
SDK untuk JavaScript 2.x	Ya	
SDK untuk Kotlin	Ya	
SDK untuk .NET 3.x	Ya	Tidak mengikuti prioritas standar; nilai config file bersama lebih diutamakan daripada variabel lingkungan.
SDK untuk PHP 3.x	Ya	
SDK untuk Python (Boto3)	Ya	
SDK untuk Ruby 3.x	Ya	
SDK untuk Rust	Tida	
SDK untuk Swift	Tida	
Alat untuk PowerShell	Ya	Tidak mengikuti prioritas standar; nilai config file bersama lebih diutamakan daripada variabel lingkungan.

Titik Akses Multi-Wilayah Amazon S3

Titik Akses Multi-Wilayah Amazon S3 menyediakan titik akhir global yang dapat digunakan aplikasi untuk memenuhi permintaan dari bucket Amazon S3 yang terletak di beberapa Wilayah AWS. Anda dapat menggunakan Titik Akses Multi-Wilayah untuk membangun aplikasi Multi-wilayah dengan arsitektur yang sama yang digunakan di satu Wilayah, dan kemudian menjalankan aplikasi tersebut di mana saja di dunia.

Untuk mempelajari lebih lanjut tentang Titik Akses Multi-Wilayah, lihat [Titik Akses Multi-Wilayah di Amazon S3 di Panduan Pengguna Amazon S3](#).

Untuk mempelajari lebih lanjut tentang Nama Sumber Daya Amazon (ARNs) Titik Akses Multi-Wilayah, lihat [Membuat permintaan menggunakan Titik Akses Multi-Wilayah](#) di Panduan Pengguna Amazon S3.

Untuk mempelajari selengkapnya tentang cara membuat Titik Akses Multi-Wilayah, lihat [Mengelola Titik Akses Multi-Wilayah](#) di Panduan Pengguna Amazon S3.

Algoritma Sigv4a adalah implementasi penandatanganan yang digunakan untuk menandatangani permintaan Wilayah global. Algoritma ini diperoleh dengan SDK melalui ketergantungan pada [AWS Pustaka Runtime \(CRT\) umum](#)

Konfigurasi fungsi ini dengan menggunakan yang berikut ini:

s3_disable_multiregion_access_points- dibagikan AWS **config** pengaturan file, **AWS_S3_DISABLE_MULTIREGION_ACCESS_POINTS**- variabel lingkungan, **aws.s3DisableMultiRegionAccessPoints**- properti JVM sistem: Hanya Java/Kotlin, Untuk mengonfigurasi nilai secara langsung dalam kode, konsultasikan spesifik Anda SDK secara langsung.

Setelah ini mengontrol apakah permintaan Lintas wilayah SDK berpotensi dilakukan. Jika kalikan ditentukan, pengaturan yang dikonfigurasi kode diutamakan, diikuti oleh pengaturan variabel lingkungan.

Nilai default: `false`

Nilai yang valid:

- **true**— Menghentikan penggunaan permintaan Lintas wilayah.
- **false**— Mengaktifkan permintaan Lintas wilayah menggunakan Titik Akses Multi-Wilayah.

Kompatibilitas dengan AWS SDKs

Berikut ini SDKs mendukung fitur dan pengaturan yang dijelaskan dalam topik ini. Setiap pengecualian sebagian dicatat. Pengaturan properti JVM sistem apa pun didukung oleh AWS SDK for Java dan AWS SDK for Kotlin hanya.

SDK	Di	Catatan atau informasi lebih lanjut
AWS CLI v2	Ya	
SDK untuk C++	Ya	
SDK untuk Go V2 (1.x)	Ya	
SDK untuk Go 1.x (V1)	Tida	
SDK untuk Java 2.x	Ya	
SDK untuk Java 1.x	Tida	
SDK untuk JavaScript 3.x	Ya	
SDK untuk JavaScript 2.x	Tida	
SDK untuk Kotlin	Ya	
SDK untuk .NET 3.x	Ya	
SDK untuk PHP 3.x	Ya	
SDK untuk Python (Boto3)	Ya	
SDK untuk Ruby 3.x	Ya	
SDK untuk Rust	Ya	
SDK untuk Swift	Tida	
Alat untuk PowerShell	Ya	

Wilayah AWS

Wilayah AWS adalah konsep penting untuk dipahami ketika bekerja dengan Layanan AWS.

Dengan Wilayah AWS, Anda dapat mengakses Layanan AWS yang secara fisik berada di wilayah geografis tertentu. Ini dapat berguna untuk menjaga data dan aplikasi Anda berjalan dekat dengan tempat Anda dan pengguna Anda akan mengaksesnya. Wilayah memberikan toleransi kesalahan,

stabilitas, serta ketahanan, dan juga dapat mengurangi latensi. Dengan Wilayah, Anda dapat membuat sumber daya redundan yang tetap tersedia dan tidak terpengaruh oleh pemadaman Regional.

Sebagian besar Layanan AWS permintaan dikaitkan dengan wilayah geografis tertentu. Sumber daya yang Anda buat di satu Wilayah tidak ada di Wilayah lain kecuali Anda secara eksplisit menggunakan fitur replikasi yang ditawarkan oleh Layanan AWS. Misalnya, Amazon S3 dan Amazon EC2 mendukung replikasi lintas wilayah. Beberapa layanan, seperti IAM, tidak memiliki sumber daya Regional.

Referensi Umum AWS Berisi informasi tentang hal berikut:

- Untuk memahami hubungan antara Wilayah dan titik akhir, dan untuk melihat daftar titik akhir Regional yang ada, lihat titik akhir [AWS layanan](#).
- Untuk melihat daftar saat ini dari semua Wilayah dan titik akhir yang didukung untuk masing-masing Layanan AWS, lihat [Titik akhir dan kuota layanan](#).

Membuat klien layanan

Untuk mengakses secara terprogram Layanan AWS, SDKs gunakan kelas/objek klien untuk masing-masing Layanan AWS. Jika aplikasi Anda perlu mengakses Amazon EC2, misalnya, aplikasi Anda akan membuat objek EC2 klien Amazon untuk berinteraksi dengan layanan itu.

Jika tidak ada Region yang secara eksplisit ditentukan untuk klien dalam kode itu sendiri, klien default menggunakan Region yang diatur melalui pengaturan berikut. `region` Namun, Region aktif untuk klien dapat secara eksplisit diatur untuk setiap objek klien individu. Mengatur Wilayah dengan cara ini lebih diutamakan daripada pengaturan global apa pun untuk klien layanan tertentu. Wilayah alternatif ditentukan selama instantiasi klien tersebut, khusus untuk Anda SDK (periksa SDK Panduan spesifik Anda atau basis kode Anda SDK).

Konfigurasi fungsi ini dengan menggunakan yang berikut:

region- Pengaturan AWS **config** file bersama, **AWS_REGION**- variabel lingkungan, **aws.region**- properti JVM sistem: Hanya Java/Kotlin

Menentukan default Wilayah AWS untuk digunakan untuk AWS permintaan. Wilayah ini digunakan untuk permintaan SDK layanan yang tidak disediakan dengan Wilayah tertentu untuk digunakan.

Nilai default: Tidak ada. Anda harus menentukan nilai ini secara eksplisit.

Nilai yang valid:

- Setiap kode Wilayah yang tersedia untuk layanan yang dipilih, seperti yang tercantum dalam [titik akhir AWS layanan](#) dalam Referensi AWS Umum. Misalnya, nilai `us-east-1` menetapkan titik akhir ke Timur Wilayah AWS AS (Virginia N.).
- `aws-global` menentukan titik akhir global untuk layanan yang mendukung titik akhir global terpisah selain titik akhir Regional, seperti AWS Security Token Service (AWS STS) dan Amazon Simple Storage Service (Amazon S3).

Contoh pengaturan nilai ini dalam config file:

```
[default]
region = us-west-2
```

Linux/macOS contoh pengaturan variabel lingkungan melalui baris perintah:

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

Contoh Windows pengaturan variabel lingkungan melalui baris perintah:

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

Sebagian besar SDKs memiliki objek “konfigurasi” yang tersedia untuk mengatur Wilayah default dari dalam kode aplikasi. Untuk detailnya, lihat panduan AWS SDK pengembang khusus Anda.

Kompatibilitas dengan AWS SDKs

Berikut ini SDKs mendukung fitur dan pengaturan yang dijelaskan dalam topik ini. Setiap pengecualian sebagian dicatat. Setiap pengaturan properti JVM sistem didukung oleh AWS SDK for Java dan AWS SDK for Kotlin satu-satunya.

SDK	Di Catatan atau informasi lebih lanjut
AWS CLI v2	Ya AWS CLI v2 menggunakan nilai apa pun <code>AWS_REGION</code> sebelum nilai apa pun di <code>AWS_DEFAULT_REGION</code> (kedua variabel diperiksa).
AWS CLI v1	Ya AWS CLI v1 menggunakan variabel lingkungan bernama <code>AWS_DEFAULT_REGION</code> untuk tujuan ini.

SDK	Di	Catatan atau informasi lebih lanjut
SDK untuk C++	Ya	
SDK untuk Go V2 (1.x)	Ya	
SDK untuk Go 1.x (V1)	Ya	Untuk menggunakan pengaturan config file bersama, Anda harus mengaktifkan pemuatan dari file konfigurasi; lihat Sesi .
SDK untuk Java 2.x	Ya	
SDK untuk Java 1.x	Ya	
SDK untuk JavaScript 3.x	Ya	
SDK untuk JavaScript 2.x	Ya	
SDK untuk Kotlin	Ya	
SDK untuk .NET 3.x	Ya	
SDK untuk PHP 3.x	Ya	
SDK untuk Python (Boto3)	Ya	Ini SDK menggunakan variabel lingkungan bernama <code>AWS_DEFAULT_REGION</code> untuk tujuan ini.
SDK untuk Ruby 3.x	Ya	
SDK untuk Rust	Ya	
SDK untuk Swift	Ya	
Alat untuk PowerShell	Ya	

AWS STS Titik akhir regional

AWS Security Token Service (AWS STS) tersedia baik sebagai layanan global maupun Regional. Beberapa AWS SDKs dan CLIs menggunakan endpoint layanan global (`https://sts.amazonaws.com`) secara default, sementara beberapa menggunakan endpoint layanan Regional (`https://sts.{region_identifier}.{partition_domain}`). Permintaan global

dipetakan ke Wilayah AS Timur (Virginia N.). Untuk informasi selengkapnya tentang AWS STS titik akhir, lihat [Titik Akhir dalam Referensi](#). AWS Security Token Service API Atau, belajar [Mengelola AWS STS Wilayah AWS dalam](#) Panduan AWS Identity and Access Management Pengguna.


Ini adalah praktik AWS terbaik untuk menggunakan titik akhir Regional bila memungkinkan dan untuk mengkonfigurasi Anda [Wilayah AWS](#). Pelanggan di [partisi](#) selain komersial harus menggunakan titik akhir Regional. Tidak semua SDKs dan alat mendukung pengaturan ini, tetapi semua memiliki perilaku yang ditentukan di sekitar titik akhir global dan Regional. Lihat bagian berikut untuk informasi selengkapnya.

Untuk SDKs dan alat yang mendukung pengaturan ini, pelanggan dapat mengonfigurasi fungsionalitas dengan menggunakan yang berikut ini:

sts_regional_endpoints- Pengaturan AWS **config** file bersama,
AWS_STS_REGIONAL_ENDPOINTS- variabel lingkungan

Pengaturan ini menentukan bagaimana SDK atau alat menentukan Layanan AWS titik akhir yang digunakannya untuk berbicara dengan AWS Security Token Service (AWS STS).

Nilai default: `legacy`

 Note

Semua versi SDK utama baru yang dirilis setelah Juli 2022 akan default ke `regional`. Versi SDK utama baru mungkin menghapus pengaturan ini dan menggunakan `regional` perilaku. Untuk mengurangi dampak masa depan terkait perubahan ini, kami sarankan Anda mulai menggunakan `regional` aplikasi Anda jika memungkinkan.

Nilai yang valid: (Nilai yang disarankan: `regional`)

- **legacy**— Menggunakan AWS STS titik akhir global, `sts.amazonaws.com`.
- **regional**— Alat SDK atau selalu menggunakan AWS STS titik akhir untuk Wilayah yang saat ini dikonfigurasi. Misalnya, jika klien dikonfigurasi untuk digunakan `us-west-2`, semua panggilan ke AWS STS dilakukan ke titik akhir Regional `sts.us-west-2.amazonaws.com`, bukan titik `sts.amazonaws.com` akhir global. Untuk mengirim permintaan ke titik akhir global saat pengaturan ini diaktifkan, Anda dapat mengatur Wilayah ke `aws-global`.

Contoh pengaturan nilai-nilai ini dalam `config` file:

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

Linux/macOS contoh pengaturan variabel lingkungan melalui baris perintah:

```
export AWS_STS_REGIONAL_ENDPOINTS=regional
```

Contoh Windows pengaturan variabel lingkungan melalui baris perintah:

```
setx AWS_STS_REGIONAL_ENDPOINTS regional
```

Kompatibilitas dengan AWS SDKs

Note

Ini adalah praktik AWS terbaik untuk menggunakan titik akhir Regional bila memungkinkan dan untuk mengkonfigurasi Anda [Wilayah AWS](#).

Tabel berikut ini merangkum, untuk alat Anda SDK atau:

- Mendukung pengaturan: Apakah variabel config file bersama dan variabel lingkungan untuk endpoint STS Regional didukung.
- Nilai pengaturan default: Nilai default pengaturan jika didukung.
- Target klien layanan default STS Endpoint: Titik akhir default apa yang digunakan oleh klien meskipun pengaturan untuk mengubahnya tidak tersedia.
- Perilaku fallback klien layanan: SDK Apa yang dilakukan ketika seharusnya menggunakan titik akhir Regional tetapi tidak ada Wilayah yang telah dikonfigurasi. Ini adalah perilaku terlepas dari apakah itu menggunakan titik akhir Regional karena default atau karena `regional` telah dipilih oleh pengaturan.

Tabel ini juga menggunakan nilai-nilai berikut:

- Titik akhir global: `https://sts.amazonaws.com`.
- Titik akhir regional: Berdasarkan konfigurasi yang [Wilayah AWS](#) digunakan oleh aplikasi Anda.

- **us-east-1**(Regional): Menggunakan titik akhir us-east-1 Wilayah tetapi dengan token sesi yang lebih panjang daripada permintaan global biasa.

SDK		Nilai pengaturan default	Target klien layanan default STS Endpoint	Perilaku fallback klien layanan	Catatan atau informasi lebih lanjut
AWS CLI v2	Tid	N/A	Titik akhir Regional	Titik akhir global	
AWS CLI v1	Y	legacy	Titik akhir global	Titik akhir global	
SDK untuk C++	Tid	N/A	Titik akhir Regional	us-east-1 (Regional)	
SDK untuk Go V2 (1.x)	Tid	N/A	Titik akhir Regional	Kegagalan permintaan	
SDK untuk Go 1.x (V1)	Y	legacy	Titik akhir global	Titik akhir global	Untuk menggunakan pengaturan config file bersama, Anda harus mengaktifkan pemuatan dari file konfigurasi; lihat Sesi .
SDK untuk Java 2.x	Tid	N/A	Titik akhir Regional	Kegagalan permintaan	Jika tidak ada Wilayah yang dikonfigurasi, AssumeRole dan AssumeRoleWithWebIdentity akan menggunakan titik STS akhir global
SDK untuk Java 1.x	Y	legacy	Titik akhir global	Titik akhir global	
SDK untuk JavaScript 3.x	Tid	N/A	Titik akhir Regional	Kegagalan permintaan	

SDK		Nilai pengaturan default	Target klien layanan default STS Endpoint	Perilaku fallback klien layanan	Catatan atau informasi lebih lanjut
SDK untuk JavaScript 2.x	Y	legacy	Titik akhir global	Titik akhir global	
SDK untuk Kotlin	Tid	N/A	Titik akhir Regional	Titik akhir global	
SDK untuk .NET 3.x	Y	legacy	Titik akhir global	Titik akhir global	
SDK untuk PHP 3.x	Y	legacy	Titik akhir global	Kegagalan permintaan	
SDK untuk Python (Boto3)	Y	legacy	Titik akhir global	Titik akhir global	
SDK untuk Ruby 3.x	Y	regional	Titik akhir Regional	Kegagalan permintaan	
SDK untuk Rust	Tid	N/A	Titik akhir Regional	Kegagalan permintaan	
SDK untuk Swift	Tid	N/A	Titik akhir Regional	Kegagalan permintaan	
Alat untuk PowerShell	Y	legacy	Titik akhir global	Titik akhir global	

Tumpukan ganda dan titik akhir FIPS

Konfigurasi fungsi ini dengan menggunakan yang berikut ini:

use_dualstack_endpoint- dibagikan AWS **config** pengaturan file, **AWS_USE_DUALSTACK_ENDPOINT**- variabel lingkungan, **aws.useDualstackEndpoint**- properti JVM sistem: Hanya Java/Kotlin

Menghidupkan atau menonaktifkan apakah SDK akan mengirim permintaan ke titik akhir tumpukan ganda. Untuk mempelajari lebih lanjut tentang titik akhir dual-stack, yang mendukung keduanya IPv4 dan IPv6 lalu lintas, lihat Menggunakan [titik akhir tumpukan ganda Amazon S3](#) di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon. Titik akhir dual-stack tersedia untuk beberapa layanan di beberapa wilayah.

Nilai default: `false`

Nilai yang valid:

- **true**— Alat SDK akan mencoba menggunakan titik akhir dual-stack untuk membuat permintaan jaringan. Jika titik akhir tumpukan ganda tidak ada untuk layanan dan/atau Wilayah AWS, permintaan akan gagal.
- **false**— Alat SDK tidak akan menggunakan titik akhir dual-stack untuk membuat permintaan jaringan.

use_fips_endpoint- dibagikan AWS **config** pengaturan file, **AWS_USE_FIPS_ENDPOINT**- variabel lingkungan, **aws.useFipsEndpoint**- properti JVM sistem: Hanya Java/Kotlin

Mengaktifkan atau menonaktifkan apakah alat SDK akan mengirim permintaan ke titik FIPS akhir yang sesuai. Standar Pemrosesan Informasi Federal (FIPS) adalah seperangkat persyaratan keamanan Pemerintah AS untuk data dan enkripsi. Instansi pemerintah, mitra, dan mereka yang ingin melakukan bisnis dengan pemerintah federal diharuskan untuk mematuhi FIPS pedoman. Tidak seperti standar AWS titik akhir, FIPS titik akhir menggunakan pustaka TLS perangkat lunak yang sesuai dengan 140-2. FIPS Jika pengaturan ini diaktifkan dan FIPS titik akhir tidak ada untuk layanan di Wilayah AWS, AWS panggilan mungkin gagal. [Titik akhir khusus layanan](#) dan `--endpoint-url` opsi untuk AWS Command Line Interface tanpa pengaturan ini.

Untuk mempelajari lebih lanjut tentang cara lain untuk menentukan FIPS titik akhir dengan Wilayah AWS, lihat [FIPSTitik Akhir berdasarkan Layanan](#). Untuk informasi selengkapnya tentang titik akhir layanan Amazon Elastic Compute Cloud, lihat Titik akhir [dual-stack \(IPv4 dan IPv6\)](#) di [Referensi](#) Amazon. EC2 API

Nilai default: `false`

Nilai yang valid:

- **true**— Alat SDK akan mengirim permintaan ke titik akhir FIPS yang sesuai.

- **false**— Alat SDK atau tidak akan mengirim permintaan ke titik akhir FIPS yang sesuai.

Kompatibilitas dengan AWS SDKs

Berikut ini SDKs mendukung fitur dan pengaturan yang dijelaskan dalam topik ini. Setiap pengecualian sebagian dicatat. Pengaturan properti JVM sistem apa pun didukung oleh AWS SDK for Java dan AWS SDK for Kotlin hanya.

SDK	Di	Catatan atau informasi lebih lanjut
AWS CLI v2	Ya	
SDK untuk C++	Ya	
SDK untuk Go V2 (1.x)	Ya	
SDK untuk Go 1.x (V1)	Ya	Untuk menggunakan pengaturan config file bersama, Anda harus mengaktifkan pemuatan dari file konfigurasi; lihat Sesi .
SDK untuk Java 2.x	Ya	
SDK untuk Java 1.x	Tida	
SDK untuk JavaScript 3.x	Ya	
SDK untuk JavaScript 2.x	Ya	
SDK untuk Kotlin	Ya	
SDK untuk .NET 3.x	Ya	
SDK untuk PHP 3.x	Ya	
SDK untuk Python (Boto3)	Ya	
SDK untuk Ruby 3.x	Ya	
SDK untuk Rust	Ya	
SDK untuk Swift	Ya	

SDK	Di	Catatan atau informasi lebih lanjut
Alat untuk PowerShell	Ya	

Penemuan titik akhir

SDK menggunakan penemuan titik akhir untuk mengakses titik akhir layanan (URLs untuk mengakses berbagai sumber daya), sambil tetap mempertahankan fleksibilitas AWS untuk mengubah sesuai URLs kebutuhan. Dengan cara ini, kode Anda dapat secara otomatis mendeteksi titik akhir baru. Tidak ada titik akhir tetap untuk beberapa layanan. Sebagai gantinya, Anda mendapatkan titik akhir yang tersedia selama runtime dengan membuat permintaan untuk mendapatkan titik akhir terlebih dahulu. Setelah mengambil endpoint yang tersedia, kode kemudian menggunakan endpoint untuk mengakses operasi lain. Misalnya, untuk Amazon Timestream, SDK membuat `DescribeEndpoints` permintaan untuk mengambil titik akhir yang tersedia, dan kemudian menggunakan titik akhir tersebut untuk menyelesaikan operasi tertentu seperti `atau. CreateDatabase CreateTable`

Konfigurasi fungsi ini dengan menggunakan yang berikut ini:

endpoint_discovery_enabled- dibagikan AWS **config** pengaturan file,
AWS_ENABLE_ENDPOINT_DISCOVERY- variabel lingkungan, **aws.endpointDiscoveryEnabled**- properti JVM sistem: Hanya Java/Kotlin, Untuk mengonfigurasi nilai secara langsung dalam kode, konsultasikan spesifik Anda SDK secara langsung.

Mengaktifkan atau menonaktifkan penemuan titik akhir untuk DynamoDB.

Penemuan titik akhir diperlukan di Timestream dan opsional di Amazon DynamoDB. Pengaturan ini default ke salah satu `true` atau `false` tergantung pada apakah layanan memerlukan penemuan titik akhir. Timestream meminta default ke `true`, dan Amazon DynamoDB meminta default ke `false`

Nilai yang valid:

- **true**— SDK Harus secara otomatis mencoba menemukan titik akhir untuk layanan di mana penemuan titik akhir adalah opsional.
- **false**— SDK Seharusnya tidak secara otomatis mencoba menemukan titik akhir untuk layanan di mana penemuan titik akhir bersifat opsional.

Kompatibilitas dengan AWS SDKs

Berikut ini SDKs mendukung fitur dan pengaturan yang dijelaskan dalam topik ini. Setiap pengecualian sebagian dicatat. Pengaturan properti JVM sistem apa pun didukung oleh AWS SDK for Java dan AWS SDK for Kotlin hanya.

SDK	Di	Catatan atau informasi lebih lanjut
AWS CLI v2	Ya	
SDK untuk C++	Ya	
SDK untuk Go V2 (1.x)	Ya	
SDK untuk Go 1.x (V1)	Ya	Untuk menggunakan pengaturan config file bersama, Anda harus mengaktifkan pemuatan dari file konfigurasi; lihat Sesi .
SDK untuk Java 2.x	Ya	SDK untuk Java 2.x digunakan <code>AWS_ENDPOINT_DISCOVERY_ENABLED</code> untuk nama variabel lingkungan.
SDK untuk Java 1.x	Parti	JVM properti sistem tidak didukung.
SDK untuk JavaScript 3.x	Ya	
SDK untuk JavaScript 2.x	Ya	
SDK untuk Kotlin	Ya	
SDK untuk .NET 3.x	Ya	
SDK untuk PHP 3.x	Ya	
SDK untuk Python (Boto3)	Ya	
SDK untuk Ruby 3.x	Ya	
SDK untuk Rust	Parti	Didukung hanya untuk Timestream.
SDK untuk Swift	Tida	
Alat untuk PowerShell	Ya	

Pengaturan konfigurasi umum

SDK mendukung beberapa pengaturan umum yang mengonfigurasi SDK perilaku keseluruhan.

Konfigurasi fungsi ini dengan menggunakan yang berikut ini:

api_versions- dibagikan AWS **config** pengaturan file

Beberapa AWS layanan mempertahankan beberapa API versi untuk mendukung kompatibilitas mundur. Secara default, SDK dan AWS CLI operasi menggunakan API versi terbaru yang tersedia. Untuk meminta API versi tertentu untuk digunakan untuk permintaan Anda, sertakan `api_versions` pengaturan di profil Anda.

Nilai default: Tidak ada. (API Versi terbaru digunakan oleh SDK.)

Nilai yang valid: Ini adalah pengaturan bersarang yang diikuti oleh satu atau lebih baris indentasi yang masing-masing mengidentifikasi satu AWS layanan dan API versi yang akan digunakan. Lihat dokumentasi untuk AWS layanan untuk memahami API versi mana yang tersedia.

Contoh menetapkan API versi tertentu untuk dua AWS layanan dalam `config` file. API Versi ini hanya digunakan untuk perintah yang berjalan di bawah profil yang berisi pengaturan ini. Perintah untuk layanan lain menggunakan versi terbaru dari layanan itu API.

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

ca_bundle- dibagikan AWS **config** pengaturan file, **AWS_CA_BUNDLE**- variabel lingkungan

Menentukan jalur ke bundel sertifikat kustom (file dengan `.pem` ekstensi) untuk digunakan saat membuat SSL/TLS koneksi.

Nilai default: tidak ada

Nilai yang valid: Tentukan jalur lengkap atau nama file dasar. Jika ada nama file dasar, sistem mencoba untuk menemukan program dalam folder yang ditentukan oleh variabel `PATH` lingkungan.

Contoh pengaturan nilai ini dalam `config` file:

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

Karena perbedaan dalam cara sistem operasi menangani jalur dan melarikan diri dari karakter jalur, berikut ini adalah contoh pengaturan nilai ini dalam config file di Windows:

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

Linux/macOS contoh pengaturan variabel lingkungan melalui baris perintah:

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

Contoh Windows pengaturan variabel lingkungan melalui baris perintah:

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

output- dibagikan AWS **config** pengaturan file

Menentukan bagaimana hasil diformat dalam AWS CLI dan lainnya AWS SDKs dan alat-alat.

Nilai default: `json`

Nilai yang valid:

- **json**— Output diformat sebagai [JSON](#) string.
- **yaml**— Output diformat sebagai [YAML](#) string.
- **yaml-stream**— Output dialirkan dan diformat sebagai [YAML](#) string. Streaming memungkinkan penanganan tipe data besar yang lebih cepat.
- **text**— Output diformat sebagai beberapa baris nilai string yang dipisahkan tab. Ini dapat berguna untuk meneruskan output ke prosesor teks, seperti `grep`, `sed`, atau `awk`.
- **table**— Output diformat sebagai tabel menggunakan karakter `+|-` untuk membentuk batas sel. Ini biasanya menyajikan informasi dalam format “ramah manusia” yang jauh lebih mudah dibaca daripada yang lain, tetapi tidak berguna secara terprogram.

parameter_validation- dibagikan AWS **config** pengaturan file

Menentukan apakah SDK atau alat mencoba untuk memvalidasi parameter baris perintah sebelum mengirim mereka ke AWS titik akhir layanan.

Nilai default: `true`

Nilai yang valid:

- **true** – Default. Alat SDK melakukan validasi sisi klien dari parameter baris perintah. Ini membantu SDK atau alat mengkonfirmasi bahwa parameter valid, dan menangkap beberapa kesalahan. Alat SDK dapat menolak permintaan yang tidak valid sebelum mengirim permintaan ke AWS titik akhir layanan.
- **false**— Alat SDK atau tidak memvalidasi parameter baris perintah sebelum mengirimnya ke AWS titik akhir layanan. Bagian AWS titik akhir layanan bertanggung jawab untuk memvalidasi semua permintaan dan menolak permintaan yang tidak valid.

Kompatibilitas dengan AWS SDKs

Berikut ini SDKs mendukung fitur dan pengaturan yang dijelaskan dalam topik ini. Setiap pengecualian sebagian dicatat. Pengaturan properti JVM sistem apa pun didukung oleh AWS SDK for Java dan AWS SDK for Kotlin hanya.

SDK	Di Catatan atau informasi lebih lanjut
AWS CLI v2	Parsi <code>api_versions</code> tidak didukung.
SDK untuk C++	Ya
SDK untuk Go V2 (1.x)	Parsi <code>api_versions</code> dan <code>parameter_validation</code> tidak didukung.
SDK untuk Go 1.x (V1)	Parsi <code>api_versions</code> dan <code>parameter_validation</code> tidak didukung. Untuk menggunakan pengaturan config file bersama, Anda harus mengaktifkan pemuatan dari file konfigurasi; lihat Sesi .
SDK untuk Java 2.x	Tida
SDK untuk Java 1.x	Tida
SDK untuk JavaScript 3.x	Ya
SDK untuk JavaScript 2.x	Ya
SDK untuk Kotlin	Tida
SDK untuk .NET 3.x	Tida

SDK	Di	Catatan atau informasi lebih lanjut
SDK untuk PHP 3.x	Ya	
SDK untuk Python (Boto3)	Ya	
SDK untuk Ruby 3.x	Ya	
SDK untuk Rust	Tida	
SDK untuk Swift	Tida	
Alat untuk PowerShell	Tida	

IMDS klien

SDK mengimplementasikan klien Layanan Metadata Instans Versi 2 (IMDSv2) menggunakan permintaan berorientasi sesi. Untuk informasi selengkapnya IMDSv2, lihat [Menggunakan IMDSv2](#) di Panduan EC2 Pengguna Amazon. IMDS Klien dapat dikonfigurasi melalui objek konfigurasi klien yang tersedia di basis SDK kode.

Konfigurasi fungsi ini dengan menggunakan yang berikut ini:

retries- anggota objek konfigurasi klien

Jumlah upaya coba lagi tambahan untuk setiap permintaan yang gagal.

Nilai default: 3

Nilai yang valid: Angka lebih besar dari 0.

port- anggota objek konfigurasi klien

Port untuk titik akhir.

Nilai default: 80

Nilai yang valid: Nomor.

token_ttl- anggota objek konfigurasi klien

TTL dari token.

Nilai default: 21.600 detik (6 jam, waktu maksimum yang dialokasikan).

Nilai yang valid: Nomor.

endpoint- anggota objek konfigurasi klien

Titik akhir dari. IMDS

Nilai default: Jika `endpoint_mode` samaIPv4, maka titik akhir default adalah.

`http://169.254.169.254` Jika `endpoint_mode` samaIPv6, maka titik akhir default adalah.

`http://[fd00:ec2::254]`

Nilai yang valid: ValidURI.

Opsi berikut didukung oleh sebagian besar SDKs. Lihat basis SDK kode spesifik Anda untuk detailnya.

endpoint_mode- anggota objek konfigurasi klien

Modus endpoint dari. IMDS

Nilai default: IPv4

Nilai valid: IPv4, IPv6

http_open_timeout- anggota objek konfigurasi klien (nama dapat bervariasi)

Jumlah detik untuk menunggu koneksi terbuka.

Nilai default: 1 detik.

Nilai yang valid: Angka lebih besar dari 0.

http_read_timeout- anggota objek konfigurasi klien (nama dapat bervariasi)

Jumlah detik untuk satu potongan data yang akan dibaca.

Nilai default: 1 detik.

Nilai yang valid: Angka lebih besar dari 0.

http_debug_output- anggota objek konfigurasi klien (nama dapat bervariasi)

Menetapkan aliran output untuk debugging.

Nilai default: Tidak ada.

Nilai yang valid: Aliran I/O yang valid, sepertiSTDOUT.

backoff- anggota objek konfigurasi klien (nama dapat bervariasi)

Jumlah detik untuk tidur di antara percobaan ulang atau pelanggan menyediakan fungsi backoff untuk menelepon. Ini mengesampingkan strategi backoff eksponensial default.

Nilai default: Bervariasi berdasarkan SDK.

Nilai yang valid: Bervariasi berdasarkan SDK. Dapat berupa nilai numerik atau panggilan keluar ke fungsi kustom.

Kompatibilitas dengan AWS SDKs

Berikut ini SDKs mendukung fitur dan pengaturan yang dijelaskan dalam topik ini. Setiap pengecualian sebagian dicatat. Setiap pengaturan properti JVM sistem didukung oleh AWS SDK for Java dan AWS SDK for Kotlin satu-satunya.

SDK	Didukung	Catatan atau informasi lebih lanjut
AWS CLI v2	Ya	
SDK untuk C++	Tidak	
SDK untuk Go V2 (1.x)	Ya	
SDK untuk Go 1.x (V1)	Ya	
SDK untuk Java 2.x	Ya	
SDK untuk Java 1.x	Ya	
SDK untuk JavaScript 3.x	Ya	
SDK untuk JavaScript 2.x	Ya	
SDK untuk Kotlin	Tidak	
SDK untuk .NET 3.x	Ya	
SDK untuk PHP 3.x	Ya	
SDK untuk Python (Boto3)	Ya	

SDK	Di Catatan atau informasi lebih lanjut
SDK untuk Ruby 3.x	Ya
SDK untuk Rust	Ya
SDK untuk Swift	Ya
Alat untuk PowerShell	Ya

Coba lagi perilaku

Perilaku coba lagi mencakup pengaturan mengenai bagaimana SDKs upaya untuk memulihkan dari kegagalan yang dihasilkan dari permintaan yang dibuat Layanan AWS.

Konfigurasi fungsi ini dengan menggunakan yang berikut ini:

retry_mode- dibagikan AWS **config** pengaturan file, **AWS_RETRY_MODE**- variabel lingkungan, **aws.retryMode**- properti JVM sistem: Hanya Java/Kotlin

Menentukan bagaimana SDK atau alat pengembang mencoba mencoba ulang.

Nilai default: Nilai ini khusus untuk Anda SDK. Periksa SDK panduan spesifik Anda atau basis kode Anda SDK untuk defaultnya `retry_mode`.

Nilai yang valid:

- **standard**— (Disarankan) Seperangkat aturan coba lagi yang direkomendasikan di seluruh AWS SDKs. Mode ini mencakup serangkaian kesalahan standar yang dicoba ulang, dan secara otomatis menyesuaikan jumlah percobaan ulang untuk memaksimalkan ketersediaan dan stabilitas. Mode ini aman untuk digunakan dalam aplikasi multi-tenant. Jumlah maksimum percobaan default dengan mode ini adalah tiga, kecuali `max_attempts` dikonfigurasi secara eksplisit.
- **adaptive**— Mode coba lagi, hanya sesuai untuk kasus penggunaan khusus, yang mencakup fungsionalitas mode standar serta pembatasan laju sisi klien otomatis. Mode coba lagi ini tidak disarankan untuk aplikasi multi-penyewa, kecuali jika Anda berhati-hati untuk mengisolasi penyewa aplikasi. Untuk informasi selengkapnya, lihat [Memilih antara standard dan adaptive coba lagi mode](#). Mode ini bersifat eksperimental dan mungkin mengubah perilaku di masa depan.

- `legacy`— (Tidak Disarankan) Khusus untuk Anda SDK (periksa SDK panduan spesifik Anda SDK atau basis kode Anda).

max_attempts- dibagikan AWS **config** pengaturan file, **AWS_MAX_ATTEMPTS**- variabel lingkungan, **aws.maxAttempts**- properti JVM sistem: Hanya Java/Kotlin

Menentukan jumlah maksimum upaya untuk membuat atas permintaan.

Nilai default: Jika nilai ini tidak ditentukan, defaultnya tergantung pada nilai `retry_mode` pengaturan:

- Jika `retry_mode` ya `legacy` — Menggunakan nilai default khusus untuk Anda SDK (periksa SDK panduan spesifik Anda SDK atau basis kode Anda untuk `max_attempts` default).
- Jika `retry_mode` ada `standard` — Membuat tiga upaya.
- Jika `retry_mode` ada `adaptive` — Membuat tiga upaya.

Nilai yang valid: Angka lebih besar dari 0.

Memilih antara **standard** dan **adaptive** coba lagi mode

Kami menyarankan Anda menggunakan mode `standard` coba lagi kecuali Anda yakin bahwa penggunaan Anda lebih cocok untuk `adaptive`.

Note

`adaptiveMode` mengasumsikan bahwa Anda mengumpulkan klien berdasarkan ruang lingkup di mana layanan backend dapat membatasi permintaan. Jika Anda tidak melakukan ini, pembatasan dalam satu sumber daya dapat menunda permintaan untuk sumber daya yang tidak terkait jika Anda menggunakan klien yang sama untuk kedua sumber daya.

Standar	Adaptif
Kasus penggunaan aplikasi: Semua.	Kasus penggunaan aplikasi: <ol style="list-style-type: none"> 1. Tidak sensitif terhadap latensi. 2. Klien hanya mengakses satu sumber daya, atau, Anda menyediakan logika untuk mengumpulkan klien Anda secara terpisah

Standar	Adaptif
	oleh sumber daya layanan yang sedang diakses.
Mendukung pemutusan sirkuit untuk mencegah mencoba lagi selama SDK pemadaman.	Mendukung pemutusan sirkuit untuk mencegah mencoba lagi selama SDK pemadaman.
Menggunakan backoff eksponensial gelisah jika terjadi kegagalan.	Menggunakan durasi backoff dinamis untuk mencoba meminimalkan jumlah permintaan yang gagal, dengan imbalan potensi peningkatan latensi.
Jangan pernah menunda upaya permintaan pertama, hanya percobaan ulang.	Dapat membatasi atau menunda upaya permintaan awal.

Jika Anda memilih untuk menggunakan `adaptive mode`, aplikasi Anda harus membangun klien yang dirancang di sekitar setiap sumber daya yang mungkin dibatasi. Sumber daya, dalam hal ini, disetel lebih baik daripada hanya memikirkan masing-masing Layanan AWS. Layanan AWS dapat memiliki dimensi tambahan yang mereka gunakan untuk membatasi permintaan. Mari kita gunakan layanan Amazon DynamoDB sebagai contoh. DynamoDB menggunakan Wilayah AWS ditambah tabel yang diakses ke permintaan throttle. Ini berarti bahwa satu tabel yang diakses kode Anda mungkin dibatasi lebih dari yang lain. Jika kode Anda menggunakan klien yang sama untuk mengakses semua tabel, dan permintaan ke salah satu tabel tersebut dibatasi, maka mode coba lagi adaptif akan mengurangi tingkat permintaan untuk semua tabel. Kode Anda harus dirancang untuk memiliki satu klien per `region-and-table` pasangan R. Jika Anda mengalami latensi tak terduga saat menggunakan `adaptive mode`, lihat spesifiknya AWS panduan dokumentasi untuk layanan yang Anda gunakan.

Coba lagi detail implementasi mode

Bagian AWS SDK menggunakan [`ember token`](#) untuk memutuskan apakah permintaan harus dicoba lagi dan (dalam kasus mode `adaptive` coba lagi) seberapa cepat permintaan harus dikirim. Dua bucket token digunakan oleh SDK: `ember token` coba lagi dan `bucket token rate` permintaan.

- Bucket token coba lagi digunakan untuk menentukan apakah SDK harus menonaktifkan percobaan ulang sementara untuk melindungi layanan hulu dan hilir selama pemadaman. Token diperoleh dari bucket sebelum percobaan ulang dicoba, dan token dikembalikan ke bucket saat permintaan

berhasil. Jika bucket kosong saat percobaan ulang dicoba, permintaan tidak SDK akan mencoba lagi.

- Bucket token rate permintaan hanya digunakan dalam mode `adaptive` coba lagi untuk menentukan tingkat pengiriman permintaan. Token diperoleh dari bucket sebelum permintaan dikirim, dan token dikembalikan ke bucket pada tingkat yang ditentukan secara dinamis berdasarkan respons pelambatan yang dikembalikan oleh layanan.

Berikut ini adalah pseudocode tingkat tinggi untuk mode `standard` dan `adaptive` coba lagi:

```
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)
  }
}
```

Berikut ini adalah rincian lebih lanjut tentang komponen yang digunakan dalam pseudocode:

GetSendToken:

Langkah ini hanya digunakan dalam mode `adaptive` coba lagi. Langkah ini memperoleh token dari bucket token rate permintaan. Jika token tidak tersedia, itu akan menunggu hingga tersedia. Anda SDK mungkin memiliki opsi konfigurasi yang tersedia untuk gagal permintaan alih-alih menunggu. Token dalam bucket diisi ulang pada tingkat yang ditentukan secara dinamis, berdasarkan jumlah respons pelambatan yang diterima oleh klien.

SendHTTPRequest:

Langkah ini mengirimkan permintaan ke AWS. Kebanyakan AWS SDK menggunakan HTTP pustaka yang menggunakan kumpulan koneksi untuk menggunakan kembali koneksi yang ada saat membuat

HTTP permintaan. Umumnya, koneksi digunakan kembali jika permintaan gagal karena kesalahan pelambatan tetapi tidak jika permintaan gagal karena kesalahan sementara.

RequestBookkeeping:

Token ditambahkan ke ember token jika permintaan berhasil. Hanya untuk mode adaptive coba lagi, tingkat pengisian bucket token rate permintaan diperbarui berdasarkan jenis respons yang diterima.

Retryable:

Langkah ini menentukan apakah respons dapat dicoba ulang berdasarkan hal berikut:

- Kode HTTP status.
- Kode kesalahan dikembalikan dari layanan.
- Kesalahan koneksi, didefinisikan sebagai kesalahan yang diterima oleh SDK di mana HTTP respons dari layanan tidak diterima.

Kesalahan sementara (kode HTTP status 400, 408, 500, 502, 503, dan 504) dan kesalahan pelambatan (kode HTTP status 400, 403, 429, 502, 503, dan 509) semuanya berpotensi dicoba ulang. SDK perilaku coba lagi ditentukan dalam kombinasi dengan kode kesalahan atau data lain dari layanan.

MAX_ATTEMPTS:

Jumlah default upaya maksimum diatur oleh `retry_mode` pengaturan, kecuali diganti oleh pengaturan. `max_attempts`

HasRetryQuota

Langkah ini memperoleh token dari ember token coba lagi. Jika ember token coba lagi kosong, permintaan tidak akan dicoba lagi.

ExponentialBackoff

Untuk kesalahan yang dapat dicoba lagi, penundaan coba lagi dihitung menggunakan backoff eksponensial terpotong. SDKs Penggunaan backoff eksponensial biner terpotong dengan jitter. Algoritma berikut menunjukkan bagaimana jumlah waktu tidur, dalam detik, didefinisikan untuk respons permintaani:

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

Dalam algoritma sebelumnya, nilai-nilai berikut berlaku:

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

$r = 2$

`MAX_BACKOFF = 20 seconds` untuk sebagian besar SDKs. Lihat SDK panduan atau kode sumber khusus Anda untuk konfirmasi.

Kompatibilitas dengan AWS SDKs

Berikut ini SDKs mendukung fitur dan pengaturan yang dijelaskan dalam topik ini. Setiap pengecualian sebagian dicatat. Pengaturan properti JVM sistem apa pun didukung oleh AWS SDK for Java dan AWS SDK for Kotlin hanya.

SDK	Di	Catatan atau informasi lebih lanjut
AWS CLI v2	Ya	
SDK untuk C++	Ya	
SDK untuk Go V2 (1.x)	Ya	
SDK untuk Go 1.x (V1)	Tida	
SDK untuk Java 2.x	Ya	
SDK untuk Java 1.x	Ya	JVM properti sistem: gunakan <code>com.amazonaws.sdk.maxAttempts</code> sebagai pengganti <code>aws.maxAttempts</code> ; gunakan <code>com.amazonaws.sdk.retryMode</code> sebagai pengganti <code>aws.retryMode</code> .
SDK untuk JavaScript 3.x	Ya	
SDK untuk JavaScript 2.x	Tida	Mendukung jumlah percobaan ulang maksimum, backoff eksponensial dengan jitter, dan opsi untuk metode khusus untuk coba lagi backoff.
SDK untuk Kotlin	Ya	

SDK	Di	Catatan atau informasi lebih lanjut
SDKuntuk .NET3.x	Ya	
SDKuntuk PHP 3.x	Ya	
SDKuntuk Python (Boto3)	Ya	
SDKuntuk Ruby 3.x	Ya	
SDKuntuk Rust	Ya	
SDKuntuk Swift	Ya	
Alat untuk PowerShell	Ya	

Minta kompresi

Note

Untuk bantuan dalam memahami tata letak halaman pengaturan, atau dalam menafsirkan Kompatibilitas dengan AWS SDKstabel berikut, lihat [Halaman pengaturan](#).

AWS SDKsdan alat dapat secara otomatis mengompres muatan saat mengirim permintaan Layanan AWS yang mendukung menerima muatan terkompresi. Mengompresi payload pada klien sebelum mengirimnya ke layanan dapat mengurangi jumlah keseluruhan permintaan dan bandwidth yang diperlukan untuk mengirim data ke layanan, serta mengurangi permintaan yang gagal karena keterbatasan layanan pada ukuran payload. Untuk kompresi, alat SDK or memilih algoritma pengkodean yang didukung oleh layanan dan. SDK Namun, daftar kemungkinan pengkodean saat ini hanya terdiri dari gzip, tetapi dapat berkembang di masa depan.

Kompresi permintaan dapat sangat berguna jika aplikasi Anda menggunakan [Amazon CloudWatch](#). CloudWatch adalah layanan pemantauan dan observabilitas yang mengumpulkan data pemantauan dan operasional dalam bentuk log, metrik, dan peristiwa. Salah satu contoh operasi layanan yang mendukung kompresi CloudWatch adalah [PutMetricDataAPI](#)metode.

Konfigurasi fungsi ini dengan menggunakan yang berikut ini:

disable_request_compression- dibagikan AWS **config** pengaturan file, **AWS_DISABLE_REQUEST_COMPRESSION**- variabel lingkungan, **aws.disableRequestCompression**- properti JVM sistem: Hanya Java/Kotlin

Menghidupkan atau menonaktifkan apakah alat SDK atau akan memampatkan muatan sebelum mengirim permintaan.

Nilai default: `false`

Nilai yang valid:

- **true**— Matikan kompresi permintaan.
- **false**— Gunakan kompresi permintaan bila memungkinkan.

request_min_compression_size_bytes- dibagikan AWS **config** pengaturan file, **AWS_REQUEST_MIN_COMPRESSION_SIZE_BYTES**- variabel lingkungan, **aws.requestMinCompressionSizeBytes**- properti JVM sistem: Hanya Java/Kotlin

Menetapkan ukuran minimum dalam byte dari badan permintaan yang harus dikompres oleh alat SDK atau. Muatan kecil dapat menjadi lebih lama ketika dikompresi, dengan demikian, ada batas bawah di mana masuk akal untuk melakukan kompresi. Nilai ini inklusif, ukuran permintaan lebih besar dari atau sama dengan nilai dikompresi.

Nilai default: 10240 byte

Nilai valid: Nilai integer antara 0 dan 10485760 byte inklusif.

Kompatibilitas dengan AWS SDKs

Berikut ini SDKs mendukung fitur dan pengaturan yang dijelaskan dalam topik ini. Setiap pengecualian sebagian dicatat. Pengaturan properti JVM sistem apa pun didukung oleh AWS SDK for Java dan AWS SDK for Kotlin hanya.

SDK	Di Catatan atau informasi lebih lanjut
AWS CLI v2	Ya
SDK untuk C++	Ya
SDK untuk Go V2 (1.x)	Ya

SDK	Di	Catatan atau informasi lebih lanjut
SDK untuk Go 1.x (V1)	Tida	
SDK untuk Java 2.x	Ya	
SDK untuk Java 1.x	Tida	
SDK untuk JavaScript 3.x	Ya	
SDK untuk JavaScript 2.x	Tida	
SDK untuk Kotlin	Ya	
SDK untuk .NET 3.x	Ya	
SDK untuk PHP 3.x	Ya	
SDK untuk Python (Boto3)	Ya	
SDK untuk Ruby 3.x	Ya	
SDK untuk Rust	Ya	
SDK untuk Swift	Tida	
Alat untuk PowerShell	Ya	

Titik akhir khusus layanan

Konfigurasi titik akhir khusus layanan menyediakan opsi untuk menggunakan titik akhir yang Anda pilih untuk API permintaan dan agar pilihan itu tetap ada. Pengaturan ini memberikan fleksibilitas untuk mendukung titik akhir lokal, VPC titik akhir, dan lokal pihak ketiga AWS lingkungan pengembangan. Titik akhir yang berbeda dapat digunakan untuk lingkungan pengujian dan produksi. Anda dapat menentukan titik akhir URL untuk individu Layanan AWS.

Konfigurasi fungsi ini dengan menggunakan yang berikut ini:

endpoint_url- dibagikan AWS **config** pengaturan file, **AWS_ENDPOINT_URL**- variabel lingkungan, **aws.endpointUrl**- properti JVM sistem: Hanya Java/Kotlin

Ketika ditentukan langsung dalam profil atau sebagai variabel lingkungan, pengaturan ini menentukan titik akhir yang digunakan untuk semua permintaan layanan. Titik akhir ini diganti oleh titik akhir khusus layanan yang dikonfigurasi.

Anda juga dapat menggunakan setelan ini dalam **services** bagian yang dibagikan AWS **config** file untuk menetapkan titik akhir kustom untuk layanan tertentu. Untuk daftar semua kunci pengenalan layanan yang akan digunakan untuk subbagian dalam **services** bagian, lihat [Pengidentifikasi untuk titik akhir khusus layanan](#)

Nilai default: none

Nilai yang valid: A URL termasuk skema dan host untuk titik akhir. Secara opsional URL dapat berisi komponen jalur yang berisi satu atau lebih segmen jalur.

AWS_ENDPOINT_URL_<SERVICE>- variabel lingkungan, **aws.endpointUrl<ServiceName>**- properti JVM sistem: Hanya Java/Kotlin

AWS_ENDPOINT_URL_<SERVICE>, **<SERVICE>** di mana Layanan AWS identifier, menetapkan titik akhir kustom untuk layanan tertentu. Untuk daftar semua variabel lingkungan khusus layanan, lihat [Pengidentifikasi untuk titik akhir khusus layanan](#)

Titik akhir khusus layanan ini mengesampingkan titik akhir global apa pun yang ditetapkan.

AWS_ENDPOINT_URL

Nilai default: none

Nilai yang valid: A URL termasuk skema dan host untuk titik akhir. Secara opsional URL dapat berisi komponen jalur yang berisi satu atau lebih segmen jalur.

ignore_configured_endpoint_urls- dibagikan AWS **config** pengaturan file, **AWS_IGNORE_CONFIGURED_ENDPOINT_URLS**- variabel lingkungan, **aws.ignoreConfiguredEndpointUrls**- properti JVM sistem: Hanya Java/Kotlin

Pengaturan ini digunakan untuk mengabaikan semua konfigurasi titik akhir kustom.

Perhatikan bahwa setiap titik akhir eksplisit yang ditetapkan dalam kode atau pada klien layanan itu sendiri digunakan terlepas dari pengaturan ini. Misalnya, termasuk parameter baris `--endpoint-url` perintah dengan AWS CLI perintah atau meneruskan titik akhir URL ke konstruktor klien akan selalu berlaku.

Nilai default: `false`

Nilai yang valid:

- **true**— Alat SDK atau tidak membaca opsi konfigurasi khusus apa pun dari `config` file bersama atau dari variabel lingkungan untuk mengatur titik akhir URL.
- **false**— Alat SDK atau menggunakan titik akhir yang disediakan pengguna yang tersedia dari `config` file bersama atau dari variabel lingkungan.

Konfigurasi titik akhir menggunakan variabel lingkungan

Untuk merutekan permintaan semua layanan ke titik akhir kustom URL, tetapkan variabel lingkungan `AWS_ENDPOINT_URL` global.

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

Untuk merutekan permintaan untuk yang spesifik Layanan AWS ke titik akhir khusus URL, gunakan variabel `AWS_ENDPOINT_URL_<SERVICE>` lingkungan. Amazon DynamoDB memiliki `serviceId` dari [DynamoDB](#). Untuk layanan ini, variabel URL lingkungan endpoint adalah `AWS_ENDPOINT_URL_DYNAMODB`. Titik akhir ini lebih diutamakan daripada titik akhir global yang ditetapkan untuk layanan ini. `AWS_ENDPOINT_URL`

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

Sebagai contoh lain, AWS Elastic Beanstalk memiliki `serviceId` dari [Elastic Beanstalk](#). Bagian Layanan AWS identifier didasarkan pada API model `serviceId` dengan mengganti semua spasi dengan garis bawah dan huruf atas semua huruf. Untuk mengatur titik akhir untuk layanan ini, variabel lingkungan yang sesuai adalah `AWS_ENDPOINT_URL_ELASTIC_BEANSTALK`. Untuk daftar semua variabel lingkungan khusus layanan, lihat. [Pengidentifikasi untuk titik akhir khusus layanan](#)

```
export AWS_ENDPOINT_URL_ELASTIC_BEANSTALK=http://localhost:5567
```

Konfigurasi titik akhir menggunakan file bersama **config**

Dalam `config` file bersama, `endpoint_url` digunakan di tempat yang berbeda untuk fungsionalitas yang berbeda.

- `endpoint_url` ditentukan secara langsung dalam a profile menjadikan titik akhir itu titik akhir global.

- `endpoint_url` bersarang di bawah kunci pengenalan layanan dalam `services` bagian membuat titik akhir tersebut berlaku untuk permintaan yang dibuat hanya untuk layanan tersebut. Untuk detail tentang mendefinisikan `services` bagian dalam config file bersama Anda, lihat [Format file konfigurasi](#).

Contoh berikut menggunakan `services` definisi untuk mengonfigurasi titik akhir khusus layanan yang akan digunakan URL untuk Amazon S3 dan titik akhir global khusus yang akan digunakan untuk semua layanan lainnya:

```
[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
```

Satu profil dapat mengonfigurasi titik akhir untuk beberapa layanan. Contoh ini menunjukkan cara menyetel titik akhir khusus layanan untuk Amazon URLs S3 dan AWS Elastic Beanstalk dalam profil yang sama. AWS Elastic Beanstalk memiliki `serviceId` dari [Elastic Beanstalk](#). Bagian Layanan AWS identifier didasarkan pada API model `serviceId` dengan mengganti semua spasi dengan garis bawah dan huruf kecil semua huruf. Dengan demikian, kunci pengidentifikasi layanan menjadi `elastic_beanstalk` dan pengaturan untuk layanan ini dimulai pada `teleponelastic_beanstalk =` . Untuk daftar semua kunci pengenalan layanan yang akan digunakan di `services` bagian ini, lihat [Pengidentifikasi untuk titik akhir khusus layanan](#).

```
[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
```

Bagian konfigurasi layanan dapat digunakan dari beberapa profil. Misalnya, dua profil dapat menggunakan `services` definisi yang sama sambil mengubah properti profil lainnya:

```
[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
```

Konfigurasi titik akhir di profil menggunakan kredensial berbasis peran

Jika profil Anda memiliki kredensial berbasis peran yang dikonfigurasi melalui `source_profile` parameter untuk fungsi peran IAM asumsikan, SDK hanya menggunakan konfigurasi layanan untuk profil yang ditentukan. Itu tidak menggunakan profil yang dirantai peran untuk itu. Misalnya, menggunakan config file bersama berikut:

```
[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
```

Jika Anda menggunakan profil B dan membuat panggilan dalam kode Anda ke AmazonEC2, titik akhir akan diselesaikan sebagai `https://profile-b-ec2-endpoint.aws`. Jika kode Anda membuat permintaan ke layanan lain, resolusi titik akhir tidak akan mengikuti logika kustom apa pun. Titik akhir tidak menyelesaikan titik akhir global yang ditentukan dalam profil. A Agar titik akhir global berlaku untuk profil B, Anda perlu mengatur `endpoint_url` langsung di dalam profil B. Untuk informasi lebih lanjut tentang `source_profile` pengaturan, lihat [Asumsikan penyedia kredensial peran](#).

Prioritas pengaturan

Pengaturan untuk fitur ini dapat digunakan pada saat yang sama tetapi hanya satu nilai yang akan diprioritaskan per layanan. Untuk API panggilan yang dilakukan ke yang diberikan Layanan AWS, urutan berikut digunakan untuk memilih nilai:

1. Pengaturan eksplisit apa pun yang disetel dalam kode atau pada klien layanan itu sendiri lebih diutamakan daripada yang lain.
 - Untuk AWS CLI, ini adalah nilai yang disediakan oleh parameter baris `--endpoint-url` perintah. Untuk SDK, tugas eksplisit dapat mengambil bentuk parameter yang Anda tetapkan saat Anda membuat instance Layanan AWS klien atau objek konfigurasi.
2. Nilai yang diberikan oleh variabel lingkungan khusus layanan seperti `AWS_ENDPOINT_URL_DYNAMODB`
3. Nilai yang diberikan oleh variabel lingkungan endpoint `AWS_ENDPOINT_URL` global.
4. Nilai yang diberikan oleh `endpoint_url` pengaturan bersarang di bawah kunci pengenalan layanan dalam `services` bagian file bersama `config`.
5. Nilai yang diberikan oleh `endpoint_url` pengaturan ditentukan langsung `profile` dalam `config` file bersama.
6. Setiap titik akhir default URL untuk masing-masing Layanan AWS digunakan terakhir.

Kompatibilitas dengan AWS SDKs

Berikut ini SDKs mendukung fitur dan pengaturan yang dijelaskan dalam topik ini. Setiap pengecualian sebagian dicatat. Pengaturan properti JVM sistem apa pun didukung oleh AWS SDK for Java dan AWS SDK for Kotlin hanya.

SDK	Di Catatan atau informasi lebih lanjut
AWS CLI v2	Ya
SDK untuk C++	Tidak
SDK untuk Go V2 (1.x)	Ya
SDK untuk Go 1.x (V1)	Tidak
SDK untuk Java 2.x	Ya

SDK	Di	Catatan atau informasi lebih lanjut
SDK untuk Java 1.x	Tida	
SDK untuk JavaScript 3.x	Ya	
SDK untuk JavaScript 2.x	Tida	
SDK untuk Kotlin	Ya	
SDK untuk .NET 3.x	Ya	
SDK untuk PHP 3.x	Ya	
SDK untuk Python (Boto3)	Ya	
SDK untuk Ruby 3.x	Ya	
SDK untuk Rust	Tida	
SDK untuk Swift	Tida	
Alat untuk PowerShell	Ya	

Pengidentifikasi untuk titik akhir khusus layanan

Untuk informasi tentang bagaimana dan di mana menggunakan pengenal dalam tabel berikut, lihat [Titik akhir khusus layanan](#).

serviceId	Ki Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE> pe la ur di Al Fi co
AccessAnalyzer	aw AWS_ENDPOINT_URL_ACCESSANALYZER ly
Account	aw AWS_ENDPOINT_URL_ACCOUNT
ACM	aw AWS_ENDPOINT_URL_ACM
ACM PCA	aw AWS_ENDPOINT_URL_ACM_PCA
Alexa For Business	aw AWS_ENDPOINT_URL_ALEXA_FOR_BUSINESS _l
amp	aw AWS_ENDPOINT_URL_AMP
Amplify	aw AWS_ENDPOINT_URL_AMPLIFY
AmplifyBackend	aw AWS_ENDPOINT_URL_AMPLIFYBACKEND cl
AmplifyUIBuilder	aw AWS_ENDPOINT_URL_AMPLIFYUIBUILDER bt
API Gateway	aw AWS_ENDPOINT_URL_API_GATEWAY ay
ApiGatewayManagemen tApi	aw AWS_ENDPOINT_URL_APIGATEWAYMANAGEMENTAPI yr nt

serviceId	Ki Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE> pe la ur di Al Fi co
ApiGatewayV2	aj AWS_ENDPOINT_URL_APIGATEWAYV2 y)
AppConfig	aj AWS_ENDPOINT_URL_APPCONFIG
AppConfigData	aj AWS_ENDPOINT_URL_APPCONFIGDATA da
AppFabric	aj AWS_ENDPOINT_URL_APPFABRIC
Appflow	aj AWS_ENDPOINT_URL_APPFLOW
AppIntegrations	aj AWS_ENDPOINT_URL_APPINTEGRATIONS at
Application Auto Scaling	aj AWS_ENDPOINT_URL_APPLICATION_AUTO_SCALING oi ca
Application Insights	aj AWS_ENDPOINT_URL_APPLICATION_INSIGHTS oi ts
ApplicationCostProfiler	aj AWS_ENDPOINT_URL_APPLICATIONCOSTPROFILER oi f:
App Mesh	aj AWS_ENDPOINT_URL_APP_MESH

serviceId	Kategori	Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE>
AppRunner	ap	AWS_ENDPOINT_URL_APPRUNNER
AppStream	ap	AWS_ENDPOINT_URL_APPSTREAM
AppSync	ap	AWS_ENDPOINT_URL_APPSsync
ARC Zonal Shift	a:	AWS_ENDPOINT_URL_ARC_ZONAL_SHIFT
Artifact	a:	AWS_ENDPOINT_URL_ARTIFACT
Athena	at	AWS_ENDPOINT_URL_ATHENA
AuditManager	am	AWS_ENDPOINT_URL_AUDITMANAGER
Auto Scaling	as	AWS_ENDPOINT_URL_AUTO_SCALING
Auto Scaling Plans	as	AWS_ENDPOINT_URL_AUTO_SCALING_PLANS
b2bi	b:	AWS_ENDPOINT_URL_B2BI
Backup	b:	AWS_ENDPOINT_URL_BACKUP

serviceId	Ki Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE> pe la ur di Al Fi co
Backup Gateway	b: AWS_ENDPOINT_URL_BACKUP_GATEWAY te
BackupStorage	b: AWS_ENDPOINT_URL_BACKUPSTORAGE ra
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 ge
Bedrock Agent Runtime	b: AWS_ENDPOINT_URL_BEDROCK_AGENT_RUNTIME ge ir
Bedrock Runtime	b: AWS_ENDPOINT_URL_BEDROCK_RUNTIME ur
billingconductor	b: AWS_ENDPOINT_URL_BILLINGCONDUCTOR nc
Braket	b: AWS_ENDPOINT_URL_BRAKET
Budgets	b: AWS_ENDPOINT_URL_BUDGETS

serviceId	Ki Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE>
Cost Explorer	co AWS_ENDPOINT_URL_COST_EXPLORER o:
chatbot	cl AWS_ENDPOINT_URL_CHATBOT
Chime	cl AWS_ENDPOINT_URL_CHIME
Chime SDK Identity	cl AWS_ENDPOINT_URL_CHIME_SDK_IDENTITY _:
Chime SDK Media Pipelines	cl AWS_ENDPOINT_URL_CHIME_SDK_MEDIA_PIPELINES _r pe
Chime SDK Meetings	cl AWS_ENDPOINT_URL_CHIME_SDK_MEETINGS _r
Chime SDK Messaging	cl AWS_ENDPOINT_URL_CHIME_SDK_MESSAGING _r g
Chime SDK Voice	cl AWS_ENDPOINT_URL_CHIME_SDK_VOICE _r
CleanRooms	c: AWS_ENDPOINT_URL_CLEANROOMS s

serviceId	Ki Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE> pe la ur di Al Fi co
CleanRoomsML	c: AWS_ENDPOINT_URL_CLEANROOMSML sr
Cloud9	c: AWS_ENDPOINT_URL_CLOUD9
CloudControl	c: AWS_ENDPOINT_URL_CLOUDCONTROL r()
CloudDirectory	c: AWS_ENDPOINT_URL_CLOUDDIRECTORY cl
CloudFormation	c: AWS_ENDPOINT_URL_CLOUDFORMATION at
CloudFront	c: AWS_ENDPOINT_URL_CLOUDFRONT t
CloudFront Key Value Store	c: AWS_ENDPOINT_URL_CLOUDFRONT_KEYVALUESTORE t_ e:
CloudHSM	c: AWS_ENDPOINT_URL_CLOUDHSM
CloudHSM V2	c: AWS_ENDPOINT_URL_CLOUDHSM_V2 v:
CloudSearch	c: AWS_ENDPOINT_URL_CLOUDSEARCH cl

serviceId	Ki Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE> pe la ur di Al Fi co
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	co AWS_ENDPOINT_URL_CODEARTIFACT ar
CodeBuild	co AWS_ENDPOINT_URL_CODEBUILD
CodeCatalyst	co AWS_ENDPOINT_URL_CODECATALYST y:
CodeCommit	co AWS_ENDPOINT_URL_CODECOMMIT t
CodeDeploy	co AWS_ENDPOINT_URL_CODEDEPLOY y
CodeGuru Reviewer	co AWS_ENDPOINT_URL_CODEGURU_REVIEWER r:

serviceId	Ki Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE>
CodeGuru Security	co AWS_ENDPOINT_URL_CODEGURU_SECURITY se
CodeGuruProfiler	co AWS_ENDPOINT_URL_CODEGURUPROFILER ic
CodePipeline	co AWS_ENDPOINT_URL_CODEPIPELINE in
CodeStar	co AWS_ENDPOINT_URL_CODESTAR
CodeStar connections	co AWS_ENDPOINT_URL_CODESTAR_CONNECTIONS co ns
codestar notificat ions	co AWS_ENDPOINT_URL_CODESTAR_NOTIFICATIONS no ic
Cognito Identity	co AWS_ENDPOINT_URL_COGNITO_IDENTITY de
Cognito Identity Provider	co AWS_ENDPOINT_URL_COGNITO_IDENTITY_PROVIDER de ic
Cognito Sync	co AWS_ENDPOINT_URL_COGNITO_SYNC yl

serviceId	Ki Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE> pe la ur di Al Fi co
Comprehend	co AWS_ENDPOINT_URL_COMPREHEND d
ComprehendMedical	co AWS_ENDPOINT_URL_COMPREHENDMEDICAL dr
Compute Optimizer	co AWS_ENDPOINT_URL_COMPUTE_OPTIMIZER pt
Config Service	co AWS_ENDPOINT_URL_CONFIG_SERVICE r\
Connect	co AWS_ENDPOINT_URL_CONNECT
Connect Contact Lens	co AWS_ENDPOINT_URL_CONNECT_CONTACT_LENS or n:
ConnectCampaigns	co AWS_ENDPOINT_URL_CONNECTCAMPAIGNS mj
ConnectCases	co AWS_ENDPOINT_URL_CONNECTCASES s:
ConnectParticipant	co AWS_ENDPOINT_URL_CONNECTPARTICIPANT r\
ControlTower	co AWS_ENDPOINT_URL_CONTROLTOWER w:

serviceId	Kategori	Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE>
Cost Optimization Hub	cost	AWS_ENDPOINT_URL_COST_OPTIMIZATION_HUB
Cost and Usage Report Service	cost	AWS_ENDPOINT_URL_COST_AND_USAGE_REPORT_SERVICE
Customer Profiles	customer	AWS_ENDPOINT_URL_CUSTOMER_PROFILES
DataBrew	data	AWS_ENDPOINT_URL_DATABREW
DataExchange	data	AWS_ENDPOINT_URL_DATAEXCHANGE
Data Pipeline	data	AWS_ENDPOINT_URL_DATA_PIPELINE
DataSync	data	AWS_ENDPOINT_URL_DATASYNC
DataZone	data	AWS_ENDPOINT_URL_DATAZONE
DAX	data	AWS_ENDPOINT_URL_DAX
Detective	data	AWS_ENDPOINT_URL_DETECTIVE

serviceId	Ki Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE> pe la ur di Al Fi co
Device Farm	d: AWS_ENDPOINT_URL_DEVICE_FARM IT
DevOps Guru	d: AWS_ENDPOINT_URL_DEVOPS_GURU IT
Direct Connect	d: AWS_ENDPOINT_URL_DIRECT_CONNECT ni
Application Discovery Service	a: AWS_ENDPOINT_URL_APPLICATION_DISCOVER o: Y_SERVICE e: co
DLM	d: AWS_ENDPOINT_URL_DLM
Database Migration Service	d: AWS_ENDPOINT_URL_DATABASE_MIGRATION_ m: SERVICE _:
DocDB	d: AWS_ENDPOINT_URL_DOCDB
DocDB Elastic	d: AWS_ENDPOINT_URL_DOCDB_ELASTIC st
drs	d: AWS_ENDPOINT_URL_DRS
Directory Service	d: AWS_ENDPOINT_URL_DIRECTORY_SERVICE _:

serviceId	Ki	Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE>
DynamoDB	dy	AWS_ENDPOINT_URL_DYNAMODB
DynamoDB Streams	dy	AWS_ENDPOINT_URL_DYNAMODB_STREAMS
EBS	el	AWS_ENDPOINT_URL_EBS
EC2	ec	AWS_ENDPOINT_URL_EC2
EC2 Instance Connect	ec	AWS_ENDPOINT_URL_EC2_INSTANCE_CONNECT
ECR	ec	AWS_ENDPOINT_URL_ECR
ECR PUBLIC	ec	AWS_ENDPOINT_URL_ECR_PUBLIC
ECS	ec	AWS_ENDPOINT_URL_ECS
EFS	ef	AWS_ENDPOINT_URL_EFS
EKS	el	AWS_ENDPOINT_URL_EKS
EKS Auth	el	AWS_ENDPOINT_URL_EKS_AUTH
Elastic Inference	ei	AWS_ENDPOINT_URL_ELASTIC_INFERENCE
ElastiCache	el	AWS_ENDPOINT_URL_ELASTICACHE

serviceId	Ki Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE> pe la ur di Al Fi co
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:
EntityResolution	er AWS_ENDPOINT_URL_ENTITYRESOLUTION o:
Elasticsearch Service	e: AWS_ENDPOINT_URL_ELASTICSEARCH_SERVICE a: i:

serviceId	Ki Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE> pe la ur di Al Fi co
EventBridge	ev AWS_ENDPOINT_URL_EVENTBRIDGE ge
Evidently	ev AWS_ENDPOINT_URL_EVIDENTLY
finspace	f: AWS_ENDPOINT_URL_FINSPLACE
finspace data	f: AWS_ENDPOINT_URL_FINSPLACE_DATA da
Firehose	f: AWS_ENDPOINT_URL_FIREHOSE
fis	f: AWS_ENDPOINT_URL_FIS
FMS	fr AWS_ENDPOINT_URL_FMS
forecast	fo AWS_ENDPOINT_URL_FORECAST
forecastquery	fo AWS_ENDPOINT_URL_FORECASTQUERY ur
FraudDetector	f: AWS_ENDPOINT_URL_FRAUDETECTOR ct
FreeTier	f: AWS_ENDPOINT_URL_FREETIER
FSx	f: AWS_ENDPOINT_URL_FSX
GameLift	g: AWS_ENDPOINT_URL_GAMELIFT

serviceId	Kategori	Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE>
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
Greengrass	g:	AWS_ENDPOINT_URL_GREENGRASS
GreengrassV2	g:	AWS_ENDPOINT_URL_GREENGRASSV2
GroundStation	g:	AWS_ENDPOINT_URL_GROUNDSTATION
GuardDuty	g:	AWS_ENDPOINT_URL_GUARDDUTY
Health	h:	AWS_ENDPOINT_URL_HEALTH
HealthLake	h:	AWS_ENDPOINT_URL_HEALTHLAKE
Honeycode	h:	AWS_ENDPOINT_URL_HONEYCODE

serviceId	Ki Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE> pe la ur di Al Fi co
IAM	i: AWS_ENDPOINT_URL_IAM
identitystore	i: AWS_ENDPOINT_URL_IDENTITYSTORE t:
imagebuilder	i: AWS_ENDPOINT_URL_IMAGEBUILDER d:
ImportExport	i: AWS_ENDPOINT_URL_IMPORTEXPORT o:
Inspector	i: AWS_ENDPOINT_URL_INSPECTOR
Inspector Scan	i: AWS_ENDPOINT_URL_INSPECTOR_SCAN _:
Inspector2	i: AWS_ENDPOINT_URL_INSPECTOR2 2
InternetMonitor	i: AWS_ENDPOINT_URL_INTERNETMONITOR o:
IoT	i: AWS_ENDPOINT_URL_IOT
IoT Data Plane	i: AWS_ENDPOINT_URL_IOT_DATA_PLANE p:

serviceId	Keterangan
	Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE>
IoT Jobs Data Plane	<code>AWS_ENDPOINT_URL_IOT_JOBS_DATA_PLANE</code>
IoT 1Click Devices Service	<code>AWS_ENDPOINT_URL_IOT_1CLICK_DEVICES_SERVICE</code>
IoT 1Click Projects	<code>AWS_ENDPOINT_URL_IOT_1CLICK_PROJECTS</code>
IoTAnalytics	<code>AWS_ENDPOINT_URL_IOTANALYTICS</code>
IotDeviceAdvisor	<code>AWS_ENDPOINT_URL_IOTDEVICEADVISOR</code>
IoT Events	<code>AWS_ENDPOINT_URL_IOT_EVENTS</code>
IoT Events Data	<code>AWS_ENDPOINT_URL_IOT_EVENTS_DATA</code>
IoTFleetHub	<code>AWS_ENDPOINT_URL_IOTFLEETHUB</code>
IoTFleetWise	<code>AWS_ENDPOINT_URL_IOTFLEETWISE</code>

serviceId	Kategori	Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE>
IoTSecureTunneling	iot	AWS_ENDPOINT_URL_IOTSECURETUNNELING
IoTSiteWise	iot	AWS_ENDPOINT_URL_IOTSITWISE
IoTThingsGraph	iot	AWS_ENDPOINT_URL_IOTTHINGSGRAPH
IoTTwinMaker	iot	AWS_ENDPOINT_URL_IOTTWINMAKER
IoT Wireless	iot	AWS_ENDPOINT_URL_IOT_WIRELESS
ivs	ivs	AWS_ENDPOINT_URL_IVS
IVS RealTime	ivs	AWS_ENDPOINT_URL_IVS_REALTIME
ivschat	ivs	AWS_ENDPOINT_URL_IVSCHAT
Kafka	kafka	AWS_ENDPOINT_URL_KAFKA
KafkaConnect	kafka	AWS_ENDPOINT_URL_KAFKACONNECT
kendra	kendra	AWS_ENDPOINT_URL_KENDRA

serviceId	Ki Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE> pe la ur di Al Fi co
Kendra Ranking	ke AWS_ENDPOINT_URL_KENDRA_RANKING nl
Keyspaces	ke AWS_ENDPOINT_URL_KEYSPACES
Kinesis	k: AWS_ENDPOINT_URL_KINESIS
Kinesis Video Archived Media	k: AWS_ENDPOINT_URL_KINESIS_VIDEO_ARCHI ic VED_MEDIA iv a
Kinesis Video Media	k: AWS_ENDPOINT_URL_KINESIS_VIDEO_MEDIA ic a
Kinesis Video Signaling	k: AWS_ENDPOINT_URL_KINESIS_VIDEO_SIGNALING ic a:
Kinesis Video WebRTC Storage	k: AWS_ENDPOINT_URL_KINESIS_VIDEO_WEBRT ic C_STORAGE to e
Kinesis Analytics	k: AWS_ENDPOINT_URL_KINESIS_ANALYTICS na

serviceId	Ki Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE> pe la ur di Al Fi co
Kinesis Analytics V2	k: AWS_ENDPOINT_URL_KINESIS_ANALYTICS_V2 na vi
Kinesis Video	k: AWS_ENDPOINT_URL_KINESIS_VIDEO ic
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 zi
Lex Model Building Service	l: AWS_ENDPOINT_URL_LEX_MODEL_BUILDING_ _I SERVICE _s
Lex Runtime Service	l: AWS_ENDPOINT_URL_LEX_RUNTIME_SERVICE me e
Lex Models V2	l: AWS_ENDPOINT_URL_LEX_MODELS_V2 S_
Lex Runtime V2	l: AWS_ENDPOINT_URL_LEX_RUNTIME_V2 me

serviceId	Ki Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE> pe la ur di Al Fi co
License Manager	l: AWS_ENDPOINT_URL_LICENSE_MANAGER a:
License Manager Linux Subscriptions	l: AWS_ENDPOINT_URL_LICENSE_MANAGER_LIN a: UX_SUBSCRIPTIONS n: r:
License Manager User Subscriptions	l: AWS_ENDPOINT_URL_LICENSE_MANAGER_USE a: R_SUBSCRIPTIONS e: i:
Lightsail	l: AWS_ENDPOINT_URL_LIGHTSAIL
Location	l: AWS_ENDPOINT_URL_LOCATION
CloudWatch Logs	c: AWS_ENDPOINT_URL_CLOUDWATCH_LOGS h:
CloudWatch Logs	c: AWS_ENDPOINT_URL_CLOUDWATCH_LOGS h:
LookoutEquipment	l: AWS_ENDPOINT_URL_LOOKOUTEQUIPMENT u:
LookoutMetrics	l: AWS_ENDPOINT_URL_LOOKOUTMETRICS t:

serviceId	Kategori	Contoh
	Kategori	Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE>
LookoutVision	LookoutVision	AWS_ENDPOINT_URL_LOOKOUTVISION
m2	Machine Learning	AWS_ENDPOINT_URL_M2
Machine Learning	Machine Learning	AWS_ENDPOINT_URL_MACHINE_LEARNING
Macie2	Macie2	AWS_ENDPOINT_URL_MACIE2
ManagedBlockchain	ManagedBlockchain	AWS_ENDPOINT_URL_MANAGEDBLOCKCHAIN
ManagedBlockchain Query	ManagedBlockchain Query	AWS_ENDPOINT_URL_MANAGEDBLOCKCHAIN_QUERY
Marketplace Agreement	Marketplace Agreement	AWS_ENDPOINT_URL_MARKETPLACE_AGREEMENT
Marketplace Catalog	Marketplace Catalog	AWS_ENDPOINT_URL_MARKETPLACE_CATALOG
Marketplace Deployment	Marketplace Deployment	AWS_ENDPOINT_URL_MARKETPLACE_DEPLOYMENT

serviceId	Ki Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE> pe la ur di Al Fi co
Marketplace Entitlement Service	m: AWS_ENDPOINT_URL_MARKETPLACE_ENTITLE c: MENT_SERVICE er v:
Marketplace Commerce Analytics	m: AWS_ENDPOINT_URL_MARKETPLACE_COMMERC c: E_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:
MediaPackage Vod	m: AWS_ENDPOINT_URL_MEDIAPACKAGE_VOD a:
MediaPackageV2	m: AWS_ENDPOINT_URL_MEDIAPACKAGEV2 a:

serviceId	Ki Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE> pe la ur di Al Fi co
MediaStore	me AWS_ENDPOINT_URL_MEDIASTORE e
MediaStore Data	me AWS_ENDPOINT_URL_MEDIASTORE_DATA e_
MediaTailor	me AWS_ENDPOINT_URL_MEDIATAILOR o:
Medical Imaging	me AWS_ENDPOINT_URL_MEDICAL_IMAGING mi
MemoryDB	me AWS_ENDPOINT_URL_MEMORYDB
Marketplace Metering	mi AWS_ENDPOINT_URL_MARKETPLACE_METERING ce ng
Migration Hub	m: AWS_ENDPOINT_URL_MIGRATION_HUB _f
mgn	m: AWS_ENDPOINT_URL_MGN
Migration Hub Refactor Spaces	m: AWS_ENDPOINT_URL_MIGRATION_HUB_REFACTOR_SPACES ct e:

serviceId	Kategori	Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE>
MigrationHub Config	migrasi	AWS_ENDPOINT_URL_MIGRATIONHUB_CONFIG
MigrationHubOrchestrator	migrasi	AWS_ENDPOINT_URL_MIGRATIONHUBORCHESTRATOR
MigrationHubStrategy	migrasi	AWS_ENDPOINT_URL_MIGRATIONHUBSTRATEGY
Mobile	migrasi	AWS_ENDPOINT_URL_MOBILE
mq	migrasi	AWS_ENDPOINT_URL_MQ
MTurk	migrasi	AWS_ENDPOINT_URL_MTURK
MWAA	migrasi	AWS_ENDPOINT_URL_MWAA
Neptune	neptunus	AWS_ENDPOINT_URL_NEPTUNE
Neptune Graph	neptunus	AWS_ENDPOINT_URL_NEPTUNE_GRAPH
neptunedata	neptunus	AWS_ENDPOINT_URL_NEPTUNEDATA
Network Firewall	neptunus	AWS_ENDPOINT_URL_NETWORK_FIREWALL

serviceId	Ki Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE> pe la ur di Al Fi co
NetworkManager	ne AWS_ENDPOINT_URL_NETWORKMANAGER n:
NetworkMonitor	ne AWS_ENDPOINT_URL_NETWORKMONITOR n:
nimble	n: AWS_ENDPOINT_URL_NIMBLE
OAM	o: AWS_ENDPOINT_URL_OAM
Omics	or AWS_ENDPOINT_URL_OMICS
OpenSearch	o: AWS_ENDPOINT_URL_OPENSEARCH h
OpenSearchServerless	o: AWS_ENDPOINT_URL_OPENSEARCHSERVERLESS h: s:
OpsWorks	o: AWS_ENDPOINT_URL_OPSWORKS
OpsWorksCM	o: AWS_ENDPOINT_URL_OPSWORKSCM m
Organizations	o: AWS_ENDPOINT_URL_ORGANIZATIONS ic
OSIS	o: AWS_ENDPOINT_URL_OSIS
Outposts	o: AWS_ENDPOINT_URL_OUTPOSTS

serviceId	Ki Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE> pe la ur di Al Fi co
p8data	p: AWS_ENDPOINT_URL_P8DATA
p8data	p: AWS_ENDPOINT_URL_P8DATA
Panorama	p: AWS_ENDPOINT_URL_PANORAMA
Payment Cryptography	p: AWS_ENDPOINT_URL_PAYMENT_CRYPTOGRAPHY ry hy
Payment Cryptography Data	p: AWS_ENDPOINT_URL_PAYMENT_CRYPTOGRAPHY_DATA ry hy
Pca Connector Ad	p: AWS_ENDPOINT_URL_PCA_CONNECTOR_AD ct
Personalize	p: AWS_ENDPOINT_URL_PERSONALIZE ze
Personalize Events	p: AWS_ENDPOINT_URL_PERSONALIZE_EVENTS ze
Personalize Runtime	p: AWS_ENDPOINT_URL_PERSONALIZE_RUNTIME ze e
PI	p: AWS_ENDPOINT_URL_PI

serviceId	Kategori	Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE>
Pinpoint	p	AWS_ENDPOINT_URL_PINPOINT
Pinpoint Email	p	AWS_ENDPOINT_URL_PINPOINT_EMAIL
Pinpoint SMS Voice	p	AWS_ENDPOINT_URL_PINPOINT_SMS_VOICE
Pinpoint SMS Voice V2	p	AWS_ENDPOINT_URL_PINPOINT_SMS_VOICE_V2
Pipes	p	AWS_ENDPOINT_URL_PIPES
Polly	p	AWS_ENDPOINT_URL_POLLY
Pricing	p	AWS_ENDPOINT_URL_PRICING
PrivateNetworks	p	AWS_ENDPOINT_URL_PRIVATENETWORKS
Proton	p	AWS_ENDPOINT_URL_PROTON
QBusiness	q	AWS_ENDPOINT_URL_QBUSINESS
QConnect	q	AWS_ENDPOINT_URL_QCONNECT
QLDB	q	AWS_ENDPOINT_URL_QLDB

serviceId	Ki Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE> pe la ur di Al Fi co
QLDB Session	q AWS_ENDPOINT_URL_QLDB_SESSION ic
QuickSight	q AWS_ENDPOINT_URL_QUICKSIGHT t
RAM	r AWS_ENDPOINT_URL_RAM
rbin	r AWS_ENDPOINT_URL_RBIN
RDS	r AWS_ENDPOINT_URL_RDS
RDS Data	r AWS_ENDPOINT_URL_RDS_DATA
Redshift	r AWS_ENDPOINT_URL_REDSHIFT
Redshift Data	r AWS_ENDPOINT_URL_REDSHIFT_DATA di
Redshift Serverless	r AWS_ENDPOINT_URL_REDSHIFT_SERVERLESS S S
Rekognition	r AWS_ENDPOINT_URL_REKOGNITION O
repostspace	r AWS_ENDPOINT_URL_REPOSTSPACE C

serviceId	Kategori	Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE>
resiliencehub	Region	AWS_ENDPOINT_URL_RESILIENCEHUB
Resource Explorer 2	Region	AWS_ENDPOINT_URL_RESOURCE_EXPLORER_2
Resource Groups	Region	AWS_ENDPOINT_URL_RESOURCE_GROUPS
Resource Groups Tagging API	Region	AWS_ENDPOINT_URL_RESOURCE_GROUPS_TAGGING_API
RoboMaker	Region	AWS_ENDPOINT_URL_ROBOMAKER
RolesAnywhere	Region	AWS_ENDPOINT_URL_ROLESEANYWHERE
Route 53	Region	AWS_ENDPOINT_URL_ROUTE_53
Route53 Recovery Cluster	Region	AWS_ENDPOINT_URL_ROUTE53_RECOVERY_CLUSTER

serviceId	Kategori	Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE>
Route53 Recovery Control Config	recovery	AWS_ENDPOINT_URL_ROUTE53_RECOVERY_CONTROL_CONFIG
Route53 Recovery Readiness	recovery	AWS_ENDPOINT_URL_ROUTE53_RECOVERY_READINESS
Route 53 Domains	domains	AWS_ENDPOINT_URL_ROUTE_53_DOMAINS
Route53Resolver	resolver	AWS_ENDPOINT_URL_ROUTE53RESOLVER
RUM	rum	AWS_ENDPOINT_URL_RUM
S3	s3	AWS_ENDPOINT_URL_S3
S3 Control	s3	AWS_ENDPOINT_URL_S3_CONTROL
S3Outposts	s3	AWS_ENDPOINT_URL_S3OUTPOSTS
SageMaker	sagemaker	AWS_ENDPOINT_URL_SAGEMAKER

serviceId	Ki Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE> pe la ur di Al Fi co
SageMaker A2I Runtime	s: AWS_ENDPOINT_URL_SAGEMAKER_A2I_RUNTIME _i ir
Sagemaker Edge	s: AWS_ENDPOINT_URL_SAGEMAKER_EDGE _f
SageMaker FeatureStore Runtime	s: AWS_ENDPOINT_URL_SAGEMAKER_FEATURESTORE_RUNTIME _t ir
SageMaker Geospatial	s: AWS_ENDPOINT_URL_SAGEMAKER_GEOSPATIAL _f a:
SageMaker Metrics	s: AWS_ENDPOINT_URL_SAGEMAKER_METRICS _f
SageMaker Runtime	s: AWS_ENDPOINT_URL_SAGEMAKER_RUNTIME _f
savingsplans	s: AWS_ENDPOINT_URL_SAVINGSPLANS a:
Scheduler	s: AWS_ENDPOINT_URL_SCHEDULER
schemas	s: AWS_ENDPOINT_URL_SCHEMAS

serviceId	Ki Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE> pe la ur di Al Fi co
SimpleDB	s: AWS_ENDPOINT_URL_SIMPLEDB
Secrets Manager	s: AWS_ENDPOINT_URL_SECRETS_MANAGER a:
SecurityHub	s: AWS_ENDPOINT_URL_SECURITYHUB ul
SecurityLake	s: AWS_ENDPOINT_URL_SECURITYLAKE al
ServerlessApplicat ionRepository	s: AWS_ENDPOINT_URL_SERVERLESSAPPLICATI s: ONREPOSITORY ic tc
Service Quotas	s: AWS_ENDPOINT_URL_SERVICE_QUOTAS uc
Service Catalog	s: AWS_ENDPOINT_URL_SERVICE_CATALOG a:
Service Catalog AppRegistry	s: AWS_ENDPOINT_URL_SERVICE_CATALOG_APP a: REGISTRY p:
ServiceDiscovery	s: AWS_ENDPOINT_URL_SERVICEDISCOVERY s:

serviceId	Kategori	Detail
	Kategori	Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE>
SES	Service	AWS_ENDPOINT_URL_SES
SESV2	Service	AWS_ENDPOINT_URL_SESV2
Shield	Service	AWS_ENDPOINT_URL_SHIELD
signer	Service	AWS_ENDPOINT_URL_SIGNER
SimSpaceWeaver	Service	AWS_ENDPOINT_URL_SIMSPACEWEAVER
SMS	Service	AWS_ENDPOINT_URL_SMS
Snow Device Management	Service	AWS_ENDPOINT_URL_SNOW_DEVICE_MANAGEMENT
Snowball	Service	AWS_ENDPOINT_URL_SNOWBALL
SNS	Service	AWS_ENDPOINT_URL_SNS
SQS	Service	AWS_ENDPOINT_URL_SQS
SSM	Service	AWS_ENDPOINT_URL_SSM
SSM Contacts	Service	AWS_ENDPOINT_URL_SSM_CONTACTS
SSM Incidents	Service	AWS_ENDPOINT_URL_SSM_INCIDENTS

serviceId	Kategori	Detail
	Kategori	Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE>
Ssm Sap	Parameter	s: AWS_ENDPOINT_URL_SSM_SAP
SSO	Parameter	s: AWS_ENDPOINT_URL_SSO
SSO Admin	Parameter	s: AWS_ENDPOINT_URL_SSO_ADMIN
SSO OIDC	Parameter	s: AWS_ENDPOINT_URL_SSO_OIDC
SFN	Parameter	s: AWS_ENDPOINT_URL_SFN
Storage Gateway	Parameter	s: AWS_ENDPOINT_URL_STORAGE_GATEWAY
STS	Parameter	s: AWS_ENDPOINT_URL_STS
SupplyChain	Parameter	s: AWS_ENDPOINT_URL_SUPPLYCHAIN
Support	Parameter	s: AWS_ENDPOINT_URL_SUPPORT
Support App	Parameter	s: AWS_ENDPOINT_URL_SUPPORT_APP
SWF	Parameter	s: AWS_ENDPOINT_URL_SWF
synthetics	Parameter	s: AWS_ENDPOINT_URL_SYNTHETICS
Textract	Parameter	t: AWS_ENDPOINT_URL_TEXTRACT

serviceId	Ki Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE> pe la ur di Al Fi co
Timestream InfluxDB	t: AWS_ENDPOINT_URL_TIMESTREAM_INFLUXDB m_ b
Timestream Query	t: AWS_ENDPOINT_URL_TIMESTREAM_QUERY m_
Timestream Write	t: AWS_ENDPOINT_URL_TIMESTREAM_WRITE m_
tnb	tr AWS_ENDPOINT_URL_TNB
Transcribe	t: AWS_ENDPOINT_URL_TRANSCRIBE e
Transfer	t: AWS_ENDPOINT_URL_TRANSFER
Translate	t: AWS_ENDPOINT_URL_TRANSLATE
TrustedAdvisor	t: AWS_ENDPOINT_URL_TRUSTEDADVISOR v:
VerifiedPermissions	vr AWS_ENDPOINT_URL_VERIFIEDPERMISSIONS e: s
Voice ID	vr AWS_ENDPOINT_URL_VOICE_ID

serviceId	Ki	Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE>
VPC Lattice	vpc-lattice	AWS_ENDPOINT_URL_VPC_LATTICE
WAF	waf	AWS_ENDPOINT_URL_WAF
WAF Regional	waf-regional	AWS_ENDPOINT_URL_WAF_REGIONAL
WAFV2	wafv2	AWS_ENDPOINT_URL_WAFV2
WellArchitected	well-architected	AWS_ENDPOINT_URL_WELLARCHITECTED
Wisdom	wisdom	AWS_ENDPOINT_URL_WISDOM
WorkDocs	workdocs	AWS_ENDPOINT_URL_WORKDOCS
WorkLink	worklink	AWS_ENDPOINT_URL_WORKLINK
WorkMail	workmail	AWS_ENDPOINT_URL_WORKMAIL
WorkMailMessageFlow	workmail-messageflow	AWS_ENDPOINT_URL_WORKMAILMESSAGEFLOW
WorkSpaces	workspaces	AWS_ENDPOINT_URL_WORKSPACES

serviceId	Ki Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE> pe la ur di Al Fi co
WorkSpaces Thin Client	wc AWS_ENDPOINT_URL_WORKSPACES_THIN_CLIENT s_ it
WorkSpaces Web	wc AWS_ENDPOINT_URL_WORKSPACES_WEB s_
XRay	x: AWS_ENDPOINT_URL_XRAY

Default konfigurasi cerdas

Dengan fitur default konfigurasi cerdas, AWS SDKs dapat memberikan nilai default yang telah ditentukan dan dioptimalkan untuk pengaturan konfigurasi lainnya.

Konfigurasi fungsi ini dengan menggunakan yang berikut:

defaults_mode- dibagikan AWS **config** pengaturan file, **AWS_DEFAULTS_MODE**- variabel lingkungan, **aws.defaultsMode**- properti JVM sistem: Hanya Java/Kotlin

Dengan pengaturan ini, Anda dapat memilih mode yang selaras dengan arsitektur aplikasi Anda, yang kemudian memberikan nilai default yang dioptimalkan untuk aplikasi Anda. Jika AWS SDK pengaturan memiliki nilai yang ditetapkan secara eksplisit, maka nilai itu selalu diutamakan. Jika AWS SDK pengaturan tidak memiliki nilai yang ditetapkan secara eksplisit, dan `defaults_mode` tidak sama dengan warisan, maka fitur ini dapat memberikan nilai default yang berbeda untuk berbagai pengaturan yang dioptimalkan untuk aplikasi Anda. Pengaturan dapat mencakup hal berikut: pengaturan HTTP komunikasi, perilaku coba lagi, pengaturan titik akhir Regional layanan, dan, berpotensi, konfigurasi SDK terkait apa pun. Pelanggan yang menggunakan fitur ini bisa mendapatkan default konfigurasi baru yang disesuaikan dengan skenario penggunaan umum. Jika Anda `defaults_mode` tidak sama dengan `legacy`, sebaiknya

lakukan pengujian aplikasi saat Anda memutakhirkan SDK, karena nilai default yang diberikan mungkin berubah seiring dengan perkembangan praktik terbaik.

Nilai default: `legacy`

Catatan: Versi utama baru SDKs akan default ke `standard`.

Nilai yang valid:

- `legacy`— Menyediakan pengaturan default yang bervariasi menurut SDK dan ada sebelum pembentuk `defaults_mode`.
- `standard`— Memberikan nilai default terbaru yang direkomendasikan yang harus aman untuk dijalankan di sebagian besar skenario.
- `in-region`— Dibangun pada mode standar dan termasuk pengoptimalan yang disesuaikan untuk aplikasi yang memanggil Layanan AWS dari dalam yang sama Wilayah AWS.
- `cross-region`— Dibangun pada mode standar dan termasuk pengoptimalan yang disesuaikan untuk aplikasi yang memanggil Layanan AWS di daerah yang berbeda.
- `mobile`— Dibangun pada mode standar dan termasuk pengoptimalan yang disesuaikan untuk aplikasi seluler.
- `auto`— Dibangun pada mode standar dan termasuk fitur eksperimental. SDK upaya untuk menemukan lingkungan runtime untuk menentukan pengaturan yang sesuai secara otomatis. Deteksi otomatis berbasis heuristik dan tidak memberikan akurasi 100%. Jika lingkungan runtime tidak dapat ditentukan, `standard` mode digunakan. Deteksi otomatis mungkin menanyakan [metadata instance](#), yang mungkin memperkenalkan latensi. Jika latensi startup sangat penting untuk aplikasi Anda, sebaiknya pilih yang eksplisit `defaults_mode`.

Contoh pengaturan nilai ini dalam `config` file:

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

Parameter berikut dapat dioptimalkan berdasarkan pemilihan `defaults_mode`:

- `retryMode`— Menentukan bagaimana SDK upaya mencoba lagi. Lihat [Coba lagi perilaku](#).
- `stsRegionalEndpoints`— Menentukan bagaimana SDK menentukan Layanan AWS titik akhir yang digunakannya untuk berbicara dengan AWS Security Token Service (AWS STS). Lihat [AWS STS Titik akhir regional](#).

- `s3UsEast1RegionalEndpoints`- Menentukan bagaimana SDK menentukan AWS titik akhir layanan yang digunakannya untuk berbicara dengan Amazon S3 untuk `us-east-1` Wilayah tersebut.
- `connectTimeoutInMillis`— Setelah melakukan upaya koneksi awal pada soket, jumlah waktu sebelum waktu habis. Jika klien tidak menerima penyelesaian jabat tangan terhubung, klien menyerah dan gagal dalam operasi.
- `tlsNegotiationTimeoutInMillis`Jumlah maksimum waktu yang dapat diambil oleh TLS jabat tangan dari saat CLIENT HELLO pesan dikirim ke waktu klien dan server telah sepenuhnya menegosiasikan sandi dan bertukar kunci.

Nilai default untuk setiap pengaturan berubah tergantung pada yang `defaults_mode` dipilih untuk aplikasi Anda. Nilai-nilai ini saat ini ditetapkan sebagai berikut (dapat berubah sewaktu-waktu):

Parameter	standard modus	in-region modus	cross-region modus	mobile modus
<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

Misalnya, jika yang `defaults_mode` Anda pilih adalah `standard`, maka nilai `standard` akan ditetapkan untuk `retry_mode` (dari `retry_mode` opsi yang valid) dan nilai `regional` akan ditetapkan untuk `stsRegionalEndpoints` (dari `stsRegionalEndpoints` opsi yang valid).

Kompatibilitas dengan AWS SDKs

Berikut ini SDKs mendukung fitur dan pengaturan yang dijelaskan dalam topik ini. Setiap pengecualian sebagian dicatat. Pengaturan properti JVM sistem apa pun didukung oleh AWS SDK for Java dan AWS SDK for Kotlin hanya.

SDK	Didukung	Catatan atau informasi lebih lanjut
AWS CLI v2	Tidak	
SDK untuk C++	Ya	Parameter tidak dioptimalkan: <code>stsRegionalEndpoints</code> , <code>s3UsEast1RegionalEndpoints</code> , <code>tlsNegotiationTimeoutInMillis</code> .
SDK untuk Go V2 (1.x)	Ya	Parameter tidak dioptimalkan: <code>retryMode</code> , <code>stsRegionalEndpoints</code> , <code>s3UsEast1RegionalEndpoints</code> .
SDK untuk Go 1.x (V1)	Tidak	
SDK untuk Java 2.x	Ya	Parameter tidak dioptimalkan: <code>stsRegionalEndpoints</code> .
SDK untuk Java 1.x	Tidak	
SDK untuk JavaScript 3.x	Ya	Parameter tidak dioptimalkan: <code>stsRegionalEndpoints</code> , <code>s3UsEast1RegionalEndpoints</code> , <code>tlsNegoti</code>

SDK	Didukung	Catatan atau informasi lebih lanjut
		<pre> ationTime outInMillis . connectTimeoutInMi llis disebutconnectio nTimeout . </pre>
SDK untuk JavaScript 2.x	Tidak	
SDK untuk Kotlin	Tidak	
SDK untuk .NET 3.x	Ya	Parameter tidak dioptimalkan: <code>connectTimeoutInMillis</code> , <code>tlsNegotiationTimeoutInMillis</code> .
SDK untuk PHP 3.x	Ya	Parameter tidak dioptimalkan: <code>tlsNegotiationTimeoutInMillis</code> .
SDK untuk Python (Boto3)	Ya	Parameter tidak dioptimalkan: <code>tlsNegotiationTimeoutInMillis</code> .
SDK untuk Ruby 3.x	Ya	
SDK untuk Rust	Tidak	
SDK untuk Swift	Tidak	
Alat untuk PowerShell	Ya	Parameter tidak dioptimalkan: <code>connectTimeoutInMillis</code> , <code>tlsNegotiationTimeoutInMillis</code> .

AWS Pustaka Runtime (CRT) umum

Pustaka AWS Common Runtime (CRT) adalah pustaka dasar dari SDKs CRT. Ini adalah keluarga modular paket independen, ditulis dalam C. Setiap paket memberikan kinerja yang baik dan jejak minimal untuk berbagai fungsi yang diperlukan. Fungsionalitas ini umum dan dibagikan di semua SDKs menyediakan penggunaan kembali kode, pengoptimalan, dan akurasi yang lebih baik. Paket-paketnya adalah:

- [awslabs/aws-c-auth](#): otentikasi AWS sisi klien (penyedia kredensi standar dan penandatanganan (sigv4))
- [awslabs/aws-c-cal](#): Jenis primitif kriptografi, hash (MD5, SHA256, HMAC, SHA256 penandatanganan, AES)
- [awslabs/aws-c-common](#): Struktur data dasar, tipe primitif utas/sinkronisasi, manajemen buffer, fungsi terkait stdlib
- [awslabs/aws-c-compression](#): Algoritma kompresi (pengkodean/decoding Huffman)
- [awslabs/aws-c-event-stream](#): Pemrosesan pesan aliran acara (header, pendahuluan, payload, crc/trailer), implementasi panggilan prosedur jarak jauh () melalui aliran acara RPC
- [awslabs/aws-c-http](#): Implementasi C99 dari spesifikasi HTTP /1.1 dan HTTP /2
- [awslabs/aws-c-io](#): Soket (TCP,UDP), DNS, pipa, loop acara, saluran, SSL/TLS
- [awslabs/aws-c-iot](#): Implementasi C99 integrasi layanan cloud AWS IoT dengan perangkat
- [awslabs/aws-c-mqtt](#): Protokol pesan standar dan ringan untuk Internet of Things (IoT)
- [awslabs/aws-c-s3](#): Implementasi pustaka C99 untuk berkomunikasi dengan layanan Amazon S3, yang dirancang untuk memaksimalkan throughput pada instans Amazon dengan bandwidth tinggi EC2
- [awslabs/aws-c-sdkutils](#): Pustaka utilitas untuk mengurai dan mengelola profil AWS
- [awslabs/aws-checksums](#): Perangkat keras lintas platform dipercepat CRC32c dan CRC32 dengan mundur ke implementasi perangkat lunak yang efisien
- [awslabs/aws-1c](#): Perpustakaan kriptografi tujuan umum yang dikelola oleh tim AWS Cryptography untuk AWS dan pelanggan mereka, berdasarkan kode dari proyek Google Boring dan SSL proyek Open SSL
- [awslabs/s2n](#): Implementasi C99 dari SSL protokol TLS /, dirancang agar kecil dan cepat dengan keamanan sebagai prioritas

Tersedia melalui semua SDKs kecuali Go dan Rust. CRT

CRT dependensi

CRT Perustakaan membentuk jaringan hubungan dan dependensi yang kompleks. Mengetahui hubungan ini sangat membantu jika Anda perlu membangun CRT langsung dari sumber. Namun, sebagian besar pengguna mengakses CRT fungsionalitas melalui bahasa mereka SDK (seperti AWS SDK untuk C++ atau AWS SDK untuk Java) atau SDK perangkat IoT bahasa mereka (seperti IoT untuk C++ atau AWS SDK AWS IoT untuk Java). SDK Dalam diagram berikut, kotak Language CRT Bindings mengacu pada paket yang membungkus CRT pustaka untuk bahasa tertentu. SDK Ini adalah kumpulan paket `aws-crt-*`, di mana `*` adalah SDK bahasa (seperti [aws-crt-cpp](#) atau [aws-crt-java](#)).

Berikut ini adalah ilustrasi dependensi hierarkis perustakaan. CRT

AWS Kebijakan pemeliharaan SDK dan Alat

Gambaran Umum

Dokumen ini menguraikan kebijakan pemeliharaan untuk AWS Perangkat Pengembangan Perangkat Lunak (SDK) dan Alat, termasuk SDK Seluler dan IoT, dan dependensi yang mendasarinya. AWS secara teratur menyediakan AWS SDK dan Tools dengan pembaruan yang mungkin berisi dukungan untuk AWS API baru atau yang diperbarui, fitur baru, penyempurnaan, perbaikan bug, tambalan keamanan, atau pembaruan dokumentasi. Pembaruan juga dapat mengatasi perubahan dengan dependensi, runtime bahasa, dan sistem operasi. AWS Rilis SDK dipublikasikan ke manajer paket (misalnya Maven, NuGet PyPI), dan tersedia sebagai kode sumber di GitHub.

Kami menyarankan pengguna untuk tetap up-to-date menggunakan rilis SDK untuk mengikuti fitur terbaru, pembaruan keamanan, dan dependensi yang mendasarinya. Penggunaan berkelanjutan dari versi SDK yang tidak didukung tidak disarankan dan dilakukan atas kebijaksanaan pengguna.

Penentuan Versi

Versi rilis AWS SDK dalam bentuk X.Y.Z di mana X mewakili versi utama. Peningkatan versi utama SDK menunjukkan bahwa SDK ini mengalami perubahan signifikan dan substansif untuk mendukung idiom dan pola baru dalam bahasa tersebut. Versi utama diperkenalkan ketika antarmuka publik (misalnya kelas, metode, jenis, dll.), Perilaku, atau semantik telah berubah. Aplikasi perlu diperbarui agar dapat bekerja dengan versi SDK terbaru. Penting untuk memperbarui versi utama dengan hati-hati dan sesuai dengan pedoman peningkatan yang disediakan oleh AWS.

Siklus hidup versi utama SDK

Siklus hidup untuk versi SDK dan Tools utama terdiri dari 5 fase, yang diuraikan di bawah ini.

- Pratinjau Pengembang (Fase 0) - Selama fase ini, SDK tidak didukung, tidak boleh digunakan di lingkungan produksi, dan dimaksudkan untuk akses awal dan tujuan umpan balik saja. Dimungkinkan untuk rilis future untuk memperkenalkan perubahan yang melanggar. Setelah AWS mengidentifikasi rilis untuk menjadi produk yang stabil, itu dapat menandainya sebagai Kandidat Rilis. Kandidat Rilis siap untuk rilis GA kecuali bug signifikan muncul, dan akan menerima AWS dukungan penuh.

- **Ketersediaan Umum (GA) (Fase 1)** - Selama fase ini, SDK didukung penuh. AWS akan menyediakan rilis SDK reguler yang mencakup dukungan untuk layanan baru, pembaruan API untuk layanan yang ada, serta perbaikan bug dan keamanan. Untuk Tools, AWS akan menyediakan rilis reguler yang mencakup pembaruan fitur baru dan perbaikan bug. AWS akan mendukung SDK versi GA setidaknya selama 24 bulan.
- **Pengumuman Pemeliharaan (Tahap 2)** - AWS akan membuat pengumuman publik setidaknya 6 bulan sebelum SDK memasuki mode pemeliharaan. Selama periode ini, SDK akan terus didukung sepenuhnya. Biasanya, mode pemeliharaan diumumkan bersamaan dengan versi utama berikutnya dialihkan ke GA.
- **Pemeliharaan (Fase 3)** - Selama mode pemeliharaan, AWS membatasi rilis SDK untuk mengatasi perbaikan bug kritis dan masalah keamanan saja. SDK tidak akan menerima pembaruan API untuk layanan baru atau yang sudah ada, atau diperbarui untuk mendukung wilayah baru. Mode pemeliharaan memiliki durasi default 12 bulan, kecuali ditentukan lain.
- **End-of-Support (Fase 4)** - Ketika SDK mencapai akhir dukungan, SDK tidak akan lagi menerima pembaruan atau rilis. Rilis yang diterbitkan sebelumnya akan terus tersedia melalui manajer paket publik dan kode akan tetap aktif GitHub. GitHub Repositori dapat diarsipkan. Penggunaan SDK yang telah mencapai end-of-support dilakukan atas kebijaksanaan pengguna. Kami menyarankan pengguna meningkatkan ke versi utama yang baru.

Berikut ini adalah ilustrasi visual dari siklus hidup versi utama SDK. Harap dicatat bahwa garis waktu yang ditunjukkan di bawah ini adalah ilustrasi dan tidak mengikat.

Siklus hidup ketergantungan

Sebagian besar AWS SDK memiliki dependensi yang mendasarinya, seperti runtime bahasa, sistem operasi, atau pustaka dan kerangka kerja pihak ketiga. Dependensi ini biasanya terkait dengan komunitas bahasa atau vendor yang memiliki komponen tertentu. Setiap komunitas atau vendor menerbitkan end-of-support jadwal mereka sendiri untuk produk mereka.

Istilah berikut digunakan untuk mengklasifikasikan dependensi pihak ketiga yang mendasarinya:

- **Sistem Operasi (OS):** Contohnya termasuk Amazon Linux AMI, Amazon Linux 2, Windows 2008, Windows 2012, Windows 2016, dll.
- **Runtime Bahasa:** Contohnya termasuk Java 7, Java 8, Java 11, .NET Core, .NET Standard, .NET PCL, dll.

- Library/Framework pihak ketiga: Contohnya termasuk OpenSSL, .NET Framework 4.5, Java EE, dll.

Kebijakan kami adalah untuk terus mendukung dependensi SDK setidaknya selama 6 bulan setelah komunitas atau vendor mengakhiri dukungan untuk dependensi. Kebijakan ini, bagaimanapun, dapat bervariasi tergantung pada ketergantungan spesifik.

Note

AWS berhak untuk menghentikan dukungan untuk ketergantungan yang mendasarinya tanpa meningkatkan versi SDK utama

Metode komunikasi

Pengumuman pemeliharaan dikomunikasikan dalam beberapa cara:

- Pengumuman email dikirim ke akun yang terpengaruh, mengumumkan rencana kami untuk mengakhiri dukungan untuk versi SDK tertentu. Email akan menguraikan jalur ke end-of-support, menentukan jadwal kampanye, dan memberikan panduan peningkatan.
- AWS Dokumentasi SDK, seperti dokumentasi referensi API, panduan pengguna, halaman pemasaran produk SDK, dan GitHub readme diperbarui untuk menunjukkan garis waktu kampanye dan memberikan panduan tentang peningkatan aplikasi yang terpengaruh.
- Sebuah posting AWS blog diterbitkan yang menguraikan jalan ke end-of-support, serta mengulangi jadwal kampanye.
- Peringatan penghentian ditambahkan ke SDK, menguraikan jalur ke end-of-support dan menautkan ke dokumentasi SDK.

Untuk melihat daftar versi utama AWS SDK dan Alat yang tersedia serta di mana mereka berada dalam siklus hidup pemeliharaannya, lihat. [Dukungan versi](#)

AWS SDKsdan dukungan versi Tools

Tabel di bawah ini menunjukkan daftar yang tersedia AWS Software Development Kit (SDK) versi utama dan di mana mereka berada dalam siklus hidup pemeliharaan dengan jadwal terkait. Untuk informasi rinci tentang siklus hidup untuk versi utama AWS SDKsdan Alat dan dependensi yang mendasarinya, lihat. [Kebijakan pemeliharaan](#)

SDK	Versi mayor	Fase Saat Ini	Tanggal Ketersediaan Umum	Catatan
AWS CLI	1.x	Ketersediaan Umum	9/2/2013	
AWS CLI	2.x	Ketersediaan Umum	2/10/2020	
SDKuntuk C ++	1.x	Ketersediaan Umum	9/2/2015	
SDKuntuk Go V2	V2 1.x	Ketersediaan Umum	1/19/2021	
SDKuntuk Go	1.x	Maintenance	11/19/2015	Lihat pengumuman untuk detail dan tanggal
SDKuntuk Java	1.x	Maintenance	25/03/2010	Lihat pengumuman untuk detail dan tanggal
SDKuntuk Java	2.x	Ketersediaan Umum	11/20/2018	
SDKuntuk JavaScript	1.x	Akhir dukungan	5/6/2013	

SDK	Versi mayor	Fase Saat Ini	Tanggal Ketersediaan Umum	Catatan
SDK untuk JavaScript	2.x	Maintenance	6/19/2014	Lihat pengumuman untuk detail dan tanggal
SDK untuk JavaScript	3.x	Ketersediaan Umum	12/15/2020	
SDK untuk Kotlin	1.x	Ketersediaan Umum	11/27/2023	
SDK untuk .NET	1.x	Akhir dukungan	11/2009	
SDK untuk .NET	2.x	Akhir dukungan	11/8/2013	
SDK untuk .NET	3.x	Ketersediaan Umum	7/28/2015	
SDK untuk PHP	2.x	Akhir dukungan	11/2/2012	
SDK untuk PHP	3.x	Ketersediaan Umum	5/27/2015	
SDK untuk Python (Boto2)	1.x	Akhir dukungan	7/13/2011	
SDK untuk Python (Boto3)	1.x	Ketersediaan Umum	6/22/2015	
SDK untuk Python (Botocore)	1.x	Ketersediaan Umum	6/22/2015	
SDK untuk Ruby	1.x	Akhir dukungan	7/14/2011	
SDK untuk Ruby	2.x	Akhir dukungan	2/15/2015	

SDK	Versi mayor	Fase Saat Ini	Tanggal Ketersediaan Umum	Catatan
SDK untuk Ruby	3.x	Ketersediaan Umum	29/8/2017	
SDK untuk Rust	1.x	Ketersediaan Umum	11/27/2023	
SDK untuk Swift	1.x	Ketersediaan Umum	9/17/2024	
Alat untuk PowerShell	2.x	Akhir dukungan	11/8/2013	
Alat untuk PowerShell	3.x	Akhir dukungan	29/7/2015	
Alat untuk PowerShell	4.x	Ketersediaan Umum	11/21/2019	

Mencari alat SDK atau yang tidak disebutkan? Enkripsi SDKs, Perangkat IoT SDKs, dan Seluler SDKs, misalnya, tidak termasuk dalam panduan ini. Untuk menemukan dokumentasi tentang alat lain ini, lihat [Alat untuk Dibangun AWS](#).

Riwayat dokumen untuk AWS SDKsdan Panduan Referensi Alat

Tabel berikut menjelaskan penambahan penting dan pembaruan ke AWS SDKsdan Panduan Referensi Alat. Untuk pemberitahuan tentang pembaruan dokumentasi ini, Anda dapat berlangganan RSS umpan.

Perubahan	Deskripsi	Tanggal
Menambahkan referensi Swift SDK ke Pengaturan	Menambahkan SDK dukungan Swift ke semua referensi Pengaturan Kompatibilitas dengan AWS SDKstabel.	September 17, 2024
SDKuntuk properti sistem Java 1.x	Tambahkan detail tentang pengaturan konfigurasi JVM sistem yang didukung oleh AWS SDK for Java 1.x.	30 Mei 2024
Pembaruan pengaturan	Tambahkan pengaturan konfigurasi JVM sistem.	Maret 27, 2024
Pembaruan tabel kompatibilitas	Pembaruan kompatibilitas untuk SDK dukungan, pembaruan prosedur Pusat IAM Identitas.	Februari 20, 2024
Pembaruan kredensi kontainer . IMDSmemperbarui.	Menambahkan dukungan untuk AmazonEKS. Menambahkan pengaturan untuk menonaktifkan IMDSv1 fallback.	Desember 29, 2023
Minta kompresi	Menambahkan pengaturan untuk fitur kompresi permintaan.	27 Desember 2023

Tabel kompatibilitas	Tabel kompatibilitas untuk SDK dan fitur alat yang diperbarui untuk disertakan SDK untuk Kotlin, SDK untuk Rust, dan AWS Tools for PowerShell.	Desember 10, 2023
Pembaruan otentikasi	Pembaruan untuk metode otentikasi yang didukung untuk SDKs dan alat.	Juli 1, 2023
IAM pembaruan praktik terbaik	Panduan yang diperbarui untuk menyelaraskan dengan praktik IAM terbaik. Untuk informasi selengkapnya, lihat Praktik terbaik keamanan di IAM .	27 Februari 2023
SSO pembaruan	Pembaruan SSO kredensi untuk konfigurasi SSO token baru.	November 19, 2022
Pembaruan pengaturan	Pembaruan untuk mendukung tabel untuk konfigurasi Umum dan untuk Amazon S3 Multi-Region Access Points.	17 November 2022
Pembaruan pengaturan	Pembaruan untuk kejelasan IMDS klien dan IMDS kredensial. Pembaruan untuk variabel Lingkungan.	4 November 2022
Memperbarui halaman selamat datang	Mengumumkan Amazon CodeWhisperer.	September 22, 2022

Perubahan nama layanan untuk sistem masuk tunggal	Pembaruan untuk mencerminkan hal itu AWS SSO. Sekarang disebut sebagai AWS IAM Identity Center.	26 Juli 2022
Pembaruan pengaturan	Pembaruan kecil untuk detail file konfigurasi dan pengaturan yang didukung.	15 Juni 2022
Perbarui	Pembaruan besar-besaran dari hampir semua bagian panduan ini.	1 Februari 2022
Rilis awal	Rilis pertama dari panduan ini dirilis ke publik.	13 Maret 2020

AWSGlosarium

Untuk AWS terminologi terbaru, lihat [AWSglosarium di Referensi](#). Glosarium AWS

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.