



Panduan Referensi

AWS SDK dan Alat



AWS SDK dan 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 menghina atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan properti dari masing-masing pemilik, yang mungkin berafiliasi, terkait dengan, atau disponsori oleh Amazon, atau tidak.

Table of Contents

AWS Panduan Referensi SDK dan 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	10
Resolusi direktori rumah	11
Ubah lokasi default file-file ini	11
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	17
ID AWS Builder	19
Autentikasi Pusat Identitas IAM	19
Konfigurasi akses terprogram menggunakan IAM Identity Center	20
Memahami autentikasi Pusat Identitas IAM	23
Peran IAM Di Mana Saja	27
Langkah 1: Konfigurasi Peran IAM Di Mana Saja	27
Langkah 2: Gunakan Peran IAM Di Mana Saja	27
Asumsikan peran	29
Asumsikan peran IAM	29
Bersekutu dengan identitas web atau OpenID Connect	30
AWS kunci akses	32
Gunakan kredensial jangka pendek	32
Gunakan kredensi jangka panjang	32
Kredensi jangka pendek	34
Kredensi jangka panjang	35
Peran IAM untuk instans Amazon EC2	38
Membuat peran IAM	39

Luncurkan instans Amazon EC2 dan tentukan peran IAM Anda	39
Connect ke instans EC2	40
Jalankan aplikasi sampel pada instans EC2	40
Referensi pengaturan	41
Membuat klien layanan	41
Prioritas pengaturan	41
Configdaftar pengaturan file	42
Credentialsdaftar pengaturan file	46
Daftar variabel lingkungan	46
Daftar properti sistem JVM	50
Penyedia kredensi standar	53
Rantai penyedia kredensi	54
AWS kunci akses	55
Asumsikan penyedia peran	58
Penyedia kontainer	64
Penyedia Pusat Identitas IAM	68
Penyedia IMDS	74
Penyedia proses	78
Fitur standar	82
ID Aplikasi	83
Metadata instans Amazon EC2	85
Titik akses Amazon S3	87
Titik Akses Multi-Wilayah Amazon S3	89
Wilayah AWS	91
AWS STS Titik akhir regional	94
Dual-stack dan titik akhir FIPS	97
Penemuan titik akhir	99
Konfigurasi umum	101
Klien IMDS	104
Coba lagi perilaku	107
Minta kompresi	113
Titik akhir khusus layanan	115
Default konfigurasi cerdas	158
Runtime Umum	164
Ketergantungan CRT	165
Kebijakan pemeliharaan	166

Gambaran Umum	166
Penentuan Versi	166
Siklus hidup versi utama SDK	166
Siklus hidup ketergantungan	167
Metode komunikasi	168
Matriks dukungan versi	169
Riwayat dokumen	172
AWSGlosarium	175
.....	clxxvi

AWS Panduan Referensi SDK dan Alat

Banyak SDK 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 mengonfigurasi AWS SDK dan alat Anda.
- [Otentikasi dan akses](#)— Tetapkan bagaimana kode atau alat Anda mengautentikasi AWS saat Anda mengembangkan. Layanan AWS
- [Referensi pengaturan](#)— Referensi untuk semua pengaturan standar yang tersedia untuk otentikasi dan konfigurasi.
- [AWSPustaka Runtime Umum \(CRT\)](#)— Ikhtisar pustaka AWS Common Runtime (CRT) bersama yang tersedia untuk hampir semua SDK.
- [AWS Kebijakan pemeliharaan SDK dan Alat](#) mencakup kebijakan pemeliharaan dan pembuatan versi untuk AWS Perangkat Pengembangan Perangkat Lunak (SDK) dan alat, termasuk SDK Seluler dan Internet of Things (IoT), dan dependensi yang mendasarinya.

Panduan Referensi AWS SDK dan Alat ini dimaksudkan untuk menjadi basis informasi yang berlaku untuk beberapa SDK dan alat. Panduan khusus untuk SDK atau alat yang Anda gunakan harus digunakan selain informasi apa pun 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"> • SDK atau alat apa pun 	AWS Kebijakan pemeliharaan SDK dan Alat
<ul style="list-style-type: none"> • AWS Cloud Development Kit (AWS CDK) Panduan Pengembang • AWS Serverless Application Model Panduan Pengembang • AWS Toolkit for Eclipse Panduan Pengguna 	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 Toolkit for JetBrains Panduan Pengguna • AWS Toolkit for Visual Studio Panduan Pengguna • AWS Toolkit for Visual Studio Code Panduan Pengguna 	
<ul style="list-style-type: none"> • AWS Command Line Interface Panduan Pengguna • AWS SDK for C++ Panduan Pengembang • AWS SDK for Go Panduan Pengembang • AWS SDK for Java Panduan Pengembang • AWS SDK for JavaScript Panduan Pengembang • AWS SDK for Kotlin • AWS SDK for .NET Panduan Pengembang • AWS SDK for PHP Panduan Pengembang • AWS SDK untuk Python (Boto3) Memulai • AWS SDK for Ruby Panduan Pengembang • AWS SDK for Rust • AWS SDK for Swift • AWS Tools for Windows PowerShell Panduan Pengguna 	<ul style="list-style-type: none"> • Konfigurasi • Otentikasi dan akses • Referensi pengaturan • AWSPustaka Runtime Umum (CRT) • AWS Kebijakan pemeliharaan SDK dan Alat • AWS Matriks dukungan versi SDK dan 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 dari IDE Anda. Untuk informasi rinci tentang masing-masing Toolkit IDE, lihat Panduan Pengguna Toolkit di tabel sebelumnya.

AWS IDE Toolkit dapat mengumpulkan dan menyimpan data telemetri sisi klien untuk menginformasikan keputusan mengenai rilis Toolkit masa depan. AWS Data yang dikumpulkan mengukur penggunaan Toolkit Anda. AWS

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

Untuk informasi rinci tentang data telemetri yang dikumpulkan oleh masing-masing Toolkit AWS IDE, rujuk dokumen sumber daya di repositori Github Toolkit berikut: AWS

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

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), AWS SDK, dan alat lainnya. Nilai konfigurasi dilampirkan ke profil untuk mengonfigurasi beberapa aspek SDK/alat saat profil itu digunakan. File-file ini “dibagikan” karena nilai mempengaruhi aplikasi, proses, atau SDK apa pun di lingkungan lokal untuk pengguna.

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

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 yang digunakan oleh SDK atau operasi 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 SDK atau alat Anda saat menjalankan kode Anda. Profil juga dapat dipilih dalam kode, atau per-perintah saat bekerja dengan file. AWS CLI

Konfigurasi fungsi ini dengan mengatur salah satu dari berikut ini:

AWS_PROFILE- variabel lingkungan

Ketika variabel lingkungan ini diatur ke profil bernama atau "default", semua kode SDK 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 sistem JVM

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

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 `bernamafoo`. 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 kredensi yang digunakan. AWS IAM Identity Center Untuk informasi selengkapnya tentang mengonfigurasi autentikasi masuk tunggal, lihat [Autentikasi Pusat Identitas IAM](#) 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 kredensi jangka pendek untuk peran IAM "SampleRole" di akun "111122223333" 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

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

`services` bagian ini selanjutnya dipisahkan menjadi subbagian dengan `<SERVICE> = baris`, di mana `<SERVICE>` adalah kunci Layanan AWS pengenalan. Layanan AWS Pengenal didasarkan pada model API `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"` `services` bagian ini direferensikan di `profile` bagian dengan nama menggunakan `services` kunci. Kunci Layanan AWS pengenalan adalah `dynamodb`. Subbagian Amazon DynamoDB layanan dimulai pada `telepondynamodb =` . Baris yang segera mengikuti yang menjorok termasuk dalam ayat itu dan berlaku untuk layanan itu.

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

Untuk informasi selengkapnya tentang konfigurasi titik akhir kustom, lihat [Titik akhir khusus layanan](#).

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:

- `config` variabel lingkungan file: **AWS_CONFIG_FILE**
- `credentials` variabel 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 SDK, 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...zrkuWJ0gQs8IZZaIv2BXIa2R40L"
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.

Di Kerangka Tanpa Server, Anda sering dapat mengatur variabel lingkungan SDK 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

- [Autentikasi Pusat Identitas IAM](#)— 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 SAMP 2.0 (iDP), atau secara individual menggabungkan IDP Anda ke Akun AWS. Untuk memeriksa apakah Wilayah Anda mendukung Pusat Identitas IAM, lihat [AWS IAM Identity Center titik akhir dan kuota](#) di Referensi Umum Amazon Web
- [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. Untuk menggunakan Peran IAM Di Mana Saja, beban kerja Anda harus menggunakan sertifikat X.509.
- [Asumsikan peran](#)— Anda dapat mengambil peran IAM 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

- [Menggunakan peran IAM untuk instans Amazon EC2](#)— Gunakan peran IAM untuk menjalankan aplikasi Anda dengan aman di instans Amazon EC2.
- Anda dapat berinteraksi secara terprogram dengan AWS menggunakan IAM Identity Center dengan cara berikut:
 - Gunakan [AWS CloudShell](#) untuk menjalankan AWS CLI perintah dari konsol.
 - Gunakan [AWS Cloud9](#) untuk memulai pemrograman AWS menggunakan lingkungan pengembangan terintegrasi (IDE) dengan AWS sumber daya.

- [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 kredensial 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 yang kompatibel dengan OpenID Connect (OIDC) lainnya. Mereka dapat menerima token otentikasi, dan kemudian menukar token itu dengan kredensial keamanan sementara di peta AWS itu ke peran IAM dengan izin untuk menggunakan sumber daya di Anda. Akun AWS

Untuk mempelajari cara mengonfigurasi ini untuk SDK atau alat Anda, lihat [Bersekitar 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 Panduan Pengguna IAM.

Informasi lebih lanjut tentang manajemen akses

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

- [Identitas IAM \(pengguna, grup pengguna, dan peran\)](#) - Memahami dasar-dasar identitas di AWS
- [Praktik terbaik keamanan di 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 BuilderMasuk 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 fileID AWS Builder.
- [CodeCatalystkonsep - ID AWS Builder](#) di Panduan CodeCatalyst Pengguna Amazon - Pelajari cara CodeCatalyst menggunakan fileID AWS Builder.

Autentikasi Pusat Identitas IAM

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 (Amazon EC2) AWS Cloud9 atau, sebaiknya Anda mendapatkan kredensi dari layanan tersebut.

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

- Portal AWS akses adalah lokasi web tempat Anda masuk secara manual ke Pusat Identitas IAM. 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/alat.
- AWS CLI Ini digunakan untuk mengonfigurasi SDK atau alat Anda untuk menggunakan autentikasi IAM Identity Center untuk panggilan API yang dibuat oleh kode Anda. Proses satu kali ini memperbarui AWS config file bersama Anda, yang kemudian digunakan oleh SDK atau alat saat Anda menjalankan kode.

Konfigurasi akses terprogram menggunakan IAM Identity Center

Langkah 1: Tetapkan akses dan pilih set izin yang sesuai

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

Pilih salah satu metode berikut untuk mengakses AWS kredensial Anda.

Saya tidak memiliki akses melalui IAM Identity Center

1. Tambahkan pengguna dan tambahkan izin administratif dengan mengikuti [Konfigurasi akses pengguna dengan prosedur direktori Pusat Identitas IAM default](#) di AWS IAM Identity Center Panduan 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 Identitas IAM 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/ SDK Anda.

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 SDK dan alat untuk menggunakan IAM Identity Center

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 URL ini atau Anda mungkin mendapatkannya dalam email berikut Langkah 1: Tetapkan akses. Jika tidak, temukan URL portal AWS akses Anda 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 Identitas IAM untuk mendapatkan SSO Region nilai SSO Start URL dan nilai yang Anda butuhkan untuk langkah berikutnya.
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](#).
- Untuk nama profil CLI, kami sarankan 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 IAM Identity Center yang Anda konfigurasi.
5. Jika Anda menggunakan AWS SDK, buat aplikasi untuk SDK Anda di lingkungan pengembangan Anda.
- a. Untuk beberapa SDK, paket tambahan seperti SSO dan SSOIDC harus ditambahkan ke aplikasi Anda sebelum Anda dapat menggunakan autentikasi IAM Identity Center. Untuk detailnya, lihat SDK spesifik Anda.
 - 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 SDK atau alat akan menggunakan kredensial Pusat Identitas IAM karena diutamakan. [Rantai penyedia kredensi](#)

Untuk mempelajari lebih dalam cara SDK dan alat 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 SDK atau alat 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 IAM Identity Center 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 :

- Durasi sesi IAM Identity Center — [Konfigurasi durasi sesi portal AWS akses pengguna Anda](#)
- Izin mengatur durasi sesi - [Mengatur durasi sesi](#)

Untuk detail tentang semua pengaturan penyedia Pusat Identitas IAM untuk SDK dan alat, lihat [Penyedia kredensi Pusat Identitas IAM](#) 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 di AWS IAM Identity Center Panduan Pengguna.
Konsol Pusat Identitas IAM Konsol administratif		Konsol yang Anda gunakan untuk mengonfigurasi sistem masuk tunggal.

Nama Pusat Identitas IAM	Nama API SDK	Deskripsi
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.”
Penyedia kredensial Pusat Identitas IAM	Penyedia kredensial 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 kredensial 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 AWSKredensial 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. [Konfigurasi penyedia token SSO](#)

- 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 kredensial kedaluwarsa. Jika kredensialnya tidak ditemukan atau tidak valid, maka SDK akan memanggil API Pusat Identitas IAM untuk mendapatkan kredensial 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: Konfigurasikan 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 SDKAWS 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 kredensial proses, lihat [Penyedia kredensial proses](#) di panduan ini.

Asumsikan 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 mempelajari lebih lanjut tentang AWS Security Token Service (AWS STS) permintaan API, lihat [Tindakan](#) di Referensi AWS Security Token Service API.

Untuk menyiapkan SDK atau alat untuk mengambil peran, Anda harus terlebih dahulu membuat atau mengidentifikasi peran tertentu yang akan diambil. [Peran IAM diidentifikasi secara unik oleh peran Amazon Resource Name \(ARN\)](#). Peran membangun hubungan kepercayaan dengan entitas lain. Entitas tepercaya yang menggunakan peran tersebut mungkin Layanan AWS, yang lain Akun AWS, penyedia identitas web atau OIDC, atau federasi SAMP. Untuk mempelajari selengkapnya tentang peran IAM, lihat [Menggunakan peran IAM](#) di Panduan Pengguna IAM.

Setelah peran IAM diidentifikasi, jika Anda dipercaya oleh peran tersebut, Anda dapat mengonfigurasi SDK atau alat untuk menggunakan izin yang diberikan oleh peran tersebut. Untuk melakukan ini, salah satu [Asumsikan peran IAM](#) atau [Bersekutu dengan identitas web atau OpenID Connect](#).

Asumsikan peran IAM

Saat mengasumsikan peran, AWS STS mengembalikan satu set kredensial keamanan sementara. Kredensi ini bersumber dari profil lain atau dari instance atau wadah tempat kode Anda berjalan. Contoh lain dari asumsi peran termasuk mengelola beberapa Akun AWS dari Amazon EC2, AWS CodeCommit menggunakan Akun AWS lintas, atau mengakses akun lain dari. AWS CodeBuild

Langkah 1: Siapkan peran IAM

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

Langkah 2: Konfigurasikan SDK atau alat

Konfigurasikan SDK atau alat untuk mendapatkan kredensial dari atau. `credential_source`
`source_profile`

Gunakan `credential_source` untuk mendapatkan kredensial dari wadah Amazon ECS, instans Amazon EC2, 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, SDK atau alat secara otomatis membuat panggilan AWS STS [AssumeRole](#) API 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 peran IAM 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.

Bersekutu dengan identitas web atau OpenID Connect

Saat membuat aplikasi seluler atau aplikasi web berbasis klien yang memerlukan akses ke AWS, AWS STS mengembalikan satu set kredensi keamanan sementara untuk pengguna federasi yang diautentikasi melalui penyedia identitas publik (iDP). Contoh penyedia identitas umum mencakup

Login with Amazon, Facebook, Google, atau penyedia identitas mana pun yang kompatibel dengan OpenID Connect (OIDC). Dengan metode ini, pengguna Anda tidak memerlukan identitas mereka sendiri AWS atau IAM.

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

Langkah 1: Siapkan penyedia identitas dan peran IAM

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

Langkah 2: Konfigurasi SDK atau alat

Konfigurasi SDK atau alat untuk menggunakan token identitas web AWS STS untuk otentikasi.

Saat Anda menentukan ini di profil, SDK atau alat secara otomatis membuat panggilan AWS STS [AssumeRoleWithWebIdentity](#) API 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 peran IAM 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 SDK dan pustaka inti alat seperti penyedia identitas lainnya. Untuk mengakses Amazon Cognito API, sertakan klien layanan Amazon Cognito di build atau pustaka untuk SDK atau alat Anda. Untuk penggunaan dengan AWS SDK, 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 [Autentikasi Pusat Identitas IAM](#) 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 SDK atau alat Anda untuk digunakan [Autentikasi Pusat Identitas IAM](#) dengan opsi durasi sesi yang diperpanjang. Namun, Anda dapat menyalin dan menggunakan kredensi sementara yang tersedia di portal AWS akses. Kredensi baru perlu disalin ketika ini kedaluwarsa. Anda dapat menggunakan kredensi sementara di profil atau menggunakannya sebagai nilai untuk properti sistem dan variabel lingkungan.

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>
```

3. Simpan file tersebut. File sekarang `~/.aws/credentials` harus ada di sistem pengembangan lokal Anda. File ini berisi [profil \[default\]](#) yang digunakan SDK atau alat jika profil bernama tertentu tidak ditentukan.
4. [Masuk ke portal AWS akses](#).
5. Ikuti petunjuk ini untuk [penyegaran kredensial manual](#) untuk menyalin kredensi peran IAM dari portal akses. AWS
 - a. Untuk langkah 4 dalam petunjuk terkait, pilih nama peran IAM yang memberikan akses untuk kebutuhan pengembangan Anda. Peran ini biasanya memiliki nama seperti `PowerUserAccess` atau `Pengembang`.

- JANGAN menaruh 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.
- JANGAN sertakan file yang berisi kredensi 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 [peran IAM untuk tugas untuk tugas](#) Amazon Elastic Container Service (Amazon ECS).
- Gunakan [peran IAM](#) untuk aplikasi yang berjalan di instans Amazon EC2.

Prasyarat: Buat akun AWS

Untuk menggunakan pengguna IAM 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 Pengguna IAM.

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 pengguna IAM Anda

- Buat pengguna IAM Anda dengan mengikuti prosedur [Membuat pengguna IAM \(konsol\)](#) di Panduan Pengguna IAM. Saat membuat pengguna IAM 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 tutorial SDK “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 konsol IAM, 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 SDK.
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 pengguna IAM dan kunci akses rahasia Anda. 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.

`credentialsFile` 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 IAM Identity Center atau kredensi sementara lainnya sesegera mungkin. Setelah Anda beralih dari penggunaan kredensi jangka panjang, ingatlah untuk menghapus kredensi ini dari file bersama `credentials`.

Menggunakan peran IAM untuk instans Amazon EC2

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

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

Membuat peran IAM

Buat peran IAM yang memberikan akses hanya-baca ke Amazon S3.

1. Masuk ke AWS Management Console dan buka konsol IAM 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 meluncurkan instans Amazon EC2 Anda.

Luncurkan instans Amazon EC2 dan tentukan peran IAM Anda

Anda dapat meluncurkan instans Amazon EC2 dengan peran IAM menggunakan konsol Amazon EC2.

Ikuti petunjuk untuk meluncurkan instans di Panduan Pengguna [Amazon EC2](#) atau [Panduan Pengguna Amazon EC2](#).

Saat Anda mencapai halaman Peluncuran Instance Tinjauan, pilih Edit detail instans. Dalam peran IAM, pilih peran IAM yang Anda buat sebelumnya. Selesaikan prosedur sesuai petunjuk.

Note

Anda perlu membuat atau menggunakan grup keamanan dan key pair yang ada untuk terhubung ke instance.

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

Connect ke instans EC2

Connect ke instans EC2 sehingga Anda dapat mentransfer aplikasi sampel ke sana dan kemudian menjalankan aplikasi. Anda akan memerlukan file yang berisi bagian pribadi dari key pair yang Anda gunakan untuk meluncurkan instance; yaitu, file PEM.

Anda dapat melakukannya dengan mengikuti prosedur koneksi di Panduan Pengguna [Amazon EC2](#) atau [Panduan Pengguna Amazon EC2](#). Ketika Anda terhubung, lakukan sedemikian rupa sehingga Anda dapat mentransfer file dari mesin pengembangan Anda ke instans Anda.

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 sampel pada instans EC2

1. Salin file aplikasi dari drive lokal Anda ke instans Anda.

Untuk informasi tentang cara mentransfer file ke instans, lihat Panduan Pengguna [Amazon EC2](#) atau [Panduan Pengguna Amazon EC2](#).

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 konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.](https://console.aws.amazon.com/ec2/)
 - b. Pilih instance dan lepaskan peran IAM melalui Tindakan, Pengaturan Instans, Lampirkan/Ganti Peran IAM.
 - c. Jalankan aplikasi lagi dan konfirmasikan bahwa ia mengembalikan kesalahan otorisasi.

Referensi pengaturan

SDK menyediakan API khusus bahasa untuk. Layanan AWS Mereka menangani beberapa pekerjaan berat yang diperlukan untuk berhasil melakukan panggilan API, termasuk otentikasi, perilaku coba lagi, dan banyak lagi. Untuk melakukan ini, SDK memiliki strategi fleksibel untuk mendapatkan kredensial yang akan digunakan untuk permintaan Anda, untuk mempertahankan pengaturan yang akan 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 SDK dan Tools penyedia kredensi standar](#)— Penyedia kredensi umum distandarisasi di beberapa SDK.
- [AWS Fitur standar SDK dan Tools](#)— Fitur umum distandarisasi di beberapa SDK.

Membuat klien layanan

Untuk mengakses secara terprogram Layanan AWS, SDK menggunakan kelas/objek klien untuk masing-masing. Layanan AWS Misalnya, jika aplikasi Anda perlu mengakses Amazon EC2, aplikasi Anda akan membuat objek klien Amazon EC2 untuk berinteraksi dengan layanan tersebut. Anda kemudian menggunakan klien layanan untuk membuat permintaan untuk itu Layanan AWS. Di sebagian besar SDK, 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 lainnya yang didukung oleh sebagian besar SDK dan memiliki dampak luas. Layanan AWS Semua SDK 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, penetapan eksplisit dapat berupa parameter yang Anda tetapkan saat membuat instance Layanan AWS klien atau objek konfigurasi, atau terkadang saat Anda memanggil API individual.
2. Hanya Java/Kotlin: Properti sistem JVM 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 ada, SDK menggunakannya.
 - Variabel `AWS_PROFILE` lingkungan atau properti sistem `aws.profile` JVM dapat digunakan untuk menentukan profil mana yang dimuat SDK.
 6. Setiap nilai default yang disediakan oleh kode sumber SDK itu sendiri digunakan terakhir.

Note

Beberapa SDK dan alat mungkin memeriksa dalam urutan yang berbeda. Selain itu, beberapa SDK dan alat mendukung metode lain untuk menyimpan dan mengambil parameter. Misalnya, AWS SDK for .NET mendukung sumber tambahan yang disebut [SDK Store](#). Untuk informasi selengkapnya tentang penyedia yang unik untuk SDK atau alat, 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, profil tersebut hanya ditemukan dan digunakan setelah SDK atau alat 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 parameter API) atau dalam kode akan mengesampingkan semua nilai lain untuk satu perintah itu.

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 setelan dan variabel lingkungan yang hanya didukung oleh SDK atau alat individual, lihat SDK atau panduan alat tertentu.

Nama pengaturan	Detail
api_versions	Pengaturan konfigurasi umum
aws_access_key_id	AWS kunci akses
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	Penyedia kredensi IMDS
ec2_metadata_service	Penyedia kredensi IMDS

Nama pengaturan	Detail	
ce_endpoint_mode		
ec2_metadata_v1_disabled	Penyedia kredensi IMDS	
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_num_attempts	Metadana instans Amazon EC2	
metadata_service_timeout	Metadana instans Amazon EC2	
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	Penyedia kredensi Pusat Identitas IAM
sso_region	Penyedia kredensi Pusat Identitas IAM
sso_regis tration_scopes	Penyedia kredensi Pusat Identitas IAM
sso_role_name	Penyedia kredensi Pusat Identitas IAM
sso_start_url	Penyedia kredensi Pusat Identitas IAM
sts_regio nal_endpoints	AWS STS Titik akhir regional

Nama pengaturan	Detail
use_duals tack_endpoint	Dual-stack dan titik akhir FIPS
use_fips_ endpoint	Dual-stack 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. SDK dan alat juga dapat mendukung pengaturan unik dan variabel lingkungan. Untuk melihat setelan dan variabel lingkungan yang hanya didukung oleh SDK atau alat individual, lihat SDK atau panduan alat tertentu.

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 SDK tercantum dalam tabel berikut. Mereka global dan mempengaruhi semua Layanan AWS. SDK dan alat juga dapat mendukung pengaturan unik dan variabel lingkungan. Untuk melihat setelan dan variabel lingkungan yang hanya didukung oleh SDK atau alat individual, lihat SDK atau panduan alat tertentu.

Nama pengaturan	Detail	
AWS_ACCESS_KEY_ID	AWS kunci akses	
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	
AWS_DISABLE_REQUEST_COMPRESSION	Minta kompresi	

Nama pengaturan	Detail
AWS_EC2_METADATA_DISABLED	Penyedia kredensi IMDS
AWS_EC2_METADATA_SERVICE_ENDPOINT	Penyedia kredensi IMDS
AWS_EC2_METADATA_SERVICE_ENDPOINT_MODE	Penyedia kredensi IMDS
AWS_EC2_METADATA_V1_DISABLED	Penyedia kredensi IMDS
AWS_ENABLE_ENDPOINT_DISCOVERY	Penemuan titik akhir
AWS_ENDPOINT_URL	Titik akhir khusus layanan
AWS_ENDPOINT_URL_<SERVICE>	Titik akhir khusus layanan
AWS_IAM_ROLE_ARN	Asumsikan penyedia kredensi peran
AWS_IAM_ROLE_SESSION_NAME	Asumsikan penyedia kredensi peran

Nama pengaturan	Detail
AWS_IGNORE_CONFIGURED_ENDPOINT_URLS	Titik akhir khusus layanan
AWS_MAX_ATTEMPTS	Coba lagi perilaku
AWS_METADATA_SERVICES_NUM_ATTEMPTS	Metadata instans Amazon EC2
AWS_METADATA_SERVICE_TIMEOUT	Metadata instans Amazon EC2
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_S3_MULTIREGION_ACCESS_POINTS	Titik Akses Multi-Wilayah Amazon S3
AWS_S3_USE_ARN_REGION	Titik akses Amazon S3

Nama pengaturan	Detail
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	Dual-stack dan titik akhir FIPS
AWS_USE_F IPS_ENDPOINT	Dual-stack dan titik akhir FIPS
AWS_WEB_I DENTITY_T OKEN_FILE	Asumsikan penyedia kredensi peran

Daftar properti sistem JVM

Anda dapat menggunakan properti sistem JVM 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 sistem JVM.

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

Nama pengaturan	Detail
<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
<code>aws.sessionToken</code>	AWS kunci akses

Nama pengaturan	Detail
<code>aws.shareCredentialsFile</code>	Lokasi yang dibagikan config dan credentials file
<code>aws.useDualstackEndpoint</code>	Dual-stack dan titik akhir FIPS
<code>aws.useFipsEndpoint</code>	Dual-stack dan titik akhir FIPS
<code>aws.userAgentAppId</code>	ID Aplikasi
<code>aws.webIdentityTokenFile</code>	Asumsikan penyedia kredensi peran

AWS SDK dan Tools penyedia kredensi standar

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

Important

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

Topik

- [Rantai penyedia kredensi](#)
- [AWS kunci akses](#)
- [Asumsikan penyedia kredensi peran](#)
- [Penyedia kredensi kontainer](#)

- [Penyedia kredensi Pusat Identitas IAM](#)
- [Penyedia kredensi IMDS](#)
- [Penyedia kredensi proses](#)

Rantai penyedia kredensi

Semua SDK 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 kredensial default.

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

Penyedia kredensi	Deskripsi
AWS kunci akses	AWS kunci akses untuk pengguna IAM (seperti <code>AWS_ACCESS_KEY_ID</code> , dan <code>AWS_SECRET_ACCESS_KEY</code>).
Bersekutu dengan identitas web atau OpenID Connect - Asumsikan penyedia kredensial 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 peran IAM menggunakan token identitas web from AWS Security Token Service (AWS STS).
Penyedia kredensi Pusat Identitas IAM	Dapatkan kredensial dari. AWS IAM Identity Center
Asumsikan penyedia kredensial peran	Dapatkan akses ke sumber daya lain dengan mengasumsikan izin peran IAM. (Ambil dan kemudian gunakan kredensial sementara untuk peran).
Penyedia kredensial kontainer	Amazon Elastic Container Service (Amazon ECS) dan kredensial Amazon Elastic Kubernetes Service (Amazon

Penyedia kredensi	Deskripsi
	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 Roles Anywhere.
Penyedia kredensi IMDS	Kredensial profil instans Amazon Elastic Compute Cloud (Amazon EC2). Kaitkan peran IAM dengan setiap instans EC2 Anda. Kredensi sementara untuk peran itu tersedia untuk kode yang berjalan dalam instance. Kredensialnya dikirimkan melalui layanan metadata Amazon EC2.

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](#).

AWS kunci akses

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](#).

AWS kunci akses untuk pengguna IAM dapat digunakan sebagai AWS kredensial Anda. AWS SDK secara otomatis menggunakan AWS kredensial ini untuk menandatangani permintaan API 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 kredensi 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 `credentialsFile` bersama adalah lokasi yang disarankan untuk menyimpan informasi kredensial karena aman di luar direktori sumber aplikasi dan terpisah dari pengaturan khusus SDK dari file bersama. `config`

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

Konfigurasi fungsi ini dengan menggunakan yang berikut ini:

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 sistem JVM: 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 sistem JVM: 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 sistem JVM: 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 kredensi 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/bPxrFcIYEXAMPLEKEY
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/bPxrFcIYEXAMPLEKEY
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/bPxrFcIYEXAMPLEKEY
setx
  AWS_SESSION_TOKEN AQoEXAMPLEH4aoAH0gNCAPy...truncated...zrkuWJ0gQs8IZZaIv2BXIa2R40lgk
```

Kompatibilitas dengan AWS SDK

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

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

SDK	Di	Catatan atau informasi lebih lanjut
SDK for Java 2.x	Ya	
SDK for Java 1.x	Ya	
SDK untuk 3.x JavaScript	Ya	
SDK untuk 2.x JavaScript	Ya	
SDK para Kotlin	Ya	
SDK for .NET 3.x	Ya	Variabel lingkungan tidak didukung.
SDK for PHP 3.x	Ya	
SDK untuk Python (Boto3)	Ya	
SDK for Ruby 3.x	Ya	
SDK untuk Rust	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 menyiapkan SDK atau alat untuk mengambil peran, Anda harus terlebih dahulu membuat atau mengidentifikasi peran tertentu yang akan diambil. [Peran IAM diidentifikasi secara unik oleh peran Amazon Resource Name \(ARN\)](#). Peran membangun hubungan kepercayaan dengan entitas lain. Entitas tepercaya yang menggunakan peran tersebut mungkin Layanan AWS, yang lain Akun AWS, penyedia identitas web atau OIDC, atau federasi SAFL.

Setelah peran IAM 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](#) di panduan ini.

Asumsikan pengaturan penyedia kredensi peran

Konfigurasi fungsi ini dengan menggunakan yang berikut ini:

credential_source- Pengaturan AWS **config** file bersama

Digunakan dalam instans Amazon EC2 atau penampung Amazon Elastic Container Service untuk menentukan tempat SDK atau alat 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 menggunakan [peran IAM yang dilampirkan ke profil instans EC2 untuk mendapatkan](#) kredensial sumber.
- `EcsContainer`— Menentukan bahwa SDK atau alat adalah menggunakan [peran IAM yang dilampirkan ke wadah ECS 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 Pengguna IAM.

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 Pengguna IAM.

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 itu mencakup kondisi yang memerlukan otentikasi MFA.

Nilai default: Tidak ada.

Nilai yang valid: Nilai dapat berupa nomor seri untuk perangkat keras (seperti `GAHT12345678`), atau Nama Sumber Daya Amazon (ARN) untuk perangkat MFA virtual. Untuk informasi selengkapnya tentang MFA, lihat [Mengonfigurasi Akses API yang dilindungi MFA di Panduan Pengguna IAM](#).

Contoh pengaturan ini dalam config file:

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

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

Menentukan Nama Sumber Daya Amazon (ARN) peran IAM yang ingin Anda gunakan untuk melakukan operasi yang diminta menggunakan profil ini.

Nilai default: Tidak ada.

Nilai yang valid: Nilai harus ARN dari peran IAM, 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 instans Amazon EC2, atau instans 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_IAM_ROLE_SESSION_NAME**- variabel lingkungan, **aws.roleSessionName**- Properti sistem JVM: 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.

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 saat AWS API AWS CLI atau 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.
```


untuk mengakses AWS sumber daya. Peran yang ditentukan harus memiliki kebijakan izin IAM yang dilampirkan yang memungkinkan kode yang diminta berjalan, seperti perintah Layanan AWS, atau metode API.

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

AWS_WEB_IDENTITY_TOKEN_FILE- variabel lingkungan, **aws.webIdentityTokenFile**- Properti sistem JVM: 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. SDK atau alat pengembang 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 SDK

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

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

SDK	Di	Catatan atau informasi lebih lanjut
SDK for Java 2.x	Parsi	<code>mfa_serial</code> tidak didukung. Gunakan <code>AWS_ROLE_ARN</code> sebagai pengganti <code>AWS_IAM_ROLE_ARN</code> . Gunakan <code>AWS_ROLE_SESSION_NAME</code> sebagai pengganti <code>AWS_IAM_ROLE_SESSION_NAME</code> .
SDK for Java 1.x	Parsi	<code>mfa_serial</code> tidak didukung. Properti sistem JVM tidak didukung.
SDK untuk 3.x JavaScript	Ya	
SDK untuk 2.x JavaScript	Parsi	<code>credential_source</code> tidak didukung.
SDK para Kotlin	Ya	Gunakan <code>AWS_ROLE_ARN</code> sebagai pengganti <code>AWS_IAM_ROLE_ARN</code> . Gunakan <code>AWS_ROLE_SESSION_NAME</code> sebagai pengganti <code>AWS_IAM_ROLE_SESSION_NAME</code> .
SDK for .NET 3.x	Ya	
SDK for PHP 3.x	Ya	
SDK untuk Python (Boto3)	Ya	
SDK for Ruby 3.x	Ya	
SDK untuk Rust	Ya	
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). SDK mencoba memuat kredensi dari titik akhir HTTP yang ditentukan melalui permintaan GET.

Jika Anda menggunakan Amazon ECS, sebaiknya gunakan tugas IAM Role untuk meningkatkan isolasi kredensi, otorisasi, dan auditabilitas. Saat dikonfigurasi, Amazon ECS menetapkan variabel

`AWS_CONTAINER_CREDENTIALS_RELATIVE_URI` lingkungan yang digunakan SDK dan alat untuk mendapatkan kredensial. Untuk mengonfigurasi Amazon ECS untuk fungsionalitas ini, lihat [Peran IAM 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. Pod Anda dan peran IAM dikaitkan dengan akun layanan Kubernetes untuk mengelola kredensial-kredensial aplikasi Anda. Untuk mempelajari selengkapnya tentang Identitas Pod [Amazon EKS](#), lihat [Identitas Pod](#) Amazon EKS di Panduan Pengguna Amazon EKS. Saat dikonfigurasi, Amazon EKS menyetel variabel `AWS_CONTAINER_CREDENTIALS_FULL_URI` dan `AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE` lingkungan yang digunakan SDK dan alat untuk mendapatkan kredensial. Untuk informasi penyiapan, lihat [Menyiapkan Agen Identitas Pod Amazon EKS](#) di Panduan Pengguna Amazon EKS atau [Amazon EKS Pod Identity menyederhanakan izin IAM untuk aplikasi di kluster Amazon EKS](#) di situs web Blog. AWS

Konfigurasi fungsi ini dengan menggunakan yang berikut:

`AWS_CONTAINER_CREDENTIALS_FULL_URI`- variabel lingkungan

Menentukan titik akhir URL HTTP lengkap untuk SDK yang akan digunakan saat membuat permintaan kredensial. Ini termasuk skema dan tuan rumah.

Nilai default: Tidak ada.

Nilai yang valid: URI yang valid.

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 titik akhir URL HTTP relatif untuk SDK yang akan digunakan saat membuat permintaan kredensial. Nilai ditambahkan ke nama host Amazon ECS default dari 169.254.170.2

Nilai default: Tidak ada.

Nilai yang valid: URI relatif yang valid.

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 disetel, SDK akan mengatur header Otorisasi pada permintaan HTTP 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 SDK

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

SDK	Di	Catatan atau informasi lebih lanjut
AWS CLI v2	Ya	
SDK for C++	Ya	
SDK for Go V2 (1.x)	Ya	
SDK for Go 1.x (V1)	Ya	
SDK for Java 2.x	Ya	
SDK for Java 1.x	Parsi	Amazon EKS Pod Identity dan AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE tidak didukung.
SDK untuk 3.x JavaScript	Ya	
SDK untuk 2.x JavaScript	Ya	
SDK para Kotlin	Ya	
SDK for .NET 3.x	Ya	
SDK for PHP 3.x	Ya	
SDK untuk Python (Boto3)	Ya	
SDK for Ruby 3.x	Ya	
SDK untuk Rust	Ya	
Alat untuk PowerShell	Ya	

Penyedia kredensi Pusat Identitas IAM

Mekanisme otentikasi ini digunakan AWS IAM Identity Center untuk mendapatkan akses masuk tunggal (SSO) untuk Layanan AWS kode Anda.

Note

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

Setelah mengaktifkan Pusat Identitas IAM, Anda menentukan profil untuk pengaturannya di AWS `config` file bersama Anda. Profil ini digunakan untuk terhubung ke portal akses Pusat Identitas IAM. Ketika pengguna berhasil mengautentikasi dengan IAM Identity Center, portal mengembalikan kredensi jangka pendek untuk peran IAM 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 IAM Identity Center melalui `config` file:

- Konfigurasi penyedia token SSO (disarankan) - Durasi sesi yang diperpanjang.
- Konfigurasi lama yang tidak dapat disegarkan - Menggunakan sesi delapan jam yang tetap.

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

Untuk mengatur durasi sesi kustom, Anda harus menggunakan konfigurasi penyedia token SSO.

Dua panduan berikut berisi informasi tambahan tentang IAM Identity Center:

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

Prasyarat

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

Atau, ikuti [Autentikasi Pusat Identitas IAM](#) instruksi dalam panduan ini. Instruksi ini memberikan panduan lengkap, mulai dari mengaktifkan IAM Identity Center hingga menyelesaikan konfigurasi config file bersama yang diperlukan berikut di sini.

Konfigurasi penyedia token SSO

Note

Untuk menggunakan AWS CLI untuk membuat konfigurasi ini untuk Anda, lihat [Mengkonfigurasi profil Anda dengan `aws configure sso wizard` di AWS CLI](#).

Saat Anda menggunakan konfigurasi penyedia token SSO, 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 Identitas IAM](#) dalam Panduan AWS IAM Identity Center Pengguna.

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

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

Note

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

Contoh berikut mengonfigurasi SDK untuk meminta kredensial Pusat Identitas IAM. Ini juga mendukung penyegaran token otomatis.

```
[profile dev]
sso_session = my-ss0
sso_account_id = 111122223333
```

```
sso_role_name = SampleRole

[sso-session my-ss0]
sso_region = us-east-1
sso_start_url = https://my-ss0-portal.awsapps.com/start
sso_registration_scopes = sso:account:access
```

Anda dapat menggunakan kembali sso-session konfigurasi di beberapa profil.

```
[profile dev]
sso_session = my-ss0
sso_account_id = 111122223333
sso_role_name = SampleRole

[profile prod]
sso_session = my-ss0
sso_account_id = 111122223333
sso_role_name = SampleRole2

[sso-session my-ss0]
sso_region = us-east-1
sso_start_url = https://my-ss0-portal.awsapps.com/start
sso_registration_scopes = sso:account:access
```

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

Lingkup pendaftaran dikonfigurasi sebagai bagian dari filesso-session. Lingkup adalah mekanisme OAuth 2.0 untuk membatasi akses aplikasi ke akun pengguna. Aplikasi dapat meminta satu atau lebih cakupan, dan token akses yang dikeluarkan untuk aplikasi terbatas pada cakupan yang diberikan. Cakupan ini menentukan izin yang diminta untuk diotorisasi untuk klien OIDC terdaftar dan token akses yang diambil oleh klien. Untuk opsi cakupan akses yang didukung, lihat [Cakupan akses](#) di Panduan AWS IAM Identity Center Pengguna. Contoh berikut ditetapkan sso_registration_scopes untuk menyediakan akses untuk daftar akun dan peran.

```
[sso-session my-ss0]
```

```
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 [Konfigurasi penyedia token SSO](#) 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`

Pengaturan penyedia kredensi Pusat Identitas IAM

Konfigurasi fungsi ini dengan menggunakan yang berikut ini:

sso_start_url- Pengaturan AWS **config** file bersama

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

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

sso_region- Pengaturan AWS **config** file bersama

Yang Wilayah AWS berisi host portal Pusat Identitas IAM Anda; yaitu, Wilayah yang Anda pilih sebelum mengaktifkan Pusat Identitas IAM. 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 Identitas IAM](#), 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 Identitas IAM](#) dan buka Akun AWSHalaman. Anda juga dapat melihat daftar akun yang tersedia menggunakan metode [ListAccounts](#)API di Referensi API 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 peran IAM yang mendefinisikan 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).

Set izin memiliki kebijakan IAM 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 Identitas IAM](#) dan buka Akun AWSHalaman. Pilih nama set izin yang benar yang tercantum dalam Akun AWS tabel. Anda juga dapat melihat daftar set izin yang tersedia menggunakan metode [ListAccountRoles](#)API di Referensi API 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` Cakupan mengotorisasi akses ke titik akhir resmi token pembawa IAM Identity Center. Cakupan minimum `sso:account:access` harus diberikan untuk mendapatkan token penyegaran kembali dari layanan IAM Identity Center. Untuk string cakupan akses yang didukung, lihat [Cakupan akses](#) di AWS IAM Identity Center Panduan Pengguna. 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 SDK

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

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

SDK	Di	Catatan atau informasi lebih lanjut
SDK untuk Python (Boto3)	Ya	
SDK for Ruby 3.x	Ya	
SDK untuk Rust	Parsi	Konfigurasi lama yang tidak dapat disegarkan saja.
Alat untuk PowerShell	Ya	

Penyedia kredensi IMDS

Layanan Metadata Instance (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 [Metadana instans dan data pengguna](#) di Panduan Pengguna Amazon EC2 atau [metadana Instans dan data pengguna di](#) Panduan Pengguna Amazon EC2. 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. SDK dapat menggunakan titik akhir tersebut untuk menyelesaikan kredensial sebagai bagian dari rantai penyedia kredensial [default](#) mereka. Instance Metadata Service Version 2 (IMDSv2), versi IMDS yang lebih aman yang menggunakan token sesi, digunakan secara default. Jika gagal karena kondisi yang tidak dapat dicoba ulang (kode kesalahan HTTP 403, 404, 405), IMDSv1 digunakan sebagai fallback.

Konfigurasi fungsi ini dengan menggunakan yang berikut:

AWS_EC2_METADATA_DISABLED- variabel lingkungan

Apakah akan mencoba menggunakan Layanan Metadata Instans Amazon EC2 (IMDS) untuk mendapatkan kredensial atau tidak.


Nilai default: `false`.

Nilai yang valid:

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

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

Apakah akan menggunakan Layanan Metadata Instance Versi 1 (IMDSv1) sebagai fallback jika IMDSv2 gagal.

 Note

SDK baru tidak mendukung IMDSv1 dan, karenanya, tidak mendukung pengaturan ini. Untuk detailnya, lihat tabel [Kompatibilitas dengan AWS SDK](#).

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 sistem JVM: Hanya Java/Kotlin

Titik akhir IMDS.

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: URI yang valid.

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

Mode titik akhir IMDS.

Nilai default: `IPv4`.

Nilai yang valid: `IPv4`, `IPv6`.

Note

Penyedia kredensi IMDS adalah bagian dari [Rantai penyedia kredensi](#). Namun, penyedia kredensi IMDS 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 IMDS dalam kode. Anda dapat menentukan sumber kredensi secara langsung saat Anda membuat klien layanan.

Keamanan untuk kredensi IMDS

Secara default, ketika AWS SDK tidak dikonfigurasi dengan kredensial yang valid, SDK akan mencoba menggunakan Layanan Metadata Instans Amazon EC2 (IMDS) untuk mengambil kredensi 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 Instans Amazon EC2 dapat ditiru.

Note

AWS Klien SDK yang dikonfigurasi dengan kredensi yang valid tidak akan pernah menggunakan IMDS untuk mengambil kredensi, terlepas dari pengaturan ini.

Menonaktifkan penggunaan kredensi 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 SDK

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

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

SDK	Di	Catatan atau informasi lebih lanjut
SDK for Java 2.x	Ya	
SDK for Java 1.x	Parsi	Properti sistem JVM: 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 3.x JavaScript	Ya	
SDK untuk 2.x JavaScript	Ya	
SDK para Kotlin	Ya	Tidak menggunakan fallback IMDSv1.
SDK for .NET 3.x	Ya	
SDK for PHP 3.x	Ya	
SDK untuk Python (Boto3)	Ya	
SDK for Ruby 3.x	Ya	
SDK untuk Rust	Ya	Tidak menggunakan fallback IMDSv1.
Alat untuk PowerShell	Ya	Anda dapat menonaktifkan fallback IMDSv1 secara eksplisit dalam kode menggunakan. <code>[Amazon.Util.EC2InstanceMetadata]::EC2MetadataV1Disabled = \$true</code>

Penyedia kredensi proses

SDK menyediakan cara untuk memperluas rantai penyedia kredensial untuk kasus penggunaan khusus.

IAM Roles Anywhere menyediakan cara untuk mendapatkan kredensial sementara untuk beban kerja atau proses yang berjalan di luar. `AWScredential_process` Untuk mengonfigurasi penggunaan ini, lihat [Peran IAM Di Mana Saja](#).

⚠ Warning

Berikut ini menjelaskan metode sumber kredensial dari proses eksternal. Ini berpotensi berbahaya, jadi lanjutkan dengan hati-hati. Penyedia kredensi lainnya harus lebih disukai jika memungkinkan. Jika menggunakan opsi ini, Anda harus memastikan bahwa `config` file tersebut dikunci semaksimal mungkin menggunakan praktik terbaik keamanan untuk sistem operasi Anda. Konfirmasikan bahwa alat kredensi kustom Anda tidak menulis informasi rahasia apa pun `stderr`, karena SDK dan 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- Pengaturan AWS `config` file bersama

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

ℹ Note

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

Menentukan jalur ke program kredensyal

Nilai setelan 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.

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"
```

Output yang valid dari program kredensyal

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 output JSON 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 ke1. Ini mungkin meningkat seiring waktu seiring berkembangnya struktur.

`Expiration` kuncinya adalah stempel waktu berformat RFC3339. Jika `Expiration` kunci tidak ada dalam output alat, SDK mengasumsikan 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 tidak menyimpan kredensial proses eksternal seperti halnya 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 kredensi.

Kompatibilitas dengan AWS SDK

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

SDK	Di Catatan atau informasi lebih lanjut
AWS CLI v2	Ya
SDK for C++	Ya

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

AWS Fitur standar SDK dan Tools

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

Important

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

Topik

- [ID Aplikasi](#)
- [Metadata instans Amazon EC2](#)
- [Titik akses Amazon S3](#)
- [Titik Akses Multi-Wilayah Amazon S3](#)
- [Wilayah AWS](#)
- [AWS STS Titik akhir regional](#)
- [Dual-stack dan titik akhir FIPS](#)
- [Penemuan titik akhir](#)
- [Pengaturan konfigurasi umum](#)
- [Klien IMDS](#)
- [Coba lagi perilaku](#)
- [Minta kompresi](#)
- [Titik akhir khusus layanan](#)
- [Default konfigurasi cerdas](#)

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 file Akun AWS. AWS SDK dan layanan tidak menggunakan atau menafsirkan nilai ini selain untuk memunculkannya kembali dalam komunikasi pelanggan. Misalnya, nilai ini dapat dimasukkan dalam email operasional atau AWS Health Dashboard untuk mengidentifikasi secara unik aplikasi mana yang terkait dengan notifikasi.

Konfigurasi fungsi ini dengan menggunakan yang berikut ini:

sdk_ua_app_id- Pengaturan AWS **config** file bersama, **AWS_SDK_UA_APP_ID**- variabel lingkungan, **aws.userAgentAppId**- Properti sistem JVM: Hanya Java/Kotlin

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

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

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

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

Metadata instans Amazon EC2

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

Konfigurasi fungsi ini dengan menggunakan yang berikut:

metadata_service_num_attempts- Pengaturan AWS **config** file bersama,
AWS_METADATA_SERVICE_NUM_ATTEMPTS- variabel lingkungan

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

Nilai default: 1

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

metadata_service_timeout- Pengaturan AWS **config** file bersama, **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 SDK

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

SDK	Di Catatan atau informasi lebih lanjut
AWS CLI v2	Ya
SDK for C++	Tida

SDK	Di	Catatan atau informasi lebih lanjut
SDK for Go V2 (1.x)	Tida	
SDK for Go 1.x (V1)	Tida	
SDK for Java 2.x	Tida	
SDK for Java 1.x	Parsi	Hanya AWS_METADATA_SERVICE_TIMEOUT didukung.
SDK untuk 3.x JavaScript	Tida	
SDK untuk 2.x JavaScript	Tida	
SDK para Kotlin	Tida	
SDK for .NET 3.x	Tida	
SDK for PHP 3.x	Ya	
SDK untuk Python (Boto3)	Ya	
SDK for Ruby 3.x	Tida	
SDK untuk Rust	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 SDK, Anda dapat menggunakan access point Amazon Resource Names (ARN) di kolom bucket untuk operasi API, bukan 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 lebih lanjut tentang jalur akses Amazon S3 dan ARN, lihat [Menggunakan titik akses di Panduan Pengguna](#) Amazon S3.

Konfigurasi fungsi ini dengan menggunakan yang berikut ini:

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

Pengaturan ini mengontrol apakah SDK menggunakan Wilayah AWS ARN titik akses untuk membangun titik akhir Regional untuk permintaan tersebut. SDK memvalidasi bahwa ARN Wilayah AWS dilayani oleh AWS partisi yang sama dengan 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 menggunakan 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 dengan ARN. Wilayah AWS Jika tidak, kesalahan akan terjadi.
- **false**- SDK menggunakan konfigurasi klien Wilayah AWS saat membangun titik akhir.

Kompatibilitas dengan AWS SDK

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

SDK	Di	Catatan atau informasi lebih lanjut
AWS CLI v2	Ya	
SDK for C++	Ya	
SDK for Go V2 (1.x)	Ya	
SDK for 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 for Java 2.x	Ya	
SDK for Java 1.x	Ya	Properti sistem JVM tidak didukung.
SDK untuk 3.x JavaScript	Ya	
SDK untuk 2.x JavaScript	Ya	
SDK para Kotlin	Ya	
SDK for .NET 3.x	Ya	Tidak mengikuti prioritas standar; nilai config file bersama lebih diutamakan daripada variabel lingkungan.
SDK for PHP 3.x	Ya	
SDK untuk Python (Boto3)	Ya	
SDK for Ruby 3.x	Ya	
SDK untuk Rust	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 tempat. 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 selengkapnya tentang Nama Sumber Daya Amazon (ARN) 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 oleh SDK melalui ketergantungan pada.

[AWSPustaka Runtime Umum \(CRT\)](#)

Konfigurasi fungsi ini dengan menggunakan yang berikut ini:

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

Setelan ini mengontrol apakah SDK berpotensi mencoba permintaan lintas wilayah. 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 SDK

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

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

SDK	Di	Catatan atau informasi lebih lanjut
SDK for Go 1.x (V1)	Tida	
SDK for Java 2.x	Ya	
SDK for Java 1.x	Tida	
SDK untuk 3.x JavaScript	Ya	
SDK untuk 2.x JavaScript	Tida	
SDK para Kotlin	Ya	
SDK for .NET 3.x	Ya	
SDK for PHP 3.x	Ya	
SDK untuk Python (Boto3)	Ya	
SDK for Ruby 3.x	Ya	
SDK untuk Rust	Ya	
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, dan 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, SDK menggunakan kelas/objek klien untuk masing-masing. Layanan AWS Jika aplikasi Anda perlu mengakses Amazon EC2, misalnya, aplikasi Anda akan membuat objek klien Amazon EC2 untuk berinteraksi dengan layanan itu.

Jika tidak ada Region yang secara eksplisit ditentukan untuk klien, 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 SDK Anda (periksa Panduan SDK spesifik Anda atau basis kode SDK Anda).

Konfigurasi fungsi ini dengan menggunakan yang berikut:

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

Menentukan default Wilayah AWS untuk digunakan untuk AWS permintaan. Wilayah ini digunakan untuk permintaan layanan SDK 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 SDK memiliki objek “konfigurasi” yang tersedia untuk mengatur Wilayah default dari dalam kode aplikasi. Untuk detailnya, lihat panduan developer AWS SDK spesifik Anda.

Kompatibilitas dengan AWS SDK

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

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 for C++	Ya	
SDK for Go V2 (1.x)	Ya	

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

AWS STS Titik akhir regional

Secara default, AWS Security Token Service (AWS STS) tersedia sebagai layanan global, dan semua AWS STS permintaan masuk ke satu titik akhir di `https://sts.amazonaws.com`. Permintaan global dipetakan ke Wilayah AS Timur (Virginia N.). AWS merekomendasikan penggunaan AWS STS titik akhir Regional alih-alih titik akhir global. Untuk informasi selengkapnya tentang AWS STS titik akhir, [Titik Akhir dalam Referensi AWS Security Token Service API](#).

Konfigurasi fungsi ini dengan menggunakan yang berikut ini:

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

Setelan ini menentukan cara 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 utama SDK baru yang dirilis setelah Juli 2022 akan default ke `regional`. Versi utama SDK 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`, untuk AWS Wilayah berikut: `ap-northeast-1`, `ap-south-1`, `ap-southeast-1`, `ap-southeast-2`, `aws-global`, `ca-central-1`, `eu-central-1`, `eu-north-1`, `eu-west-1`, `eu-west-2`, `eu-west-3`, `sa-east-1`, `us-east-1`, `us-east-2`, `us-west-1`, dan `us-west-2`. Semua Wilayah lainnya secara otomatis menggunakan titik akhir Regional masing-masing.
- **regional**- SDK atau alat 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 SDK

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

SDK	Di	Catatan atau informasi lebih lanjut
AWS CLI v2	Parsi	Nilai default-nya adalah <code>regional</code> .
SDK for C++	Parsi	Variabel lingkungan dan pengaturan config file tidak didukung. SDK bekerja dengan <code>regional</code> pengaturan.
SDK for Go V2 (1.x)	Ya	
SDK for Go 1.x (V1)	Ya	Untuk menggunakan pengaturan config file bersama, Anda harus mengaktifkan pemuatan dari file konfigurasi; lihat Sesi .
SDK for Java 2.x	Ya	
SDK for Java 1.x	Ya	
SDK untuk 3.x JavaScript	Ya	
SDK untuk 2.x JavaScript	Ya	
SDK para Kotlin	Tida	
SDK for .NET 3.x	Ya	
SDK for PHP 3.x	Ya	
SDK untuk Python (Boto3)	Ya	
SDK for Ruby 3.x	Ya	
SDK untuk Rust	Ya	

SDK	Di Catatan atau informasi lebih lanjut
Alat untuk PowerShell	Ya

Dual-stack dan titik akhir FIPS

Konfigurasi fungsi ini dengan menggunakan yang berikut ini:

use_dualstack_endpoint- Pengaturan AWS **config** file bersama, **AWS_USE_DUALSTACK_ENDPOINT**- variabel lingkungan, **aws.useDualstackEndpoint**- Properti sistem JVM: Hanya Java/Kotlin

Mengaktifkan atau menonaktifkan apakah SDK akan mengirim permintaan ke titik akhir tumpukan ganda. Untuk mempelajari lebih lanjut tentang titik akhir dual-stack, yang mendukung lalu lintas IPv4 dan IPv6, 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** SDK atau alat akan mencoba menggunakan titik akhir dual-stack untuk membuat permintaan jaringan. Jika titik akhir dual-stack tidak ada untuk layanan dan/atau Wilayah AWS, permintaan akan gagal.
- **false** SDK atau alat tidak akan menggunakan titik akhir dual-stack untuk membuat permintaan jaringan.

use_fips_endpoint- Pengaturan AWS **config** file bersama, **AWS_USE_FIPS_ENDPOINT**- variabel lingkungan, **aws.useFipsEndpoint**- Properti sistem JVM: Hanya Java/Kotlin

Mengaktifkan atau menonaktifkan apakah SDK atau alat akan mengirim permintaan ke titik akhir yang sesuai dengan FIPS. Federal Information Processing Standards (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 pedoman FIPS. Tidak seperti AWS endpoint standar, endpoint FIPS menggunakan pustaka perangkat lunak TLS yang sesuai dengan FIPS 140-2. Jika pengaturan ini diaktifkan dan titik akhir FIPS tidak ada untuk layanan di Anda Wilayah AWS, AWS panggilan mungkin gagal. [Titik](#)

[akhir khusus layanan](#) dan `--endpoint-url` opsi untuk AWS Command Line Interface mengganti pengaturan ini.

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

Nilai default: `false`

Nilai yang valid:

- **true** SDK atau alat akan mengirim permintaan ke titik akhir yang sesuai dengan FIPS.
- **false** SDK atau alat tidak akan mengirim permintaan ke titik akhir yang sesuai dengan FIPS.

Kompatibilitas dengan AWS SDK

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

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

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

Penemuan titik akhir

SDK menggunakan penemuan titik akhir untuk mengakses titik akhir layanan (URL untuk mengakses berbagai sumber daya), sambil tetap mempertahankan fleksibilitas AWS untuk mengubah URL sesuai 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, lalu menggunakan titik akhir tersebut untuk menyelesaikan operasi tertentu seperti `atau. CreateDatabase CreateTable`

Konfigurasi fungsi ini dengan menggunakan yang berikut:

endpoint_discovery_enabled- Pengaturan AWS **config** file bersama, **AWS_ENABLE_ENDPOINT_DISCOVERY**- variabel lingkungan, **aws.endpointDiscoveryEnabled**- Properti sistem JVM: Hanya Java/Kotlin, Untuk mengonfigurasi nilai secara langsung dalam kode, lihat SDK spesifik Anda 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 bersifat opsional.
- **false** SDK seharusnya tidak secara otomatis mencoba menemukan titik akhir untuk layanan di mana penemuan titik akhir bersifat opsional.

Kompatibilitas dengan AWS SDK

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

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

SDK	Di	Catatan atau informasi lebih lanjut
SDK for Ruby 3.x	Ya	
SDK untuk Rust	Parsi	Didukung hanya untuk Timestream.
Alat untuk PowerShell	Ya	

Pengaturan konfigurasi umum

SDK mendukung beberapa pengaturan umum yang mengonfigurasi perilaku SDK secara keseluruhan.

Konfigurasikan fungsi ini dengan menggunakan yang berikut:

api_versions- Pengaturan AWS **config** file bersama

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

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

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

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

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

ca_bundle- Pengaturan AWS **config** file bersama, **AWS_CA_BUNDLE**- variabel lingkungan

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

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
```

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- Pengaturan AWS **config** file bersama

Menentukan bagaimana hasil diformat dalam AWS SDK AWS CLI dan alat lainnya.

Nilai default: json

Nilai yang valid:

- **json**— Output diformat sebagai string [JSON](#).
- **yaml**- Output diformat sebagai string [YAML](#).
- **yaml-stream**— Output dialirkan dan diformat sebagai string [YAML](#). 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- Pengaturan AWS **config** file bersama

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

Nilai default: `true`

Nilai yang valid:

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

Kompatibilitas dengan AWS SDK

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

SDK	Di	Catatan atau informasi lebih lanjut
AWS CLI v2	Parsi	<code>api_versions</code> tidak didukung.
SDK for C++	Ya	
SDK for Go V2 (1.x)	Parsi	<code>api_versions</code> dan <code>parameter_validation</code> tidak didukung.
SDK for 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 for Java 2.x	Tida	
SDK for Java 1.x	Tida	
SDK untuk 3.x JavaScript	Ya	
SDK untuk 2.x JavaScript	Ya	

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

Klien IMDS

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

Konfigurasi fungsi ini dengan menggunakan yang berikut:

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 token.

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

Nilai yang valid: Nomor.

endpoint- anggota objek konfigurasi klien

Titik akhir 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: URI yang valid.

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

endpoint_mode- anggota objek konfigurasi klien

Mode titik akhir 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, seperti STDOUT.

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 menurut SDK.

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

Kompatibilitas dengan AWS SDK

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

SDK	Di	Catatan atau informasi lebih lanjut
AWS CLI v2	Ya	
SDK for C++	Tida	IMDSv2 hanya digunakan secara internal. Lihat Penyedia kredensi IMDS .
SDK for Go V2 (1.x)	Ya	
SDK for Go 1.x (V1)	Ya	
SDK for Java 2.x	Ya	
SDK for Java 1.x	Ya	
SDK untuk 3.x JavaScript	Ya	
SDK untuk 2.x JavaScript	Ya	

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

Coba lagi perilaku

Perilaku coba lagi mencakup pengaturan mengenai cara SDK mencoba memulihkan dari kegagalan yang dihasilkan dari permintaan yang dibuat. Layanan AWS

Konfigurasi fungsi ini dengan menggunakan yang berikut:

max_attempts- Pengaturan AWS **config** file bersama, **AWS_MAX_ATTEMPTS**- variabel lingkungan, **aws.maxAttempts**- Properti sistem JVM: 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 SDK Anda (periksa panduan SDK spesifik Anda atau basis kode SDK 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.

retry_mode- Pengaturan AWS **config** file bersama, **AWS_RETRY_MODE**- variabel lingkungan, **aws.retryMode**- Properti sistem JVM: Hanya Java/Kotlin

Menentukan cara SDK atau alat pengembang mencoba mencoba ulang.

Nilai default: `legacy` adalah strategi coba ulang default.

Nilai yang valid:

- `legacy`— Khusus untuk SDK Anda (periksa panduan SDK spesifik Anda atau basis kode SDK Anda).
- `standard`— Set standar aturan coba ulang di seluruh AWS SDK. Mode ini mencakup serangkaian kesalahan standar yang dicoba ulang, dan dukungan untuk kuota coba lagi. Jumlah maksimum percobaan default dengan mode ini adalah tiga, kecuali `max_attempts` dikonfigurasi secara eksplisit.
- `adaptive`— Mode coba lagi eksperimental yang mencakup fungsionalitas mode standar tetapi mencakup pelambatan sisi klien otomatis. Karena mode ini eksperimental, mungkin mengubah perilaku di masa depan.

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 oleh sumber daya layanan yang sedang diakses.

Standar	Adaptif
Mendukung pemutusan sirkuit untuk mencegah SDK mencoba lagi selama pemadaman.	Mendukung pemutusan sirkuit untuk mencegah SDK mencoba lagi selama 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 Wilayah AWS menggunakan plus 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 `egion-and-table` pasangan R. Jika Anda mengalami latensi tak terduga saat menggunakan `adaptive mode`, lihat panduan AWS dokumentasi khusus untuk layanan yang Anda gunakan.

Coba lagi detail implementasi mode

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:

Bucket token hanya digunakan dalam mode `adaptive` coba lagi. Bucket token memberlakukan tingkat permintaan maksimum dengan mewajibkan token tersedia untuk memulai permintaan. Klien SDK dapat dikonfigurasi untuk gagal permintaan atau pemblokiran dengan cepat hingga token tersedia.

Pembatasan Tingkat Sisi Klien adalah algoritma yang awalnya memungkinkan permintaan dibuat pada tingkat berapa pun hingga tunjangan token. Namun, setelah respons terbatas terdeteksi, klien kemudian `rate-of-request` dibatasi. Tunjangan token juga meningkat jika tanggapan berhasil diterima.

Dengan pembatasan laju adaptif, SDK dapat memperlambat laju pengiriman permintaan untuk mengakomodasi kapasitas dengan lebih baik. Layanan AWS

SendHTTPRequest:

Sebagian besar AWS SDK menggunakan pustaka HTTP yang menggunakan kumpulan koneksi sehingga Anda dapat menggunakan kembali koneksi yang ada saat membuat permintaan HTTP. Umumnya, koneksi digunakan kembali saat mencoba ulang permintaan karena kesalahan pelambatan. Permintaan tidak digunakan kembali saat mencoba ulang karena kesalahan sementara.

RequestBookkeeping:

Kuota coba lagi harus diperbarui jika permintaan berhasil. Hanya untuk mode `adaptive` coba lagi, variabel status `maxsendrate` diperbarui berdasarkan jenis respons yang diterima.

Retryable:

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

- Kode status HTTP.

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

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

MAX_ATTEMPTS:

Ditentukan oleh pengaturan config file atau variabel lingkungan.

HasRetryQuota

Langkah ini membatasi permintaan coba lagi dengan mewajibkan token tersedia di bucket kuota coba lagi. Ember kuota coba lagi adalah mekanisme untuk mencegah percobaan ulang yang tidak mungkin berhasil. Kuota ini bergantung pada SDK, seringkali bergantung pada klien, dan terkadang bahkan bergantung pada titik akhir layanan. Token kuota coba lagi yang tersedia dihapus ketika permintaan gagal karena berbagai alasan, dan diisi ulang ketika berhasil. Ketika tidak ada token yang tersisa, loop coba lagi akan keluar.

ExponentialBackoff

Untuk kesalahan yang dapat dicoba lagi, penundaan coba lagi dihitung menggunakan backoff eksponensial terpotong. SDK menggunakan 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 SDK. Lihat panduan SDK atau kode sumber khusus Anda untuk konfirmasi.

Kompatibilitas dengan AWS SDK

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

SDK	Di	Catatan atau informasi lebih lanjut
AWS CLI v2	Ya	
SDK for C++	Ya	
SDK for Go V2 (1.x)	Ya	
SDK for Go 1.x (V1)	Tida	
SDK for Java 2.x	Ya	
SDK for Java 1.x	Ya	Properti sistem JVM: 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 3.x JavaScript	Ya	
SDK untuk 2.x JavaScript	Tida	Mendukung jumlah percobaan ulang maksimum, backoff eksponensial dengan jitter, dan opsi untuk metode khusus untuk coba lagi backoff.
SDK para Kotlin	Ya	
SDK for .NET 3.x	Ya	
SDK for PHP 3.x	Ya	
SDK untuk Python (Boto3)	Ya	
SDK for Ruby 3.x	Ya	
SDK untuk Rust	Ya	

SDK	Di Catatan atau informasi lebih lanjut
Alat untuk PowerShell	Ya

Minta kompresi

AWS SDK dan alat dapat secara otomatis memampatkan muatan saat mengirim permintaan ke dukungan Layanan AWS yang 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, SDK atau alat 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 metode [PutMetricDataAPI](#).

Konfigurasi fungsi ini dengan menggunakan yang berikut ini:

disable_request_compression- Pengaturan AWS **config** file bersama, **AWS_DISABLE_REQUEST_COMPRESSION**- variabel lingkungan, **aws.disableRequestCompression**- Properti sistem JVM: Hanya Java/Kotlin

Menghidupkan atau menonaktifkan apakah SDK atau alat akan mengompres 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- Pengaturan AWS **config** file bersama, **AWS_REQUEST_MIN_COMPRESSION_SIZE_BYTES**- variabel lingkungan, **aws.requestMinCompressionSizeBytes**- Properti sistem JVM: Hanya Java/Kotlin

Menetapkan ukuran minimum dalam byte dari badan permintaan yang harus dikompres oleh SDK atau alat. 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 SDK

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

SDK	Di Catatan atau informasi lebih lanjut
AWS CLI v2	Ya
SDK for C++	Ya
SDK for Go V2 (1.x)	Ya
SDK for Go 1.x (V1)	Tida
SDK for Java 2.x	Ya
SDK for Java 1.x	Tida
SDK untuk 3.x JavaScript	Ya
SDK untuk 2.x JavaScript	Tida
SDK para Kotlin	Ya
SDK for .NET 3.x	Ya

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

Titik akhir khusus layanan

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

Konfigurasi fungsi ini dengan menggunakan yang berikut ini:

endpoint_url- Pengaturan AWS **config** file bersama, **AWS_ENDPOINT_URL**- variabel lingkungan, **aws.endpointUrl**- Properti sistem JVM: 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 pengaturan ini dalam `services` bagian AWS `config` file bersama 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: URL termasuk skema dan host untuk titik akhir. URL secara opsional dapat berisi komponen jalur yang berisi satu atau beberapa segmen jalur.

AWS_ENDPOINT_URL_<SERVICE>- variabel lingkungan, **aws.endpointUrl<ServiceName>**-

Properti sistem JVM: Hanya Java/Kotlin

AWS_ENDPOINT_URL_<SERVICE>, di <SERVICE> mana Layanan AWS pengenalan, menetapkan titik akhir khusus 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: URL termasuk skema dan host untuk titik akhir. URL secara opsional dapat berisi komponen jalur yang berisi satu atau beberapa segmen jalur.

ignore_configured_endpoint_urls- Pengaturan AWS **config** file bersama, **AWS_IGNORE_CONFIGURED_ENDPOINT_URLS**- variabel lingkungan, **aws.ignoreConfiguredEndpointUrls**- Properti sistem JVM: 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, menyertakan parameter baris `--endpoint-url` perintah dengan AWS CLI perintah atau meneruskan URL titik akhir ke konstruktor klien akan selalu berlaku.

Nilai default: false

Nilai yang valid:

- **true** SDK atau alat tidak membaca opsi konfigurasi khusus apa pun dari config file bersama atau dari variabel lingkungan untuk menyetel URL titik akhir.
- **false** SDK atau alat 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 URL titik akhir kustom, setel variabel lingkungan **AWS_ENDPOINT_URL** global.

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

Untuk merutekan permintaan spesifik Layanan AWS ke URL titik akhir kustom, gunakan variabel `AWS_ENDPOINT_URL_<SERVICE>` lingkungan. Amazon DynamoDB memiliki `serviceId` dari [DynamoDB](#). Untuk layanan ini, variabel lingkungan URL 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` a [Elastic Beanstalk](#). Layanan AWS Pengenal didasarkan pada model API `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 URL titik akhir khusus layanan yang akan digunakan 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 URL titik akhir khusus layanan untuk Amazon S3 dan di profil yang sama. AWS Elastic Beanstalk memiliki `serviceId` dari [Elastic Beanstalk](#). Layanan AWS Pengenal didasarkan pada model API `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 kredensi berbasis peran

Jika profil Anda memiliki kredensial berbasis peran yang dikonfigurasi melalui `source_profile` parameter untuk fungsionalitas peran IAM, 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 Amazon EC2, 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. Agar titik akhir global berlaku untuk profilB, Anda perlu mengatur `endpoint_url` langsung di dalam profilB. Untuk informasi lebih lanjut tentang `source_profile` pengaturan, lihat [Asumsikan penyedia kredensi peran](#).

Prioritas pengaturan

Pengaturan untuk fitur ini dapat digunakan pada saat yang sama tetapi hanya satu nilai yang akan diprioritaskan per layanan. Untuk panggilan API yang dibuat 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, penetapan eksplisit dapat berupa parameter yang Anda tetapkan saat membuat instance objek Layanan AWS klien atau 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 `bersamaconfig`.
5. Nilai yang diberikan oleh `endpoint_url` pengaturan ditentukan langsung `profile` dalam `config` file bersama.
6. URL endpoint default untuk masing-masing Layanan AWS digunakan terakhir.

Kompatibilitas dengan AWS SDK

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

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

SDK	Di Catatan atau informasi lebih lanjut
SDK untuk Rust	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	Ke Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE>
AccessAnalyzer	aw AWS_ENDPOINT_URL_ACCESSANALYZER
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
amp	aw AWS_ENDPOINT_URL_AMP
Amplify	aw AWS_ENDPOINT_URL_AMPLIFY

serviceId	Ki	Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE>
AmplifyBackend	ar	AWS_ENDPOINT_URL_AMPLIFYBACKEND
AmplifyUIBuilder	ar	AWS_ENDPOINT_URL_AMPLIFYUIBUILDER
API Gateway	ap	AWS_ENDPOINT_URL_API_GATEWAY
ApiGatewayManagem entApi	ap	AWS_ENDPOINT_URL_APIGATEWAYMANAGEMENTAPI
ApiGatewayV2	ap	AWS_ENDPOINT_URL_APIGATEWAYV2
AppConfig	ap	AWS_ENDPOINT_URL_APPCONFIG
AppConfigData	ap	AWS_ENDPOINT_URL_APPCONFIGDATA
AppFabric	ap	AWS_ENDPOINT_URL_APPFABRIC
Appflow	ap	AWS_ENDPOINT_URL_APPFLOW
AppIntegrations	ap	AWS_ENDPOINT_URL_APPINTEGRATIONS

serviceId	Kategori	Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE>
Application Auto Scaling	ap	AWS_ENDPOINT_URL_APPLICATION_AUTO_SCALING
Application Insights	ai	AWS_ENDPOINT_URL_APPLICATION_INSIGHTS
ApplicationCostProfiler	ap	AWS_ENDPOINT_URL_APPLICATIONCOSTPROFILER
App Mesh	am	AWS_ENDPOINT_URL_APP_MESH
AppRunner	ar	AWS_ENDPOINT_URL_APPRUNNER
AppStream	as	AWS_ENDPOINT_URL_APPSTREAM
AppSync	ap	AWS_ENDPOINT_URL_APPS_SYNC
ARC Zonal Shift	az	AWS_ENDPOINT_URL_ARC_ZONAL_SHIFT
Artifact	ar	AWS_ENDPOINT_URL_ARTIFACT
Athena	at	AWS_ENDPOINT_URL_ATHENA

serviceId	Kategori	Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE>
AuditManager	audit	AWS_ENDPOINT_URL_AUDITMANAGER
Auto Scaling	auto	AWS_ENDPOINT_URL_AUTO_SCALING
Auto Scaling Plans	auto	AWS_ENDPOINT_URL_AUTO_SCALING_PLANS
b2bi	block	AWS_ENDPOINT_URL_B2BI
Backup	block	AWS_ENDPOINT_URL_BACKUP
Backup Gateway	block	AWS_ENDPOINT_URL_BACKUP_GATEWAY
BackupStorage	block	AWS_ENDPOINT_URL_BACKUPSTORAGE
Batch	block	AWS_ENDPOINT_URL_BATCH
BCM Data Exports	block	AWS_ENDPOINT_URL_BCM_DATA_EXPORTS
Bedrock	block	AWS_ENDPOINT_URL_BEDROCK
Bedrock Agent	block	AWS_ENDPOINT_URL_BEDROCK_AGENT

serviceId	Kategori	Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE>
Bedrock Agent Runtime	bedrock-agent	AWS_ENDPOINT_URL_BEDROCK_AGENT_RUNTIME
Bedrock Runtime	bedrock	AWS_ENDPOINT_URL_BEDROCK_RUNTIME
billingconductor	billingconductor	AWS_ENDPOINT_URL_BILLINGCONDUCTOR
Braket	braket	AWS_ENDPOINT_URL_BRAKET
Budgets	budgets	AWS_ENDPOINT_URL_BUDGETS
Cost Explorer	costexplorer	AWS_ENDPOINT_URL_COST_EXPLORER
chatbot	chatbot	AWS_ENDPOINT_URL_CHATBOT
Chime	chime	AWS_ENDPOINT_URL_CHIME
Chime SDK Identity	chime-sdk-identity	AWS_ENDPOINT_URL_CHIME_SDK_IDENTITY
Chime SDK Media Pipelines	chime-sdk-media-pipelines	AWS_ENDPOINT_URL_CHIME_SDK_MEDIA_PIPELINES

serviceId	Ki Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE> pe la ur fil be Al co
Chime SDK Meetings	cl AWS_ENDPOINT_URL_CHIME_SDK_MEETINGS _f
Chime SDK Messaging	cl AWS_ENDPOINT_URL_CHIME_SDK_MESSAGING _f g
Chime SDK Voice	cl AWS_ENDPOINT_URL_CHIME_SDK_VOICE _f
CleanRooms	c: AWS_ENDPOINT_URL_CLEANROOMS s
CleanRoomsML	c: AWS_ENDPOINT_URL_CLEANROOMSML sr
Cloud9	c: AWS_ENDPOINT_URL_CLOUD9
CloudControl	c: AWS_ENDPOINT_URL_CLOUDCONTROL rc
CloudDirectory	c: AWS_ENDPOINT_URL_CLOUDDIRECTORY cl
CloudFormation	c: AWS_ENDPOINT_URL_CLOUDFORMATION at
CloudFront	c: AWS_ENDPOINT_URL_CLOUDFRONT t

serviceId	Ki Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE> pe la ur fil be Al co
CloudFront KeyValueStore	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
CloudSearch Domain	c: AWS_ENDPOINT_URL_CLOUDSEARCH_DOMAIN cl
CloudTrail	c: AWS_ENDPOINT_URL_CLOUDTRAIL l
CloudTrail Data	c: AWS_ENDPOINT_URL_CLOUDTRAIL_DATA l_
CloudWatch	c: AWS_ENDPOINT_URL_CLOUDWATCH h
codeartifact	c: AWS_ENDPOINT_URL_CODEARTIFACT a:
CodeBuild	c: AWS_ENDPOINT_URL_CODEBUILD

serviceId	Ki Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE> pe la ur fil be Al co
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:
CodeGuru Security	co AWS_ENDPOINT_URL_CODEGURU_SECURITY s:
CodeGuruProfiler	co AWS_ENDPOINT_URL_CODEGURUPROFILER r:
CodePipeline	co AWS_ENDPOINT_URL_CODEPIPELINE i:
CodeStar	co AWS_ENDPOINT_URL_CODESTAR
CodeStar connections	co AWS_ENDPOINT_URL_CODESTAR_CONNECTIONS co n:
codestar notificat ions	co AWS_ENDPOINT_URL_CODESTAR_NOTIFICATIONS no i:

serviceId	Ki Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE>
Cognito Identity	co AWS_ENDPOINT_URL_COGNITO_IDENTITY de
Cognito Identity Provider	co AWS_ENDPOINT_URL_COGNITO_IDENTITY_PROVIDER de r
Cognito Sync	co AWS_ENDPOINT_URL_COGNITO_SYNC y
Comprehend	co AWS_ENDPOINT_URL_COMPREHEND d
ComprehendMedical	co AWS_ENDPOINT_URL_COMPREHENDMEDICAL dr
Compute Optimizer	co AWS_ENDPOINT_URL_COMPUTE_OPTIMIZER p
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 o n:
ConnectCampaigns	co AWS_ENDPOINT_URL_CONNECTCAMPAIGNS m

serviceId	Ki Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE> pe la ur fil be Al co
ConnectCases	co AWS_ENDPOINT_URL_CONNECTCASES Se
ConnectParticipant	co AWS_ENDPOINT_URL_CONNECTPARTICIPANT rt
ControlTower	co AWS_ENDPOINT_URL_CONTROLTOWER we
Cost Optimization Hub	co AWS_ENDPOINT_URL_COST_OPTIMIZATION_HUB m: ht
Cost and Usage Report Service	co AWS_ENDPOINT_URL_COST_AND_USAGE_REPO ur: RT_SERVICE o: co
Customer Profiles	co AWS_ENDPOINT_URL_CUSTOMER_PROFILES p:
DataBrew	di AWS_ENDPOINT_URL_DATABREW
DataExchange	di AWS_ENDPOINT_URL_DATAEXCHANGE ng
Data Pipeline	di AWS_ENDPOINT_URL_DATA_PIPELINE l:

serviceId	Ki Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE> pe la ur fil be Al co
DataSync	d: AWS_ENDPOINT_URL_DATASYNC
DataZone	d: AWS_ENDPOINT_URL_DATAZONE
DAX	d: AWS_ENDPOINT_URL_DAX
Detective	d: AWS_ENDPOINT_URL_DETECTIVE
Device Farm	d: AWS_ENDPOINT_URL_DEVICE_FARM ir
DevOps Guru	d: AWS_ENDPOINT_URL_DEVOPS_GURU ir
Direct Connect	d: AWS_ENDPOINT_URL_DIRECT_CONNECT ni
Application Discovery Service	a: AWS_ENDPOINT_URL_APPLICATION_DISCOVE o: RY_SERVICE e: c:
DLM	d: AWS_ENDPOINT_URL_DLM
Database Migration Service	d: AWS_ENDPOINT_URL_DATABASE_MIGRATION_ m: SERVICE _:
DocDB	d: AWS_ENDPOINT_URL_DOCDB

serviceId	Kategori	Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE>
DocDB Elastic	documentdb	AWS_ENDPOINT_URL_DOCDB_ELASTIC
drs	discovery	AWS_ENDPOINT_URL_DRS
Directory Service	directory	AWS_ENDPOINT_URL_DIRECTORY_SERVICE
DynamoDB	dynamodb	AWS_ENDPOINT_URL_DYNAMODB
DynamoDB Streams	dynamodbstreams	AWS_ENDPOINT_URL_DYNAMODB_STREAMS
EBS	elasticblockstore	AWS_ENDPOINT_URL_EBS
EC2	ec2	AWS_ENDPOINT_URL_EC2
EC2 Instance Connect	ec2instanceconnect	AWS_ENDPOINT_URL_EC2_INSTANCE_CONNECT
ECR	ecr	AWS_ENDPOINT_URL_ECR
ECR PUBLIC	ecrpublic	AWS_ENDPOINT_URL_ECR_PUBLIC
ECS	ecs	AWS_ENDPOINT_URL_ECS
EFS	elasticfilesystem	AWS_ENDPOINT_URL_EFS
EKS	eks	AWS_ENDPOINT_URL_EKS

serviceId	Kategori	Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE>
EKS Auth	environment	AWS_ENDPOINT_URL_EKS_AUTH
Elastic Inference	environment	AWS_ENDPOINT_URL_ELASTIC_INFERENCE
ElastiCache	environment	AWS_ENDPOINT_URL_ELASTICACHE
Elastic Beanstalk	environment	AWS_ENDPOINT_URL_ELASTIC_BEANSTALK
Elastic Transcoder	environment	AWS_ENDPOINT_URL_ELASTIC_TRANSCODER
Elastic Load Balancing	environment	AWS_ENDPOINT_URL_ELASTIC_LOAD_BALANCING
Elastic Load Balancing v2	environment	AWS_ENDPOINT_URL_ELASTIC_LOAD_BALANCING_V2
EMR	environment	AWS_ENDPOINT_URL_EMR
EMR containers	environment	AWS_ENDPOINT_URL_EMR_CONTAINERS
EMR Serverless	environment	AWS_ENDPOINT_URL_EMR_SERVERLESS

serviceId	Ki Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE> pe la ur fil be Al co
EntityResolution	er AWS_ENDPOINT_URL_ENTITYRESOLUTION o:
Elasticsearch Service	e: AWS_ENDPOINT_URL_ELASTICSEARCH_SERVICE a: i:
EventBridge	ev AWS_ENDPOINT_URL_EVENTBRIDGE g:
Evidently	ev AWS_ENDPOINT_URL_EVIDENTLY
finspace	f: AWS_ENDPOINT_URL_FINSPEACE
finspace data	f: AWS_ENDPOINT_URL_FINSPEACE_DATA d:
Firehose	f: AWS_ENDPOINT_URL_FIREHOSE
fis	f: AWS_ENDPOINT_URL_FIS
FMS	fr AWS_ENDPOINT_URL_FMS
forecast	fr AWS_ENDPOINT_URL_FORECAST
forecastquery	fr AWS_ENDPOINT_URL_FORECASTQUERY u:

serviceId	Kategori	Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE>
FraudDetector	f:	AWS_ENDPOINT_URL_FRAUDETECTOR
FreeTier	f:	AWS_ENDPOINT_URL_FREETIER
FSx	f:	AWS_ENDPOINT_URL_FSX
GameLift	g:	AWS_ENDPOINT_URL_GAMELIFT
Glacier	g:	AWS_ENDPOINT_URL_GLACIER
Global Accelerator	g:	AWS_ENDPOINT_URL_GLOBAL_ACCELERATOR
Glue	g:	AWS_ENDPOINT_URL_GLUE
grafana	g:	AWS_ENDPOINT_URL_GRAFANA
Greengrass	g:	AWS_ENDPOINT_URL_GREENGRASS
GreengrassV2	g:	AWS_ENDPOINT_URL_GREENGRASSV2
GroundStation	g:	AWS_ENDPOINT_URL_GROUNDSTATION
GuardDuty	g:	AWS_ENDPOINT_URL_GUARDDUTY

serviceId	Ki Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE> pe la ur fil be Al co
Health	he AWS_ENDPOINT_URL_HEALTH
HealthLake	he AWS_ENDPOINT_URL_HEALTHLAKE e
Honeycode	he AWS_ENDPOINT_URL_HONEYCODE
IAM	ia AWS_ENDPOINT_URL_IAM
identitystore	ia AWS_ENDPOINT_URL_IDENTITYSTORE te
imagebuilder	ir AWS_ENDPOINT_URL_IMAGEBUILDER de
ImportExport	ir AWS_ENDPOINT_URL_IMPORTEXPORT o:
Inspector	ii AWS_ENDPOINT_URL_INSPECTOR
Inspector Scan	ii AWS_ENDPOINT_URL_INSPECTOR_SCAN _:
Inspector2	ii AWS_ENDPOINT_URL_INSPECTOR2 2
InternetMonitor	ii AWS_ENDPOINT_URL_INTERNETMONITOR oi

serviceId	Kategori	Detail
	Kategori	Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE>
IoT	Platform	<code>AWS_ENDPOINT_URL_IOT</code>
IoT Data Plane	Platform	<code>AWS_ENDPOINT_URL_IOT_DATA_PLANE</code>
IoT Jobs Data Plane	Platform	<code>AWS_ENDPOINT_URL_IOT_JOBS_DATA_PLANE</code>
IoT 1Click Devices Service	Platform	<code>AWS_ENDPOINT_URL_IOT_1CLICK_DEVICES_SERVICE</code>
IoT 1Click Projects	Platform	<code>AWS_ENDPOINT_URL_IOT_1CLICK_PROJECTS</code>
IoTAnalytics	Platform	<code>AWS_ENDPOINT_URL_IOTANALYTICS</code>
IotDeviceAdvisor	Platform	<code>AWS_ENDPOINT_URL_IOTDEVICEADVISOR</code>
IoT Events	Platform	<code>AWS_ENDPOINT_URL_IOT_EVENTS</code>
IoT Events Data	Platform	<code>AWS_ENDPOINT_URL_IOT_EVENTS_DATA</code>

serviceId	Kategori Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE>
IoT FleetHub	iot AWS_ENDPOINT_URL_IOTFLEETHUB
IoT FleetWise	iot AWS_ENDPOINT_URL_IOTFLEETWISE
IoT SecureTunneling	iot AWS_ENDPOINT_URL_IOTSECURETUNNELING
IoT SiteWise	iot AWS_ENDPOINT_URL_IOTSITWISE
IoT ThingsGraph	iot AWS_ENDPOINT_URL_IOTTHINGSGRAPH
IoT TwinMaker	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

serviceId	Ki Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE>
KafkaConnect	k: AWS_ENDPOINT_URL_KAFKACONNECT
kendra	k: AWS_ENDPOINT_URL_KENDRA
Kendra Ranking	k: AWS_ENDPOINT_URL_KENDRA_RANKING
Keyspaces	k: AWS_ENDPOINT_URL_KEYSPACES
Kinesis	k: AWS_ENDPOINT_URL_KINESIS
Kinesis Video Archived Media	k: AWS_ENDPOINT_URL_KINESIS_VIDEO_ARCHIVED_MEDIA
Kinesis Video Media	k: AWS_ENDPOINT_URL_KINESIS_VIDEO_MEDIA
Kinesis Video Signaling	k: AWS_ENDPOINT_URL_KINESIS_VIDEO_SIGNALING

serviceId	Ki Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE>
Kinesis Video WebRTC Storage	k: AWS_ENDPOINT_URL_KINESIS_VIDEO_WEBRTC i: C_STORAGE t: e:
Kinesis Analytics	k: AWS_ENDPOINT_URL_KINESIS_ANALYTICS n:
Kinesis Analytics V2	k: AWS_ENDPOINT_URL_KINESIS_ANALYTICS_V2 n: v:
Kinesis Video	k: AWS_ENDPOINT_URL_KINESIS_VIDEO i:
KMS	kr: AWS_ENDPOINT_URL_KMS
LakeFormation	l: AWS_ENDPOINT_URL_LAKEFORMATION t:
Lambda	l: AWS_ENDPOINT_URL_LAMBDA
Launch Wizard	l: AWS_ENDPOINT_URL_LAUNCH_WIZARD z:
Lex Model Building Service	l: AWS_ENDPOINT_URL_LEX_MODEL_BUILDING_ _I SERVICE _S:

serviceId	Kategori	Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE>
Lex Runtime Service	Lex	AWS_ENDPOINT_URL_LEX_RUNTIME_SERVICE
Lex Models V2	Lex	AWS_ENDPOINT_URL_LEX_MODELS_V2
Lex Runtime V2	Lex	AWS_ENDPOINT_URL_LEX_RUNTIME_V2
License Manager	License Manager	AWS_ENDPOINT_URL_LICENSE_MANAGER
License Manager Linux Subscriptions	License Manager Linux Subscriptions	AWS_ENDPOINT_URL_LICENSE_MANAGER_LINUX_SUBSCRIPTIONS
License Manager User Subscriptions	License Manager User Subscriptions	AWS_ENDPOINT_URL_LICENSE_MANAGER_USER_SUBSCRIPTIONS
Lightsail	Lightsail	AWS_ENDPOINT_URL_LIGHTSAIL
Location	Location	AWS_ENDPOINT_URL_LOCATION

serviceId	Kategori	Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE>
CloudWatch Logs	cloudwatchlogs	AWS_ENDPOINT_URL_CLOUDWATCH_LOGS
CloudWatch Logs	cloudwatchlogs	AWS_ENDPOINT_URL_CLOUDWATCH_LOGS
LookoutEquipment	lookoutequipment	AWS_ENDPOINT_URL_LOOKOUTEQUIPMENT
LookoutMetrics	lookoutmetrics	AWS_ENDPOINT_URL_LOOKOUTMETRICS
LookoutVision	lookoutvision	AWS_ENDPOINT_URL_LOOKOUTVISION
m2	m2	AWS_ENDPOINT_URL_M2
Machine Learning	machinelearning	AWS_ENDPOINT_URL_MACHINE_LEARNING
Macie2	macie2	AWS_ENDPOINT_URL_MACIE2
ManagedBlockchain	managedblockchain	AWS_ENDPOINT_URL_MANAGEDBLOCKCHAIN
ManagedBlockchain Query	managedblockchainquery	AWS_ENDPOINT_URL_MANAGEDBLOCKCHAIN_QUERY

serviceId	Ki Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE> pe la ur fil be Al co
Marketplace Agreement	m: AWS_ENDPOINT_URL_MARKETPLACE_AGREEMENT C e
Marketplace Catalog	m: AWS_ENDPOINT_URL_MARKETPLACE_CATALOG C g
Marketplace Deployment	m: AWS_ENDPOINT_URL_MARKETPLACE_DEPLOYMENT C m
Marketplace Entitlement Service	m: AWS_ENDPOINT_URL_MARKETPLACE_ENTITLE C MENT_SERVICE e 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:

serviceId	Ki Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE> pe la ur fil be Al co
MediaLive	me AWS_ENDPOINT_URL_MEDIALIVE
MediaPackage	me AWS_ENDPOINT_URL_MEDIAPACKAGE a
MediaPackage Vod	me AWS_ENDPOINT_URL_MEDIAPACKAGE_VOD a
MediaPackageV2	me AWS_ENDPOINT_URL_MEDIAPACKAGEV2 a
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 m:
MemoryDB	me AWS_ENDPOINT_URL_MEMORYDB
Marketplace Metering	me AWS_ENDPOINT_URL_MARKETPLACE_METERING C n

serviceId	Ki Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE> pe la ur fil be Al co
Migration Hub	m: AWS_ENDPOINT_URL_MIGRATION_HUB _H
mgn	m: AWS_ENDPOINT_URL_MGN
Migration Hub Refactor Spaces	m: AWS_ENDPOINT_URL_MIGRATION_HUB_REFAC _H TOR_SPACES ct es
MigrationHub Config	m: AWS_ENDPOINT_URL_MIGRATIONHUB_CONFIG ht g
MigrationHubOrches trator	m: AWS_ENDPOINT_URL_MIGRATIONHUBORCHESTRATOR ht t:
MigrationHubStrategy	m: AWS_ENDPOINT_URL_MIGRATIONHUBSTRATEGY ht g)
Mobile	m: AWS_ENDPOINT_URL_MOBILE
mq	m: AWS_ENDPOINT_URL_MQ
MTurk	m: AWS_ENDPOINT_URL_MTURK
MWAA	m: AWS_ENDPOINT_URL_MWAA

serviceId	Kategori	Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE>
Neptune	ne	AWS_ENDPOINT_URL_NEPTUNE
Neptune Graph	ne	AWS_ENDPOINT_URL_NEPTUNE_GRAPH
neptunedata	ne	AWS_ENDPOINT_URL_NEPTUNEDATA
Network Firewall	ne	AWS_ENDPOINT_URL_NETWORK_FIREWALL
NetworkManager	ne	AWS_ENDPOINT_URL_NETWORKMANAGER
NetworkMonitor	ne	AWS_ENDPOINT_URL_NETWORKMONITOR
nimble	n:	AWS_ENDPOINT_URL_NIMBLE
OAM	o:	AWS_ENDPOINT_URL_OAM
Omics	or	AWS_ENDPOINT_URL_OMICS
OpenSearch	op	AWS_ENDPOINT_URL_OPENSEARCH
OpenSearchServerless	op	AWS_ENDPOINT_URL_OPENSEARCHSERVERLESS

serviceId	Kategori	Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE>
OpsWorks	ops	AWS_ENDPOINT_URL_OPSWORKS
OpsWorksCM	ops	AWS_ENDPOINT_URL_OPSWORKSCM
Organizations	org	AWS_ENDPOINT_URL_ORGANIZATIONS
OSIS	osis	AWS_ENDPOINT_URL_OSIS
Outposts	outposts	AWS_ENDPOINT_URL_OUTPOSTS
p8data	p8data	AWS_ENDPOINT_URL_P8DATA
p8data	p8data	AWS_ENDPOINT_URL_P8DATA
Panorama	panorama	AWS_ENDPOINT_URL_PANORAMA
Payment Cryptography	payment-cryptography	AWS_ENDPOINT_URL_PAYMENT_CRYPTOGRAPHY
Payment Cryptography Data	payment-cryptography-data	AWS_ENDPOINT_URL_PAYMENT_CRYPTOGRAPHY_DATA
Pca Connector Ad	pca-connector-ad	AWS_ENDPOINT_URL_PCA_CONNECTOR_AD

serviceId	Ki Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE> pe la ur fil be Al co
Personalize	pe AWS_ENDPOINT_URL_PERSONALIZE ze
Personalize Events	pe AWS_ENDPOINT_URL_PERSONALIZE_EVENTS ze
Personalize Runtime	pe AWS_ENDPOINT_URL_PERSONALIZE_RUNTIME ze e
PI	p: AWS_ENDPOINT_URL_PI
Pinpoint	p: AWS_ENDPOINT_URL_PINPOINT
Pinpoint Email	p: AWS_ENDPOINT_URL_PINPOINT_EMAIL er
Pinpoint SMS Voice	p: AWS_ENDPOINT_URL_PINPOINT_SMS_VOICE sr
Pinpoint SMS Voice V2	p: AWS_ENDPOINT_URL_PINPOINT_SMS_VOICE_V2 sr _\'
Pipes	p: AWS_ENDPOINT_URL_PIPES
Polly	pe AWS_ENDPOINT_URL_POLLY

serviceId	Ki Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE> pe la ur fil be Al co
Pricing	p: AWS_ENDPOINT_URL_PRICING
PrivateNetworks	p: AWS_ENDPOINT_URL_PRIVATENETWORKS tv
Proton	p: AWS_ENDPOINT_URL_PROTON
QBusiness	qb AWS_ENDPOINT_URL_QBUSINESS
QConnect	qc AWS_ENDPOINT_URL_QCONNECT
QLDB	q: AWS_ENDPOINT_URL_QLDB
QLDB Session	q: AWS_ENDPOINT_URL_QLDB_SESSION ic
QuickSight	qs AWS_ENDPOINT_URL_QUICKSIGHT t
RAM	ra: AWS_ENDPOINT_URL_RAM
rbin	rb: AWS_ENDPOINT_URL_RBIN
RDS	ra: AWS_ENDPOINT_URL_RDS
RDS Data	ra: AWS_ENDPOINT_URL_RDS_DATA
Redshift	ra: AWS_ENDPOINT_URL_REDSHIFT

serviceId	Kategori	Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE>
Redshift Data	Redshift Data	<code>AWS_ENDPOINT_URL_REDSHIFT_DATA</code>
Redshift Serverless	Redshift Serverless	<code>AWS_ENDPOINT_URL_REDSHIFT_SERVERLESS</code>
Rekognition	Rekognition	<code>AWS_ENDPOINT_URL_REKOGNITION</code>
repostspace	RepostSpace	<code>AWS_ENDPOINT_URL_REPOSTSPACE</code>
resiliencehub	Resilience Hub	<code>AWS_ENDPOINT_URL_RESILIENCEHUB</code>
Resource Explorer 2	Resource Explorer 2	<code>AWS_ENDPOINT_URL_RESOURCE_EXPLORER_2</code>
Resource Groups	Resource Groups	<code>AWS_ENDPOINT_URL_RESOURCE_GROUPS</code>
Resource Groups Tagging API	Resource Groups Tagging API	<code>AWS_ENDPOINT_URL_RESOURCE_GROUPS_TAGGING_API</code>

serviceId	Kategori	Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE>
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
Route53 Recovery Control Config	Region	AWS_ENDPOINT_URL_ROUTE53_RECOVERY_CONTROL_CONFIG
Route53 Recovery Readiness	Region	AWS_ENDPOINT_URL_ROUTE53_RECOVERY_READINESS
Route 53 Domains	Region	AWS_ENDPOINT_URL_ROUTE_53_DOMAINS
Route53Resolver	Region	AWS_ENDPOINT_URL_ROUTE53RESOLVER
RUM	Region	AWS_ENDPOINT_URL_RUM
S3	Static	AWS_ENDPOINT_URL_S3

serviceId	Ki Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE>
S3 Control	s: AWS_ENDPOINT_URL_S3_CONTROL
S3Outposts	s: AWS_ENDPOINT_URL_S3OUTPOSTS
SageMaker	s: AWS_ENDPOINT_URL_SAGEMAKER
SageMaker A2I Runtime	s: AWS_ENDPOINT_URL_SAGEMAKER_A2I_RUNTIME
Sagemaker Edge	s: AWS_ENDPOINT_URL_SAGEMAKER_EDGE
SageMaker FeatureStore Runtime	s: AWS_ENDPOINT_URL_SAGEMAKER_FEATURESTORE_RUNTIME
SageMaker Geospatial	s: AWS_ENDPOINT_URL_SAGEMAKER_GEOSPATIAL
SageMaker Metrics	s: AWS_ENDPOINT_URL_SAGEMAKER_METRICS

serviceId	Ki Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE> pe la ur fil be Al co
SageMaker Runtime	s: AWS_ENDPOINT_URL_SAGEMAKER_RUNTIME _:
savingsplans	s: AWS_ENDPOINT_URL_SAVINGSPLANS a:
Scheduler	s: AWS_ENDPOINT_URL_SCHEDULER
schemas	s: AWS_ENDPOINT_URL_SCHEMAS
SimpleDB	s: AWS_ENDPOINT_URL_SIMPLEDB
Secrets Manager	s: AWS_ENDPOINT_URL_SECRETS_MANAGER a:
SecurityHub	s: AWS_ENDPOINT_URL_SECURITYHUB ul
SecurityLake	s: AWS_ENDPOINT_URL_SECURITYLAKE al
ServerlessApplicationRepository	s: AWS_ENDPOINT_URL_SERVERLESSAPPLICATIONREPOSITORY s: ic t:
Service Quotas	s: AWS_ENDPOINT_URL_SERVICE_QUOTAS uc

serviceId	Ki	Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE>
Service Catalog	s:	AWS_ENDPOINT_URL_SERVICE_CATALOG
Service Catalog AppRegistry	s:	AWS_ENDPOINT_URL_SERVICE_CATALOG_APP_REGISTRY
ServiceDiscovery	s:	AWS_ENDPOINT_URL_SERVICEDISCOVERY
SES	s:	AWS_ENDPOINT_URL_SES
SESV2	s:	AWS_ENDPOINT_URL_SESV2
Shield	s:	AWS_ENDPOINT_URL_SHIELD
signer	s:	AWS_ENDPOINT_URL_SIGNER
SimSpaceWeaver	s:	AWS_ENDPOINT_URL_SIMSPACEWEAVER
SMS	s:	AWS_ENDPOINT_URL_SMS
Snow Device Management	s:	AWS_ENDPOINT_URL_SNOW_DEVICE_MANAGEMENT
Snowball	s:	AWS_ENDPOINT_URL_SNOWBALL

serviceId	Kategori	Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE>
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
Ssm Sap	Service	AWS_ENDPOINT_URL_SSM_SAP
SSO	Service	AWS_ENDPOINT_URL_SSO
SSO Admin	Service	AWS_ENDPOINT_URL_SSO_ADMIN
SSO OIDC	Service	AWS_ENDPOINT_URL_SSO_OIDC
SFN	Service	AWS_ENDPOINT_URL_SFN
Storage Gateway	Service	AWS_ENDPOINT_URL_STORAGE_GATEWAY
STS	Service	AWS_ENDPOINT_URL_STS
SupplyChain	Service	AWS_ENDPOINT_URL_SUPPLYCHAIN

serviceId	Ki Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE> pe la ur fil be Al co
Support	si AWS_ENDPOINT_URL_SUPPORT
Support App	si AWS_ENDPOINT_URL_SUPPORT_APP PI
SWF	si AWS_ENDPOINT_URL_SWF
synthetics	sy AWS_ENDPOINT_URL_SYNTHETICS S
Textract	te AWS_ENDPOINT_URL_TEXTRACT
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	ti AWS_ENDPOINT_URL_TNB
Transcribe	t: AWS_ENDPOINT_URL_TRANSCRIBE e
Transfer	t: AWS_ENDPOINT_URL_TRANSFER

serviceId	Ki Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE> pe la ur fil be Al co
Translate	t: AWS_ENDPOINT_URL_TRANSLATE
TrustedAdvisor	t: AWS_ENDPOINT_URL_TRUSTEDADVISOR v:
VerifiedPermissions	v: AWS_ENDPOINT_URL_VERIFIEDPERMISSIONS e: s
Voice ID	v: AWS_ENDPOINT_URL_VOICE_ID
VPC Lattice	v: AWS_ENDPOINT_URL_VPC_LATTICE c:
WAF	w: AWS_ENDPOINT_URL_WAF
WAF Regional	w: AWS_ENDPOINT_URL_WAF_REGIONAL n:
WAFV2	w: AWS_ENDPOINT_URL_WAFV2
WellArchitected	w: AWS_ENDPOINT_URL_WELLARCHITECTED t:
Wisdom	w: AWS_ENDPOINT_URL_WISDOM
WorkDocs	w: AWS_ENDPOINT_URL_WORKDOCS
WorkLink	w: AWS_ENDPOINT_URL_WORKLINK

serviceId	Ki Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE> pe la ur fil be Al co
WorkMail	wc AWS_ENDPOINT_URL_WORKMAIL
WorkMailMessageFlow	wc AWS_ENDPOINT_URL_WORKMAILMESSAGEFLOW e: w
WorkSpaces	wc AWS_ENDPOINT_URL_WORKSPACES S
WorkSpaces Thin Client	wc AWS_ENDPOINT_URL_WORKSPACES_THIN_CLIENT S_ ie
WorkSpaces Web	wc AWS_ENDPOINT_URL_WORKSPACES_WEB S_
XRay	x: AWS_ENDPOINT_URL_XRAY

Default konfigurasi cerdas

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

Konfigurasi fungsi ini dengan menggunakan yang berikut ini:

defaults_mode- Pengaturan AWS **config** file bersama, **AWS_DEFAULTS_MODE**- variabel lingkungan, **aws.defaultsMode**- Properti sistem JVM: 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 setelan

AWS SDK memiliki nilai yang ditetapkan secara eksplisit, maka nilai itu selalu diutamakan. Jika setelan AWS SDK 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-hal berikut: Setelan komunikasi HTTP, perilaku coba lagi, pengaturan titik akhir Regional layanan, dan, berpotensi, konfigurasi terkait SDK 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 perkembangan praktik terbaik.

Nilai default: `legacy`

Catatan: Versi utama SDK yang baru akan menjadi default. `standard`

Nilai yang valid:

- `legacy`— Menyediakan pengaturan default yang bervariasi menurut SDK dan ada sebelum pembentukan. `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 Wilayah 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 mencoba 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 dan data pengguna](#), 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 mencoba ulang. Lihat [Coba lagi perilaku](#).
- `stsRegionalEndpoints`— Menentukan bagaimana SDK menentukan Layanan AWS titik akhir yang digunakan untuk berbicara dengan (). AWS Security Token Service AWS STS Lihat [AWS STS Titik akhir regional](#).
- `s3UsEast1RegionalEndpoints`— Menentukan cara SDK menentukan titik akhir AWS layanan yang digunakannya untuk berbicara dengan Amazon S3 untuk Wilayah tersebut. `us-east-1`
- `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 jabat tangan TLS dari saat pesan CLIENT HELLO dikirim ke waktu klien dan server telah sepenuhnya menegosiasikan cipher 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>	<code>standard</code>	<code>standard</code>	<code>standard</code>	<code>standard</code>
<code>stsRegionalEndpoints</code>	<code>regional</code>	<code>regional</code>	<code>regional</code>	<code>regional</code>
<code>s3UsEast1RegionalEndpoints</code>	<code>regional</code>	<code>regional</code>	<code>regional</code>	<code>regional</code>
<code>connectTimeoutInMillis</code>	3100	1100	3100	30000

Parameter	standard modus	in-region modus	cross-reg ion modus	mobile modus
<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 SDK

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

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

SDK	Didukung	Catatan atau informasi lebih lanjut
SDK for Java 2.x	Ya	Parameter tidak dioptimalkan: <code>stsRegionalEndpoints</code> .
SDK for Java 1.x	Tidak	
SDK untuk 3.x JavaScript	Ya	Parameter tidak dioptimalkan: <code>stsRegionalEndpoints</code> , <code>s3UsEast1RegionalEndpoints</code> , <code>tlsNegotiationTimeoutInMillis</code> . <code>connectTimeoutInMillis</code> disebut <code>connectionTimeout</code> .
SDK untuk 2.x JavaScript	Tidak	
SDK para Kotlin	Tidak	
SDK for .NET 3.x	Ya	Parameter tidak dioptimalkan: <code>connectTimeoutInMillis</code> , <code>tlsNegotiationTimeoutInMillis</code> .
SDK for PHP 3.x	Ya	Parameter tidak dioptimalkan: <code>tlsNegotiationTimeoutInMillis</code> .
SDK untuk Python (Boto3)	Ya	Parameter tidak dioptimalkan: <code>tlsNegotiationTimeoutInMillis</code> .
SDK for Ruby 3.x	Ya	

SDK	Didukung	Catatan atau informasi lebih lanjut
SDK untuk Rust	Tidak	
Alat untuk PowerShell	Ya	Parameter tidak dioptimalkan:connectTimeoutInMillis ,tlsNegotiationTimeoutInMillis .

AWSPustaka Runtime Umum (CRT)

Pustaka AWS Common Runtime (CRT) adalah pustaka dasar SDK. CRT 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 SDK yang menyediakan penggunaan kembali, pengoptimalan, dan akurasi kode yang lebih baik. Paket-paketnya adalah:

- [awslabs/aws-c-auth](#): otentikasi AWS sisi klien (penyedia kredensial standar dan penandatanganan (sigv4))
- [awslabs/aws-c-cal](#): Jenis primitif kriptografi, hash (MD5, SHA256, SHA256 HMAC), 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 (RPC) melalui aliran acara
- [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 library C99 untuk berkomunikasi dengan layanan Amazon S3, dirancang untuk memaksimalkan throughput pada instans Amazon EC2 bandwidth tinggi
- [awslabs/aws-c-sdkutils](#): Pustaka utilitas untuk mengurai dan mengelola profil AWS
- [awslabs/aws-checksums](#): CRC32c dan CRC32 yang dipercepat perangkat keras lintas platform dengan mundur ke implementasi perangkat lunak yang efisien
- [awslabs/aws-lc](#): Perpustakaan kriptografi tujuan umum yang dikelola oleh tim AWS Cryptography untuk AWS dan pelanggan mereka, berdasarkan kode dari proyek Google BoringSSL dan proyek OpenSSL
- [awslabs/s2n](#): Implementasi C99 protokol TLS/SSL, dirancang agar kecil dan cepat dengan keamanan sebagai prioritas

CRT tersedia melalui semua SDK kecuali Go.

Ketergantungan CRT

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

Berikut ini adalah ilustrasi dependensi hierarkis dari pustaka 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 pemeliharannya, lihat. [Matriks dukungan versi](#)

AWS Matriks dukungan versi SDK dan Tools

Matriks di bawah ini menunjukkan daftar versi utama AWS Software Development Kit (SDK) yang tersedia dan di mana mereka berada dalam siklus hidup pemeliharaan dengan jadwal terkait. Untuk informasi rinci tentang siklus hidup untuk versi utama AWS SDK dan Tools serta 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	
SDK for C++	1.x	Ketersediaan Umum	9/2/2015	
SDK for Go V2	V2 1.x	Ketersediaan Umum	1/19/2021	
SDK for Go	1.x	Pengumuman Pemeliharaan	11/19/2015	Lihat pengumuman untuk detail dan tanggal
SDK for Java	1.x	Pengumuman Pemeliharaan	25/03/2010	Lihat pengumuman untuk detail dan tanggal
SDK for Java	2.x	Ketersediaan Umum	11/20/2018	
SDK untuk JavaScript	1.x	Akhir dukungan	5/6/2013	

SDK	Versi mayor	Fase Saat Ini	Tanggal Ketersediaan Umum	Catatan
SDK untuk JavaScript	2.x	Pengumuman Pemeliharaan	6/19/2014	Lihat pengumuman untuk detail dan tanggal
SDK untuk JavaScript	3.x	Ketersediaan Umum	12/15/2020	
SDK para Kotlin	1.x	Ketersediaan Umum	11/27/2023	
SDK for .NET	1.x	Akhir dukungan	11/2009	
SDK for .NET	2.x	Akhir dukungan	11/8/2013	
SDK for .NET	3.x	Ketersediaan Umum	7/28/2015	
SDK for PHP	2.x	Akhir dukungan	11/2/2012	
SDK for 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 for Ruby	1.x	Akhir dukungan	7/14/2011	
SDK for Ruby	2.x	Akhir dukungan	2/15/2015	

SDK	Versi mayor	Fase Saat Ini	Tanggal Ketersediaan Umum	Catatan
SDK for Ruby	3.x	Ketersediaan Umum	29/8/2017	
SDK untuk Rust	1.x	Ketersediaan Umum	11/27/2023	
SDK para Swift	1.x	Pratinjau Pengembang		
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	

Riwayat dokumen untuk AWS SDK dan Panduan Referensi Alat

Tabel berikut menjelaskan penambahan dan pembaruan penting pada AWS SDK dan Panduan Referensi Alat. Untuk notifikasi tentang pembaruan-pembaruan dokumentasi ini, Anda dapat berlangganan ke sebuah umpan RSS.

Perubahan	Deskripsi	Tanggal
SDK for Java 1.x properti sistem	Tambahkan detail tentang pengaturan konfigurasi sistem JVM yang didukung oleh 1.x. AWS SDK for Java	30 Mei 2024
Pembaruan pengaturan	Tambahkan pengaturan konfigurasi sistem JVM.	Maret 27, 2024
Pembaruan tabel kompatibilitas	Pembaruan kompatibilitas untuk dukungan SDK, pembaruan prosedur Pusat Identitas IAM.	Februari 20, 2024
Pembaruan kredensi kontainer . Pembaruan IMDS.	Menambahkan dukungan untuk Amazon EKS. Menambahkan pengaturan untuk menonaktifkan fallback IMDSv1.	Desember 29, 2023
Minta kompresi	Menambahkan pengaturan untuk fitur kompresi permintaan.	27 Desember 2023
Tabel kompatibilitas	Tabel kompatibilitas untuk SDK dan fitur alat diperbarui untuk menyertakan SDK untuk Kotlin, SDK untuk Rust, dan AWS Tools for PowerShell	Desember 10, 2023

Pembaruan otentikasi	Pembaruan untuk metode otentikasi yang didukung untuk SDK dan alat.	Juli 1, 2023
Pembaruan praktik terbaik IAM	Panduan yang diperbarui untuk menyelaraskan dengan praktik terbaik IAM. Untuk informasi selengkapnya, lihat Praktik terbaik keamanan di IAM .	27 Februari 2023
Pembaruan SSO	Pembaruan kredensial SSO untuk konfigurasi token SSO 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 klien IMDS dan kredensial IMDS. 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 bahwa 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.