



Panduan Referensi

AWS SDKs dan Tools



AWS SDKs dan Tools: Panduan Referensi

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

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

Table of Contents

AWS SDKs dan Panduan Referensi Alat	1
Sumber daya pengembang	3
Pemberitahuan telemetri toolkit	3
Konfigurasi	4
Berbagi config dan credentials file	5
Profil	5
Format file konfigurasi	7
Format file kredensial	10
Lokasi file bersama	11
Resolusi direktori rumah	11
Ubah lokasi default file-file ini	12
Variabel-variabel lingkungan	13
Cara mengatur variabel lingkungan	13
Pengaturan variabel lingkungan tanpa server	15
Properti sistem JVM	15
Cara mengatur properti sistem JVM	16
Otentikasi dan akses	18
Pilih metode untuk mengautentikasi kode aplikasi Anda	18
Metode otentikasi	22
ID AWS Builder	24
Login menggunakan kredensial konsol	24
Cara kerjanya	25
Autentikasi Pusat Identitas IAM	25
Prasyarat	26
Konfigurasi akses terprogram menggunakan IAM Identity Center	26
Menyegarkan sesi akses portal	29
Memahami autentikasi Pusat Identitas IAM	30
IAM Roles Anywhere	33
Langkah 1: Konfigurasi Peran IAM Di Mana Saja	34
Langkah 2: Gunakan Peran IAM Di Mana Saja	34
Asumsikan peran	35
Asumsikan peran IAM	36
Asumsikan peran (web)	38
Bersekutu dengan identitas web atau OpenID Connect	38

AWS kunci akses	40
Gunakan kredensial jangka pendek	40
Gunakan kredensi jangka panjang	40
Kredensi jangka pendek	42
Kredensi jangka panjang	43
Peran IAM untuk instance EC2	47
Membuat peran IAM	47
Luncurkan EC2 instans Amazon dan tentukan peran IAM Anda	47
Connect ke EC2 instance	48
Jalankan aplikasi Anda pada EC2 instance	48
Propagasi identitas tepercaya	49
Prasyarat untuk menggunakan plugin TIP	50
Untuk menggunakan plugin TIP dalam kode Anda	50
Contoh kode menggunakan TIP	53
Referensi pengaturan	60
Membuat klien layanan	60
Prioritas pengaturan	60
Memahami halaman pengaturan panduan ini	61
Configdaftar pengaturan file	63
Credentialsdaftar pengaturan file	67
Daftar variabel lingkungan	68
Daftar properti sistem JVM	73
Penyedia kredensi standar	76
Memahami rantai penyedia kredensi	77
Rantai penyedia kredensi khusus SDK dan khusus alat	79
AWS kunci akses	79
Penyedia login	82
Asumsikan penyedia peran	85
Penyedia kontainer	92
Penyedia Pusat Identitas IAM	96
Penyedia IMDS	102
Penyedia proses	107
Fitur standar	111
Titik akhir berbasis akun	113
ID Aplikasi	115
EC2 Metadata contoh Amazon	118

Titik akses Amazon S3	120
Titik Akses Multi-Wilayah Amazon S3	122
Otentikasi sesi S3 Express One Zone	125
Skema otentikasi	128
Wilayah AWS	130
AWS STS Titik akhir regional	134
Perlindungan Integritas Data	139
Dual-stack dan titik akhir FIPS	144
Penemuan titik akhir	147
Konfigurasi umum	149
Injeksi awalan host	153
Klien IMDS	157
Coba lagi perilaku	160
Minta kompresi	167
Titik akhir khusus layanan	169
Default konfigurasi cerdas	213
Runtime Umum	219
Ketergantungan CRT	220
Kebijakan pemeliharaan	221
Ikhtisar	221
Penentuan versi	221
Siklus hidup versi utama SDK	221
Siklus hidup ketergantungan	222
Metode komunikasi	223
Siklus hidup versi	224
Riwayat dokumen	227
.....	CCXXX

Apa yang tercakup dalam Panduan Referensi Alat AWS SDKs dan

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

Panduan ini mencakup informasi mengenai:

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

Panduan Referensi Alat AWS SDKs dan Alat ini dimaksudkan untuk menjadi basis informasi yang berlaku untuk beberapa SDKs dan alat. Panduan khusus untuk SDK atau alat yang Anda gunakan harus digunakan selain informasi 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 SDKs dan kebijakan pemeliharaan alat
<ul style="list-style-type: none"> • AWS Cloud9 • AWS CDK • AWS Toolkit for Azure DevOps • AWS Toolkit for JetBrains 	Konfigurasi AWS SDKs dan alat secara global Otentikasi dan akses menggunakan AWS SDKs dan alat AWS SDKs dan kebijakan pemeliharaan alat

Jika Anda menggunakan:	Bagian yang relevan dari panduan ini untuk Anda adalah:
<ul style="list-style-type: none"> • AWS Toolkit for Visual Studio • AWS Toolkit for Visual Studio Code • AWS Serverless Application Model • AWS CodeArtifact • AWS CodeBuild • Amazon CodeCatalyst • AWS CodeCommit • AWS CodeDeploy • AWS CodePipeline 	
<ul style="list-style-type: none"> • AWS CLI • AWS SDK untuk C++ • AWS SDK untuk Go • AWS SDK untuk Java • AWS SDK untuk JavaScript • AWS SDK untuk Kotlin • AWS SDK untuk .NET • AWS SDK untuk PHP • AWS SDK untuk Python (Boto3) • AWS SDK untuk Ruby • AWS SDK untuk Rust • AWS SDK untuk Swift • AWS Tools for Windows PowerShell 	<ul style="list-style-type: none"> • Konfigurasi AWS SDKs dan alat secara global • Otentikasi dan akses menggunakan AWS SDKs dan alat • AWS SDKs dan referensi pengaturan alat • AWS Pustaka Runtime Umum (CRT) • AWS SDKs dan kebijakan pemeliharaan alat • AWS SDKs dan siklus hidup versi Tools

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

Sumber daya pengembang

Amazon Q Developer adalah asisten percakapan bertenaga AI generatif yang dapat membantu Anda memahami, membangun, memperluas, dan mengoperasikan aplikasi. AWS Untuk mempercepat pembangunan Anda AWS, model yang mendukung Amazon Q ditambah dengan AWS konten berkualitas tinggi untuk menghasilkan jawaban yang lebih lengkap, dapat ditindaklanjuti, dan direferensikan. Untuk informasi selengkapnya, lihat [Apa itu Pengembang Amazon Q?](#) di Panduan Pengguna Pengembang Amazon Q.

Pemberitahuan telemetri toolkit

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

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

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

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

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

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

Konfigurasi AWS SDKs dan alat secara global

Dengan AWS SDKs dan alat AWS pengembang lainnya, seperti AWS Command Line Interface (AWS CLI), Anda dapat berinteraksi dengan AWS layanan APIs. 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, alat AWS SDKs dan mendukung pengaturan umum dari sumber utama di luar kode itu sendiri. Bagian ini mencakup topik-topik berikut:

Topik

- [Menggunakan credentials file bersama config dan untuk mengkonfigurasi AWS SDKs dan alat secara global](#)
- [Menemukan dan mengubah lokasi berbagi config dan credentials file AWS SDKs dan alat](#)
- [Menggunakan variabel lingkungan untuk mengkonfigurasi AWS SDKs dan alat secara global](#)
- [Menggunakan properti sistem JVM untuk mengkonfigurasi dan AWS SDK untuk Java AWS SDK untuk Kotlin](#)

Menggunakan **credentials** file bersama **config** dan untuk mengkonfigurasi AWS SDKs dan alat secara global

File bersama AWS config dan credentials file adalah cara paling umum yang dapat Anda tentukan otentikasi dan konfigurasi ke AWS SDK atau alat.

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

Note

Jika aplikasi Anda berada di server yang menjalankan beberapa aplikasi, kami sarankan Anda selalu menggunakan profil bernama daripada profil default. Profil default secara otomatis diambil oleh AWS aplikasi apa pun di lingkungan dan dibagikan di antara mereka. Jadi, jika orang lain memperbarui profil default untuk aplikasi mereka, itu dapat secara tidak sengaja memengaruhi yang lain. Untuk melindungi terhadap hal ini, tentukan profil bernama dalam config file bersama dan kemudian gunakan profil bernama itu di aplikasi Anda dengan menyetel profil bernama dalam kode Anda. Anda dapat menggunakan variabel

lingkungan atau properti sistem JVM untuk mengatur profil bernama jika Anda tahu bahwa cakupannya hanya memengaruhi aplikasi Anda.

Format file konfigurasi

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

`configFile` tersebut adalah file plaintext yang menggunakan format berikut:

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

Jenis bagian

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

Jenis bagian: **default**

Contoh baris definisi bagian: `[default]`

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

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

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

Jenis bagian: **profile**

Contoh baris definisi bagian: `[profile dev]`

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

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

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

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

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

Jenis bagian: **sso-session**

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

Baris definisi `sso-session` bagian menamai sekelompok pengaturan yang Anda gunakan untuk mengonfigurasi profil untuk menyelesaikan AWS kredensial yang digunakan. AWS IAM Identity Center Untuk informasi selengkapnya tentang mengonfigurasi autentikasi masuk tunggal, lihat [Menggunakan IAM Identity Center untuk mengautentikasi AWS SDK dan alat](#) 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 SDKs dan alat yang menyertakan fitur ini. Untuk melihat apakah fitur ini tersedia untuk SDK Anda, lihat titik akhir [Support oleh AWS SDKs dan alat](#) 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]
```


Menemukan dan mengubah lokasi berbagi **config** dan **credentials** file AWS SDKs dan alat

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

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

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

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

Resolusi direktori rumah

`~` hanya digunakan untuk resolusi direktori home ketika:

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

Saat menentukan direktori home, variabel-variabel berikut diperiksa:

- (Semua platform) Variabel `HOME` lingkungan
- (Platform Windows) Variabel `USERPROFILE` lingkungan

- (Platform Windows) Penggabungan HOMEDRIVE dan variabel HOMEPATH lingkungan ()
\$HOMEDRIVE\$HOMEPATH
- (Opsional per SDK atau alat) SDK atau fungsi resolusi home path khusus alat atau variabel

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

Ubah lokasi default file-file ini

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

Menggunakan variabel lingkungan

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

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

Linux/macOS

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

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

Windows

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

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

Untuk informasi selengkapnya tentang mengonfigurasi sistem Anda menggunakan variabel lingkungan, lihat [Menggunakan variabel lingkungan untuk mengkonfigurasi AWS SDKs dan alat secara global](#).

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**
- credentialsvariabel 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.

Menggunakan variabel lingkungan untuk mengkonfigurasi AWS SDKs dan alat secara global

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

Prioritas opsi

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

Cara mengatur variabel lingkungan

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

Linux, macOS, or Unix

```
$ export AWS_ACCESS_KEY_ID=AKIAIOSFODNN7EXAMPLE
$ export AWS_SECRET_ACCESS_KEY=wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
$ export
AWS_SESSION_TOKEN=AQoEXAMPLEH4aoAH0gNCAPy...truncated...zrkuWJ0gQs8IZZaIv2BXIa2R40Lgk
$ export AWS_REGION=us-west-2
```

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

Windows Command Prompt

```
C:\> setx AWS_ACCESS_KEY_ID AKIAIOSFODNN7EXAMPLE
C:\> setx AWS_SECRET_ACCESS_KEY wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
C:\> setx
AWS_SESSION_TOKEN AQoEXAMPLEH4aoAH0gNCAPy...truncated...zrkuWJ0gQs8IZZaIv2BXIa2R40Lgk
C:\> setx AWS_REGION us-west-2
```

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

PowerShell

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

Jika Anda menetapkan variabel lingkungan pada PowerShell prompt seperti yang ditunjukkan pada contoh sebelumnya, itu menyimpan nilai hanya untuk durasi sesi saat ini. Untuk membuat pengaturan variabel lingkungan persisten di semua sesi PowerShell Command Prompt, simpan dengan menggunakan aplikasi Sistem di Control Panel. Atau, Anda dapat mengatur variabel untuk semua PowerShell sesi future dengan menambahkannya ke PowerShell profil Anda. Lihat

[PowerShell dokumentasi](#) untuk informasi selengkapnya tentang menyimpan variabel lingkungan atau menyimpannya 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 lihat dokumentasi resmi untuk wadah yang Anda gunakan untuk menentukan bagaimana variabel lingkungan diperlakukan dan apakah ada batasan.

Menggunakan properti sistem JVM untuk mengkonfigurasi dan AWS SDK untuk Java AWS SDK untuk Kotlin

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

Presedensi 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 menggunakan AWS SDKs dan alat

Saat Anda mengembangkan aplikasi AWS SDK atau menggunakan AWS alat untuk digunakan Layanan AWS, Anda harus menetapkan bagaimana kode atau alat Anda mengotentikasi. AWS Anda dapat mengonfigurasi akses terprogram ke AWS sumber daya dengan cara yang berbeda, tergantung pada lingkungan tempat kode berjalan dan AWS akses yang tersedia untuk Anda.

Opsi di bawah ini adalah bagian dari [rantai penyedia kredensi](#). Ini berarti bahwa dengan mengonfigurasi file bersama AWS config dan credentials file yang sesuai, AWS SDK atau alat Anda akan secara otomatis menemukan dan menggunakan metode otentikasi tersebut.

Pilih metode untuk mengotentikasi kode aplikasi Anda

Pilih metode untuk mengotentikasi panggilan yang dilakukan AWS oleh aplikasi Anda.

Apakah Anda menjalankan kode DI DALAM Layanan AWS (seperti Amazon EC2, Lambda, Amazon ECS, Amazon EKS,)? CodeBuild

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

Apakah kode Anda ada dalam instans Amazon Elastic Compute Cloud?

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

Apakah kode Anda dalam suatu AWS Lambda fungsi?

Lambda membuat peran eksekusi dengan izin minimal saat Anda membuat fungsi [Lambda](#). AWS SDK atau alat kemudian secara otomatis menggunakan peran IAM yang dilampirkan ke Lambda saat runtime, melalui lingkungan eksekusi Lambda.

Apakah kode Anda di Amazon Elastic Container Service (di Amazon EC2 atau untuk AWS Fargate Amazon ECS)?

Gunakan Peran IAM untuk Tugas. Anda harus [membuat peran tugas dan menentukan peran](#) itu dalam [definisi tugas Amazon ECS](#) Anda. AWS SDK atau alat kemudian secara otomatis menggunakan peran IAM yang ditetapkan ke tugas saat runtime, melalui metadata Amazon ECS.

Apakah kode Anda ada di Amazon Elastic Kubernetes Service?

Kami menyarankan Anda menggunakan [Amazon EKS Pod Identities](#).

Catatan: Jika Anda merasa bahwa [peran IAM untuk akun layanan](#) (IRSA) mungkin lebih sesuai dengan kebutuhan unik Anda, lihat [Membandingkan Identitas Pod EKS dan IRSA di Panduan Pengguna Amazon EKS](#).

Apakah kode Anda berjalan di AWS CodeBuild

Lihat [Menggunakan kebijakan berbasis identitas](#) untuk CodeBuild

Apakah kode Anda di yang lain Layanan AWS?

Lihat panduan khusus untuk Anda Layanan AWS. Saat Anda menjalankan kode AWS, [rantai penyedia kredensi](#) SDK dapat secara otomatis memperoleh dan menyegarkan kredensial untuk Anda.

Apakah Anda membuat aplikasi seluler atau aplikasi web 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 lain yang kompatibel dengan OpenID Connect (OIDC). 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 [Dengan asumsi peran dengan identitas web atau OpenID Connect AWS SDKs untuk mengautentikasi dan alat](#).

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.

Apakah Anda mengembangkan dan menjalankan kode secara lokal?

Kami merekomendasikan [Menggunakan kredensi konsol untuk AWS SDKs mengautentikasi dan alat](#).

Setelah alur otentikasi berbasis browser yang cepat, AWS secara otomatis menghasilkan kredensial sementara yang berfungsi di seluruh alat pengembangan lokal seperti CLI, dan AWS Alat AWS untuk PowerShell AWS SDKs

Jika Anda menggunakan Pusat Identitas untuk akses AWS akun

Gunakan Pusat Identitas IAM untuk mengautentikasi AWS SDK dan alat jika Anda sudah memiliki akses ke AWS akun yang and/or perlu mengelola akses untuk tenaga kerja Anda. Sebagai praktik keamanan terbaik, sebaiknya gunakan AWS Organizations dengan IAM Identity Center untuk mengelola akses di semua AWS akun Anda. Anda dapat membuat pengguna di Pusat Identitas IAM, 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 titik akhir dan kuota Pusat [Menggunakan IAM Identity Center untuk mengautentikasi AWS SDK dan alat](#) Identitas IAM di Referensi Umum Amazon Web Services.

Jika Anda mencari cara lain untuk mengautentikasi

Buat pengguna IAM dengan hak istimewa dengan izin untuk masuk ke peran target Anda. `sts:AssumeRole` Kemudian konfigurasi profil Anda untuk mengambil peran menggunakan `source_profile` pengaturan untuk pengguna tersebut.

Anda juga dapat menggunakan kredensial IAM sementara melalui variabel lingkungan atau file kredensi bersama AWS . Lihat [Menggunakan kredensi jangka pendek untuk mengautentikasi AWS SDKs dan alat](#).

Catatan: Hanya di kotak pasir atau lingkungan belajar, Anda dapat mempertimbangkan [Menggunakan kredensi jangka panjang untuk AWS SDKs mengautentikasi dan alat](#).

Apakah kode ini berjalan di lokasi atau dalam VM hibrid/sesuai permintaan (seperti server yang membaca dari atau menulis ke Amazon S3, atau Jenkins menyebarkan ke cloud)?

Apakah Anda menggunakan sertifikat klien X.509?

Ya: Lihat [Menggunakan Peran IAM Di Mana Saja untuk mengautentikasi dan AWS SDKs alat](#). 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.

Dapatkan lingkungan terhubung dengan aman ke penyedia identitas federasi (seperti Microsoft Entra atau Okta) untuk meminta kredensial sementara? AWS

Ya: Gunakan [Penyedia kredensi proses](#)

Gunakan [Penyedia kredensi proses](#) untuk mengambil kredensial secara otomatis saat runtime. Sistem ini mungkin menggunakan alat pembantu atau plugin untuk mendapatkan kredensial, dan mungkin mengambil peran IAM di belakang layar menggunakan `sts:AssumeRole`

Tidak: Gunakan kredensial sementara yang disuntikkan melalui AWS Secrets Manager

Gunakan kredensial sementara yang disuntikkan melalui AWS Secrets Manager Untuk opsi untuk mendapatkan kunci akses berumur pendek, lihat [Meminta kredensial keamanan sementara di Panduan Pengguna](#) IAM. Untuk opsi menyimpan kredensial sementara ini, lihat [AWS kunci akses](#)

Anda dapat menggunakan kredensial ini untuk mengambil izin aplikasi yang lebih luas dari Secrets Manager dengan aman, [tempat rahasia produksi atau kredensial](#) berbasis peran yang berumur panjang dapat disimpan.

Apakah Anda menggunakan alat pihak ketiga yang tidak masuk AWS?

Gunakan dokumentasi yang ditulis oleh penyedia pihak ketiga Anda untuk panduan terbaik tentang mendapatkan kredensial.

Jika pihak ketiga Anda belum memberikan dokumentasi, apakah Anda menyuntikkan kredensial sementara dengan aman?

Ya: Gunakan variabel lingkungan dan AWS STS kredensial sementara.

Tidak: Gunakan kunci akses statis yang disimpan di manajer rahasia terenkripsi (pilihan terakhir).

Metode otentikasi

Metode otentikasi untuk kode yang berjalan dalam lingkungan AWS

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

- [Menggunakan peran IAM untuk mengautentikasi aplikasi yang digunakan ke 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.
 - [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 lain yang kompatibel dengan OpenID Connect (OIDC). 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 [Dengan asumsi peran dengan identitas web atau OpenID Connect AWS SDKs untuk mengautentikasi dan alat](#).

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.

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

- [Menggunakan kredensi konsol untuk AWS SDKs mengautentikasi dan alat](#)— Fitur ini bekerja dengan Antarmuka Baris AWS Perintah dan Alat untuk PowerShell dan memberi Anda kredensial yang dapat disegarkan yang bekerja di seluruh alat pengembangan lokal seperti AWS CLI, Alat untuk dan. PowerShell AWS
- [Menggunakan IAM Identity Center untuk mengautentikasi AWS SDK dan alat](#)— Sebagai praktik keamanan terbaik, kami sarankan menggunakan AWS Organizations dengan IAM Identity Center untuk mengelola akses di semua Anda 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 Services
- [Menggunakan Peran IAM Di Mana Saja untuk mengautentikasi dan AWS SDKs alat](#)— 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.
- [Dengan asumsi peran dengan AWS kredensi untuk mengautentikasi dan alat AWS SDKs](#)— Anda dapat mengambil peran IAM untuk mengakses AWS sumber daya sementara yang mungkin tidak dapat Anda akses sebaliknya.
- [Menggunakan tombol AWS akses untuk mengautentikasi AWS SDKs dan alat](#)— Pilihan lain yang mungkin kurang nyaman atau mungkin meningkatkan risiko keamanan untuk AWS sumber daya Anda.

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 Services memiliki dasar-dasar dasar sebagai berikut:

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

Plugin IAM Identity Center terpercaya propagasi identitas (TIP) untuk diakses Layanan AWS

- [Menggunakan plugin TIP untuk mengakses Layanan AWS](#)— Jika Anda membuat aplikasi untuk Amazon Q Business atau layanan lain yang mendukung propagasi identitas terpercaya, dan menggunakan AWS SDK untuk Java atau AWS SDK untuk JavaScript, Anda dapat menggunakan plugin TIP untuk pengalaman otorisasi yang efisien.

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 Q dan Amazon CodeCatalyst.

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

Menggunakan kredensi konsol untuk AWS SDKs mengautentikasi dan alat

Menggunakan kredensial konsol adalah metode yang disarankan untuk memberikan AWS kredensial saat mengembangkan AWS aplikasi di lingkungan lokal Anda atau lingkungan layanan AWS non-komputasi lainnya. Jika Anda mengembangkan AWS sumber daya, seperti Amazon Elastic Compute Cloud (Amazon EC2) AWS CloudShell atau, sebaiknya Anda mendapatkan kredensial dari layanan tersebut.

Anda juga dapat melakukan autentikasi melalui IAM Identity Center. [Menggunakan IAM Identity Center untuk mengautentikasi AWS SDK dan alat](#) Opsi ini adalah cara umum bagi organisasi untuk mengelola akses bagi tenaga kerja mereka dan mengharuskan Pusat Identitas diaktifkan.

Bagaimana cara kerjanya?

[Login untuk pengembangan AWS lokal menggunakan kredensial konsol](#) memungkinkan Anda menggunakan kredensial masuk Konsol AWS Manajemen yang ada untuk akses terprogram ke layanan. AWS Setelah alur otentikasi berbasis browser, buat AWS kredensial sementara yang berfungsi di seluruh alat pengembangan lokal seperti CLI AWS, Alat untuk dan. PowerShell AWS SDKs Fitur ini menyederhanakan proses konfigurasi dan pengelolaan kredensial AWS CLI, terutama jika Anda lebih memilih otentikasi interaktif daripada mengelola kunci akses jangka panjang.

Dengan proses ini, Anda dapat mengautentikasi menggunakan kredensial root yang dibuat selama penyiapan akun awal, pengguna IAM, atau identitas gabungan dari penyedia identitas Anda.

Jika Anda menggunakan SDKs untuk pengembangan, klien SDK akan menggunakan kredensial sementara melalui file. [AWS SDKs dan Penyedia kredensial standar Alat](#) Anda juga dapat mengkonfigurasi file [Penyedia kredensial login](#).

Otentikasi melalui perintah login didukung oleh AWS CLI dan Tools untuk: PowerShell

- [Login untuk pengembangan AWS lokal menggunakan kredensial konsol](#)
- [Login menggunakan kredensial konsol](#) di panduan pengguna Alat AWS untuk PowerShell

Menggunakan IAM Identity Center untuk mengautentikasi AWS SDK dan alat

AWS IAM Identity Center dapat digunakan untuk memberikan AWS kredensial ketika mengembangkan AWS aplikasi pada lingkungan layanan AWS non-komputasi. Jika Anda mengembangkan AWS sumber daya, seperti Amazon Elastic Compute Cloud (Amazon EC2) AWS Cloud9 atau, sebaiknya Anda mendapatkan kredensial dari layanan tersebut.

Gunakan autentikasi Pusat Identitas IAM jika Anda sudah menggunakan Pusat Identitas untuk akses AWS akun atau perlu mengelola akses untuk organisasi.

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 SDK/tool otentikasi.
- 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.

Prasyarat

Sebelum memulai prosedur ini, Anda harus menyelesaikan yang berikut:

- Jika Anda tidak memiliki Akun AWS, [mendaftar untuk Akun AWS](#).
- Jika Anda belum mengaktifkan IAM Identity Center, [aktifkan IAM Identity Center](#) dengan mengikuti petunjuk di AWS IAM Identity Center Panduan Pengguna.

Konfigurasi akses terprogram menggunakan IAM Identity Center

Langkah 1: Tetapkan akses dan pilih set izin yang sesuai

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 ini 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 SDKs 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 Issuer URL dan nilai yang Anda butuhkan untuk langkah berikutnya. Catatan: SSO Start URL dapat digunakan secara bergantian dengan Issuer URL
3. Di AWS CLI command prompt, jalankan `aws configure sso` perintah. Saat diminta, masukkan nilai konfigurasi yang Anda kumpulkan di langkah sebelumnya. Untuk detail tentang AWS CLI perintah ini, lihat [Mengkonfigurasi profil Anda dengan `aws configure sso wizard`](#).
 - a. Untuk prompt SSO Start URL, masukkan nilai yang Anda peroleh Issuer URL.
 - b. Untuk nama profil CLI, kami sarankan masuk `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 SDKs, 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 apa pun sebelum SDK atau alat akan menggunakan kredensial Pusat Identitas IAM karena diutamakan. [Memahami rantai penyedia kredensi](#)

Untuk mempelajari lebih dalam bagaimana alat SDKs dan menggunakan dan menyegarkan kredensial menggunakan konfigurasi ini, lihat [Bagaimana otentikasi IAM Identity Center diselesaikan untuk AWS SDKs dan alat](#)

Untuk mengonfigurasi setelan penyedia Pusat Identitas IAM secara langsung di `config` file bersama, lihat [Penyedia kredensi Pusat Identitas IAM](#) di panduan ini.

Menyegarkan sesi akses portal

Akses Anda pada akhirnya akan kedaluwarsa dan SDK atau alat akan mengalami kesalahan otentikasi. Kapan kedaluwarsa ini terjadi tergantung pada panjang sesi yang dikonfigurasi. 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 Pusat Identitas IAM — [Konfigurasi durasi sesi portal AWS akses pengguna Anda](#)
- Izin mengatur durasi sesi - [Mengatur durasi sesi](#)

Bagaimana otentikasi IAM Identity Center diselesaikan untuk AWS SDKs dan alat

Ketentuan Pusat Identitas IAM yang relevan

Istilah berikut membantu Anda memahami proses dan konfigurasi di belakang AWS IAM Identity Center. Dokumentasi untuk AWS SDK APIs 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, ruang nama 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.
AWS URL 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.

Nama Pusat Identitas IAM	Nama API SDK	Deskripsi
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	AWS kredensialnya kredensi sigv4	Kredensi 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 kredensi Pusat Identitas IAM	Penyedia kredensi SSO	Bagaimana Anda mendapatkan kredensialnya, seperti kelas atau modul yang menyediakan fungsionalitas.

Memahami resolusi kredensi SDK untuk Layanan AWS

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

Memulai sesi portal AWS akses

- Mulai proses dengan masuk ke sesi dengan kredensi Anda.
 - Gunakan `aws sso login` perintah di AWS Command Line Interface (AWS CLI). Ini memulai sesi Pusat Identitas IAM baru jika Anda belum memiliki sesi aktif.
- Saat memulai sesi baru, Anda menerima token penyegaran dan token akses dari IAM Identity Center. Ini AWS CLI juga memperbarui file JSON cache SSO dengan token akses baru dan token penyegaran dan membuatnya tersedia untuk digunakan oleh SDKs
- 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

SDKs menyediakan akses ke Layanan AWS saat Anda membuat instance objek klien per layanan. Ketika profil yang dipilih dari `AWS config` file bersama dikonfigurasi untuk resolusi kredensi 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 sigv4 APIs menggunakan sistem masuk tunggal Pusat Identitas IAM, SDK menggunakan token akses Pusat Identitas IAM untuk mendapatkan sesi IAM. Sesi IAM ini disebut sesi set izin, dan menyediakan AWS akses ke SDK dengan mengasumsikan peran IAM.

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

Kredensial set izin dikembalikan dari panggilan ke API Pusat [getRoleCredentials](#) Identitas IAM ke SDK. Objek klien SDK menggunakan peran IAM yang diasumsikan untuk melakukan panggilan ke

Layanan AWS, seperti meminta Amazon S3 untuk mencantumkan bucket di akun Anda. Objek klien dapat terus beroperasi menggunakan kredensial set izin tersebut hingga sesi set izin berakhir.

Kedaluwarsa sesi dan penyegaran

Saat menggunakan token akses per jam yang diperoleh dari IAM Identity Center secara otomatis di-refresh menggunakan token penyegaran. [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 yang di-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.

Menggunakan Peran IAM Di Mana Saja untuk mengautentikasi dan AWS SDKs alat

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 kredensi Anda.

Langkah 1: Konfigurasikan Peran IAM Di Mana Saja

IAM Roles Anywhere menyediakan cara untuk mendapatkan kredensi sementara untuk beban kerja atau proses yang berjalan di luar. AWS Jangkar kepercayaan didirikan dengan otoritas sertifikat untuk mendapatkan kredensi 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 with AWS SDKs and the AWS CLI, Anda dapat mengonfigurasi `credential_process` pengaturan dalam file bersama. AWS config SDKs 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 kredensi 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 kredensi 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. Konsol Manajemen AWS Contoh berikut menunjukkan config file bersama yang mengonfigurasi pengambilan kredensial sementara dari alat pembantu.

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

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

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

Dengan asumsi peran dengan AWS kredensi untuk mengautentikasi dan alat AWS SDKs

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 satu Layanan AWS atau lainnya Akun AWS. Untuk mempelajari lebih lanjut 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.

Note

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

Asumsikan peran IAM

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

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 mengatur ini, 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 sumber kredensial dari wadah Amazon ECS, instans Amazon EC2, atau dari variabel lingkungan.

Dengan asumsi peran dengan identitas web atau OpenID Connect AWS SDKs untuk mengautentikasi dan alat

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 penyedia identitas web atau OpenID Connect (OIDC), atau federasi SAMP. Untuk mempelajari selengkapnya tentang peran IAM, lihat [Metode untuk mengambil peran](#) dalam Panduan Pengguna IAM.

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

Note

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

Bersekutu dengan identitas web atau OpenID Connect

Anda dapat menggunakan JSON Web Tokens (JWTs) dari penyedia identitas publik, seperti Login With Amazon, Facebook, Google untuk mendapatkan AWS kredensi sementara menggunakan `AssumeRoleWithWebIdentity`. Tergantung pada bagaimana mereka digunakan, ini JWTs dapat disebut token ID atau token akses. Anda juga dapat menggunakan yang JWTs diterbitkan dari penyedia identitas (IdPs) yang kompatibel dengan protokol penemuan OIDC, seperti EntraID atau PingFederate.

Jika Anda menggunakan Amazon Elastic Kubernetes Service, fitur ini menyediakan kemampuan untuk menentukan peran IAM yang berbeda untuk setiap akun layanan Anda di kluster Amazon EKS.

Fitur Kubernetes ini mendistribusikan JWTs ke pod Anda yang kemudian digunakan oleh penyedia kredensi ini untuk mendapatkan kredensi sementara. AWS Untuk informasi selengkapnya tentang 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 JSON Web Token (JWT) untuk otentikasi, Anda harus terlebih dahulu mengatur 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 JSON Web Token (JWT) untuk otentikasi. AWS STS

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 pustaka inti SDKs dan 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 SDKs, lihat [Contoh Kode](#) di Panduan Pengembang Amazon Cognito.

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

Menggunakan tombol AWS akses untuk mengautentikasi AWS SDKs dan alat

Menggunakan tombol AWS akses adalah opsi untuk otentikasi saat menggunakan AWS SDKs dan alat.

Gunakan kredensial jangka pendek

Sebaiknya konfigurasi SDK atau alat yang akan digunakan [Menggunakan IAM Identity Center untuk mengautentikasi AWS SDK dan alat](#) untuk menggunakan opsi durasi sesi yang diperpanjang.

Namun, untuk mengatur SDK atau kredensial sementara alat secara langsung, lihat [Menggunakan kredensial jangka pendek untuk AWS SDKs mengautentikasi dan alat](#)

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](#).

Kelola 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 [Menggunakan kredensi jangka panjang untuk AWS SDKs mengautentikasi dan alat](#).

Menggunakan kredensi jangka pendek untuk AWS SDKs mengautentikasi dan alat

Sebaiknya konfigurasi AWS SDK atau alat Anda untuk digunakan [Menggunakan IAM Identity Center untuk mengautentikasi AWS SDK dan alat](#) 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.

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

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

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

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

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


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

Peringatan penting dan panduan untuk kredensial

Peringatan untuk kredensial

- JANGAN gunakan kredensi root akun Anda untuk mengakses AWS sumber daya. Kredensi ini menyediakan akses akun yang tidak terbatas dan sulit dicabut.
- 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 Konsol Manajemen AWS. 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 SDKs.
4. Untuk Buat tombol akses Langkah 2, masukkan tag opsional dan pilih Berikutnya.
5. Untuk Buat kunci akses Langkah 3, pilih Unduh file.csv untuk menyimpan .csv file dengan kunci akses 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.

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

Menggunakan peran IAM untuk mengautentikasi aplikasi yang digunakan ke Amazon EC2

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

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

Membuat peran IAM

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

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

1. Masuk ke Konsol Manajemen AWS 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 membuat EC2 instance Amazon Anda.

Luncurkan EC2 instans Amazon dan tentukan peran IAM Anda

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

- Ikuti [Luncurkan instance dengan cepat](#) di Panduan EC2 Pengguna Amazon. Namun, sebelum langkah pengajuan akhir, lakukan juga hal berikut:

- Di bawah Detail lanjutan, untuk profil Instans IAM, pilih peran yang Anda buat di langkah sebelumnya.

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

Connect ke EC2 instance

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

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

Note

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

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

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

Jalankan aplikasi Anda pada EC2 instance

1. Salin file aplikasi Anda dari drive lokal Anda ke EC2 instans Amazon Anda.
2. Mulai aplikasi dan verifikasi bahwa itu berjalan dengan hasil yang sama seperti pada mesin pengembangan Anda.
3. (Opsional) Verifikasi bahwa aplikasi menggunakan kredensial yang disediakan oleh peran IAM.

- a. Masuk ke Konsol Manajemen AWS dan buka EC2 konsol Amazon di <https://console.aws.amazon.com/ec2/>.
- b. Pilih instans.
- c. Pilih Tindakan, Keamanan, dan kemudian pilih Ubah peran IAM.
- d. Untuk peran IAM, lepaskan peran IAM dengan memilih No IAM Role.
- e. Pilih Perbarui peran IAM.
- f. Jalankan aplikasi lagi dan konfirmasikan bahwa ia mengembalikan kesalahan otorisasi.

Menggunakan plugin TIP untuk mengakses Layanan AWS

Trusted Identity Propagation (TIP) adalah fitur AWS IAM Identity Center yang memungkinkan administrator Layanan AWS untuk memberikan izin berdasarkan atribut pengguna seperti asosiasi grup. Dengan propagasi identitas tepercaya, konteks identitas ditambahkan ke peran IAM untuk mengidentifikasi pengguna yang meminta akses ke sumber daya. AWS Konteks ini disebarkan ke yang lain Layanan AWS.

Konteks identitas terdiri dari informasi yang Layanan AWS digunakan untuk membuat keputusan otorisasi ketika mereka menerima permintaan akses. Informasi ini mencakup metadata yang mengidentifikasi pemohon (misalnya, pengguna Pusat Identitas IAM), Layanan AWS akses yang diminta (misalnya, Amazon Redshift), dan ruang lingkup akses (misalnya, akses baca saja). Penerima Layanan AWS menggunakan konteks ini, dan izin apa pun yang diberikan kepada pengguna, untuk mengotorisasi akses ke sumber dayanya. Untuk informasi selengkapnya, lihat di [ikhtisar propagasi identitas tepercaya](#) di Panduan AWS IAM Identity Center Pengguna.

Plugin TIP dapat digunakan dengan Layanan AWS dukungan propagasi identitas tepercaya. Sebagai kasus penggunaan referensi, lihat [Mengonfigurasi aplikasi Amazon Q Business menggunakan AWS IAM Identity Center](#) dalam Panduan Pengguna Bisnis Amazon Q.

Note

Jika Anda menggunakan Amazon Q Business, lihat [Mengonfigurasi aplikasi Amazon Q Business menggunakan AWS IAM Identity Center petunjuk](#) khusus layanan.

Prasyarat untuk menggunakan plugin TIP

Sumber daya berikut diperlukan agar plugin berfungsi:

1. Anda harus menggunakan salah satu AWS SDK untuk Java atau AWS SDK untuk JavaScript.
2. Verifikasi bahwa layanan yang Anda gunakan mendukung propagasi identitas tepercaya.

Lihat kolom Aktifkan propagasi identitas tepercaya melalui Pusat Identitas IAM dari [aplikasi AWS terkelola yang terintegrasi dengan IAM Identity Center](#) tabel di Panduan Pengguna.AWS IAM Identity Center

3. Aktifkan Pusat Identitas IAM dan propagasi identitas tepercaya.

Lihat [prasyarat dan pertimbangan TIP](#) di Panduan Pengguna.AWS IAM Identity Center

4. Anda harus memiliki Identity-Center-integrated aplikasi.

Lihat [aplikasi AWS terkelola](#) atau [aplikasi yang dikelola Pelanggan](#) di Panduan AWS IAM Identity Center Pengguna.

5. Anda harus menyiapkan penerbit token tepercaya (TTI) dan menghubungkan layanan Anda ke IAM Identity Center.

Lihat [Prasyarat untuk penerbit token tepercaya dan Tugas untuk menyiapkan penerbit token tepercaya di Panduan Pengguna.AWS IAM Identity Center](#)

Untuk menggunakan plugin TIP dalam kode Anda

1. Buat instance plugin propagasi identitas tepercaya.
2. Buat instance klien layanan untuk berinteraksi dengan Anda Layanan AWS dan sesuaikan klien layanan dengan menambahkan plugin propagasi identitas tepercaya.

Plugin TIP mengambil parameter input berikut:

- **webTokenProvider**: Fungsi yang diterapkan pelanggan untuk mendapatkan token OpenID dari penyedia identitas eksternal mereka.
- **accessRoleArn**: Peran IAM ARN akan diasumsikan oleh plugin dengan konteks identitas pengguna untuk mendapatkan kredensial yang ditingkatkan identitas.
- **applicationArn**: String pengenal unik untuk klien atau aplikasi. Nilai ini adalah aplikasi ARN yang memiliki OAuth hibah yang dikonfigurasi.

- **ssoOidcClient**: (Opsional) Klien SSO OIDC, seperti [SsoOidcClient](#) untuk Java atau [client-sso-oidc](#) untuk JavaScript, dengan konfigurasi yang ditentukan pelanggan. Jika tidak disediakan, klien OIDC yang menggunakan `applicationRoleArn` akan dipakai dan digunakan.
- **stsClient**: (Opsional) AWS STS Klien dengan konfigurasi yang ditentukan pelanggan, digunakan untuk berasumsi `accessRoleArn` dengan konteks identitas pengguna. Jika tidak disediakan, AWS STS klien yang menggunakan `applicationRoleArn` akan dipakai dan digunakan.
- **applicationRoleArn**: (Opsional) Peran IAM ARN yang akan diasumsikan `AssumeRoleWithWebIdentity` sehingga OIDC AWS STS dan klien dapat di-bootstrap.
 - Jika tidak disediakan, kedua parameter `ssoOidcClient` dan `stsClient` parameter harus disediakan.
 - Jika disediakan, tidak `applicationRoleArn` bisa menjadi nilai yang sama dengan `accessRoleArn` parameter. `applicationRoleArn` digunakan untuk membangun `STSCient`, yang digunakan untuk mengasumsikan `AccessRole`. Jika peran yang sama digunakan untuk keduanya `applicationRole` dan `accessRole`, itu berarti menggunakan peran untuk mengasumsikan dirinya sendiri (asumsi peran diri), yang tidak disarankan oleh. AWS Lihat [pengumuman](#) untuk lebih jelasnya.

Pertimbangan untuk `ssoOidcClient`, `stsClient`, dan parameter `applicationRoleArn`

Saat mengonfigurasi plugin TIP, pertimbangkan persyaratan izin berikut berdasarkan parameter mana yang Anda berikan:

- Jika Anda menyediakan `ssoOidcClient` dan `stsClient`:
 - Kredensial pada `ssoOidcClient` harus memiliki `oauth:CreateTokenWithIAM` izin untuk memanggil pusat identitas untuk mendapatkan konteks pengguna khusus pusat identitas.
 - Kredensial pada `stsClient` harus memiliki `sts:AssumeRole`, dan `sts:SetContext` izin pada `accessRole` `accessRole` juga perlu dikonfigurasi dengan hubungan kepercayaan dengan kredensial aktif. `stsClient`
- Jika Anda menyediakan `applicationRoleArn`:
 - `applicationRole` harus memiliki `oauth:CreateTokenWithIAM`, `sts:AssumeRole` dan `sts:SetContext` izin pada sumber daya yang diperlukan (contoh `IDC`, `accessRole`) karena akan digunakan untuk membangun klien OIDC dan STS.

- `applicationRole` harus memiliki hubungan kepercayaan dengan penyedia identitas yang digunakan untuk menghasilkan `webToken`, karena `webToken` akan digunakan untuk mengasumsikan `ApplicationRole` melalui [AssumeRoleWithWebIdentity](#) panggilan oleh plugin.

Contoh `ApplicationRole` konfigurasi:

Kebijakan Kepercayaan dengan penyedia token Web:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Federated": "arn:aws:iam::ACCOUNT_ID:oidc-provider/IDENTITY_PROVIDER_URL"
      },
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
        "StringEquals": {
          "IDENTITY_PROVIDER_URL:aud": "CLIENT_ID_TO_BE_TRUSTED"
        }
      }
    }
  ]
}
```

Kebijakan Izin:

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

```
{
  "Effect": "Allow",
  "Action": [
    "sso-oauth:CreateTokenWithIAM"
  ],
  "Resource": [
    "*"
  ]
}
```

Contoh kode menggunakan TIP

Contoh di bawah ini menunjukkan bagaimana menerapkan plugin TIP dalam kode Anda menggunakan AWS SDK untuk Java atau AWS SDK untuk JavaScript.

Java

Untuk menggunakan plugin TIP dalam AWS SDK untuk Java proyek Anda, Anda perlu mendeklarasikannya sebagai dependensi dalam file proyek Anda. `pom.xml`

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

Dalam kode sumber Anda, sertakan pernyataan paket yang diperlukan untuk `software.amazon.awssdk.trustedidentitypropagation`.

Contoh berikut menunjukkan dua cara untuk membuat instance plugin propagasi identitas tepercaya dan menambahkannya ke klien layanan. Kedua contoh menggunakan Amazon S3 sebagai layanan dan digunakan `S3AccessGrantsPlugin` untuk mengelola izin khusus pengguna, tetapi dapat diterapkan ke semua Layanan AWS yang mendukung propagasi identitas tepercaya (TIP).

Note

Untuk contoh ini, Anda perlu mengatur izin khusus pengguna dari S3 Access Grants. Lihat [dokumentasi Hibah Akses S3](#) untuk detail selengkapnya.

Opsi 1: Bangun dan lewati klien OIDC dan STS

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

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

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

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

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

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

Opsi 2: Lulus applicationRoleArn dan tunda pembuatan klien ke plugin

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

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

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

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

Untuk detail dan sumber tambahan, lihat [trusted-identity-propagation-javadi](#) GitHub.

JavaScript

Jalankan perintah berikut untuk menginstal paket plugin otentikasi TIP dalam AWS SDK untuk JavaScript proyek Anda:


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

Final package.json harus mencakup ketergantungan yang mirip dengan yang berikut ini:

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

Dalam kode sumber Anda, impor `TrustedIdentityPropagationExtension` dependensi yang diperlukan.

Contoh berikut menunjukkan dua cara untuk membuat instance plugin propagasi identitas terpercaya dan menambahkannya ke klien layanan. Kedua contoh menggunakan Amazon S3 sebagai layanan dan memanfaatkan Amazon S3 Access Grants untuk mengelola izin khusus pengguna, tetapi dapat diterapkan ke semua yang mendukung propagasi identitas Layanan AWS terpercaya (TIP).

 Note

Untuk contoh ini, Anda perlu menyiapkan izin khusus pengguna dari Amazon S3 Access Grants, lihat dokumentasi Amazon S3 Access Grants untuk [detail selengkapnya](#).

Opsi 1: Bangun dan lewati klien OIDC dan STS

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

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

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

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

  const credentials = response.Credentials;

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

  // Use the temporary S3 client to perform the operation
  const s3Params = {
    Bucket: "BUCKET_NAME",
    Key: "S3_OBJECT_KEY",
  };
  const getObjectCommand = new GetObjectCommand(s3Params);
  const s3object = await temporaryS3Client.send(getObjectCommand);

  const fileContent = await s3object.Body.transformToString();

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

Opsi 2: Lulus applicationRoleArn dan tunda pembuatan klien ke plugin

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

const s3ControlClient = new S3ControlClient({
  region: "us-east-1",
  extensions: [
    TrustedIdentityPropagationExtension.create({
```

```
        webTokenProvider: async () => {
            return 'ID_TOKEN_FROM_YOUR_IDENTITY_PROVIDER';
        },
        accessRoleArn: accessRoleArn,
        applicationRoleArn: applicationRoleArn,
        applicationArn: applicationArn,
    })),
    ],
});

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

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

    const credentials = response.Credentials;

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

    const s3Params = {
        Bucket: "BUCKET_NAME",
        Key: "S3_OBJECT_KEY",
    };

    const getObjectCommand = new GetObjectCommand(s3Params);
    const s3object = await temporaryS3Client.send(getObjectCommand);

    const fileContent = await s3object.Body.transformToString();

    console.log("Successfully retrieved S3 object:", fileContent);
}
```

```
} catch (error) {  
  console.error("Error accessing S3 data:", error);  
}
```

Untuk detail dan sumber tambahan, lihat [trusted-identity-propagation-jsdi](#) GitHub.

AWS SDKs dan referensi pengaturan alat

SDKs menyediakan bahasa khusus APIs 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, SDKs memiliki strategi yang fleksibel untuk mendapatkan kredensi yang akan digunakan untuk permintaan Anda, untuk mempertahankan pengaturan untuk digunakan dengan setiap layanan, dan untuk mendapatkan nilai yang akan digunakan untuk pengaturan global.

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

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

Membuat klien layanan

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

Prioritas pengaturan

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

1. Pengaturan eksplisit apa pun yang disetel dalam kode atau pada klien layanan itu sendiri lebih diutamakan daripada yang lain.
 - Beberapa pengaturan dapat diatur berdasarkan per-operasi, dan dapat diubah sesuai kebutuhan untuk setiap operasi yang Anda panggil. Untuk AWS CLI atau Alat AWS untuk 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 SDKs dan alat mungkin memeriksa dalam urutan yang berbeda. Juga, beberapa SDKs dan alat mendukung metode lain untuk menyimpan dan mengambil parameter. Misalnya, AWS SDK untuk .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.

Memahami halaman pengaturan panduan ini

Halaman-halaman dalam bagian referensi Pengaturan dari panduan ini merinci pengaturan yang tersedia yang dapat diatur melalui berbagai mekanisme. Tabel berikut mencantumkan pengaturan file

konfigurasi dan kredensi, variabel lingkungan, dan (untuk Java dan Kotlin SDKs) setelah JVM yang dapat digunakan di luar kode Anda untuk mengonfigurasi fitur. Setiap topik tertaut di setiap daftar membawa Anda ke halaman pengaturan yang sesuai.

- [Configdaftar pengaturan file](#)
- [Credentialsdaftar pengaturan file](#)
- [Daftar variabel lingkungan](#)
- [Daftar properti sistem JVM](#)


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

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

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

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

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

 Note

Mungkin juga ada cara untuk mengonfigurasi fungsionalitas yang sama secara langsung di kode Anda, tetapi Referensi ini tidak mencakup ini karena unik untuk setiap SDK. Jika

Anda ingin menyetel konfigurasi dalam kode itu sendiri, lihat panduan SDK atau referensi API spesifik Anda.

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

Di bagian bawah setiap halaman fitur terdapat tabel Support by AWS SDKs and tools.

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

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

Configdaftar pengaturan file

Pengaturan yang tercantum dalam tabel berikut dapat ditetapkan dalam AWS config file bersama. Mereka global dan mempengaruhi semua Layanan AWS. SDKs 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
<code>account_id_endpoint_mode</code>	Titik akhir berbasis akun
<code>api_versions</code>	Pengaturan konfigurasi umum
<code>auth_scheme_preference</code>	Skema otentikasi

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

Nama pengaturan	Detail
ec2_metadata_service_endpoint_mode	Penyedia kredensi IMDS
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	EC2 Metadata contoh Amazon
metadata_service_timeout	EC2 Metadata contoh Amazon
mfa_serial	Asumsikan penyedia kredensi peran
output	Pengaturan konfigurasi umum
parameter_validation	Pengaturan konfigurasi umum

Nama pengaturan	Detail
region	Wilayah AWS
request_checksum_calculation	Perlindungan Integritas Data untuk Amazon S3
request_minimum_compression_size_bytes	Minta kompresi
response_checksum_validation	Perlindungan Integritas Data untuk Amazon S3
retry_mode	Coba lagi perilaku
role_arn	Asumsikan penyedia kredensi peran
role_session_name	Asumsikan penyedia kredensi peran
s3_disable_express_session_auth	Otentikasi sesi S3 Express One Zone
s3_disable_multiregion_access_points	Titik Akses Multi-Wilayah Amazon S3
s3_use_arn_region	Titik akses Amazon S3
sdk_ua_app_id	ID Aplikasi

Nama pengaturan	Detail
sigv4a_signing_region_set	Skema otentikasi
source_profile	Asumsikan penyedia kredensi peran
sso_account_id	Penyedia kredensi Pusat Identitas IAM
sso_region	Penyedia kredensi Pusat Identitas IAM
sso_registration_scopes	Penyedia kredensi Pusat Identitas IAM
sso_role_name	Penyedia kredensi Pusat Identitas IAM
sso_start_url	Penyedia kredensi Pusat Identitas IAM
sts_regional_endpoints	AWS STS Titik akhir regional
use_dualstack_endpoint	Dual-stack dan titik akhir FIPS
use_fips_endpoint	Dual-stack dan titik akhir FIPS
web_identity_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. SDKs 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_secret_access_key	AWS kunci akses
aws_session_token	AWS kunci akses

Daftar variabel lingkungan

Variabel lingkungan yang didukung oleh sebagian besar SDKs tercantum dalam tabel berikut. Mereka global dan mempengaruhi semua Layanan AWS. SDKs 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_ACCOUNT_ID	Titik akhir berbasis akun
AWS_ACCOUNT_ID_ENDPOINT_MODE	Titik akhir berbasis akun
AWS_AUTH_SCHEME_PREFERENCE	Skema otentikasi
AWS_CA_BUNDLE	Pengaturan konfigurasi umum

Nama pengaturan	Detail	
AWS_CONFIG_FILE	Menemukan dan mengubah lokasi berbagi config dan credentials file AWS SDKs dan alat	
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_HOST_PREFIX_INJECTION	Injeksi awalan host	
AWS_DISABLE_REQUEST_COMPRESSION	Minta kompresi	

Nama pengaturan	Detail
AWS_EC2_M ETADATA_D ISABLED	Penyedia kredensi IMDS
AWS_EC2_M ETADATA_S ERVICE_EN DPOINT	Penyedia kredensi IMDS
AWS_EC2_M ETADATA_S ERVICE_EN DPOINT_MODE	Penyedia kredensi IMDS
AWS_EC2_M ETADATA_V 1_DISABLED	Penyedia kredensi IMDS
AWS_ENABL E_ENDPOIN T_DISCOVERY	Penemuan titik akhir
AWS_ENDPO INT_URL	Titik akhir khusus layanan
AWS_ENDPO INT_URL_< SERVICE>	Titik akhir khusus layanan
AWS_IGNOR E_CONFIGU RED_ENDPO INT_URLS	Titik akhir khusus layanan
AWS_MAX_A TTEMPTS	Coba lagi perilaku

Nama pengaturan	Detail
AWS_METAD ATA_SERVI CE_NUM_AT TEMPTS	EC2 Metadata contoh Amazon
AWS_METAD ATA_SERVI CE_TIMEOUT	EC2 Metadata contoh Amazon
AWS_PROFILE	Menggunakan <code>credentials</code> file bersama <code>config</code> dan untuk mengkonfigurasi AWS SDKs dan alat secara global
AWS_REGION	Wilayah AWS
AWS_REQUE ST_CHECKS UM_CALCULATION	Perlindungan Integritas Data untuk Amazon S3
AWS_REQUE ST_MIN_CO MPRESSION _SIZE_BYTES	Minta kompresi
AWS_RESPO NSE_CHECK SUM_VALIDATION	Perlindungan Integritas Data untuk Amazon S3
AWS_RETRY_MODE	Coba lagi perilaku
AWS_ROLE_ARN	Asumsikan penyedia kredensi peran
AWS_ROLE_ SESSION_NAME	Asumsikan penyedia kredensi peran

Nama pengaturan	Detail
AWS_S3_DI SABLE_EXP RESS_SESS ION_AUTH	Otentikasi sesi S3 Express One Zone
AWS_S3_DI SABLE_MUL TIREGION_ ACCESS_POINTS	Titik Akses Multi-Wilayah Amazon S3
AWS_S3_US E_ARN_REGION	Titik akses Amazon S3
AWS_SDK_U A_APP_ID	ID Aplikasi
AWS_SECRE T_ACCESS_KEY	AWS kunci akses
AWS_SESSI ON_TOKEN	AWS kunci akses
AWS_SHARE D_CREDENT IALS_FILE	Menemukan dan mengubah lokasi berbagi config dan credentials file AWS SDKs dan alat
AWS_SIGV4 A_SIGNING _REGION_SET	Skema otentikasi
AWS_STS_R EGIONAL_E NDPOINTS	AWS STS Titik akhir regional
AWS_USE_D UALSTACK_ ENDPOINT	Dual-stack dan titik akhir FIPS

Nama pengaturan	Detail
AWS_USE_FIPS_ENDPOINT	Dual-stack dan titik akhir FIPS
AWS_WEB_IDENTITY_TOKEN_FILE	Asumsikan penyedia kredensi peran

Daftar properti sistem JVM

Anda dapat menggunakan properti sistem JVM berikut untuk AWS SDK untuk Java dan AWS SDK untuk 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.accountId</code>	Titik akhir berbasis akun
<code>aws.accountIdEndpointMode</code>	Titik akhir berbasis akun
<code>aws.authSchemePreference</code>	Skema otentikasi
<code>aws.configFile</code>	Menemukan dan mengubah lokasi berbagi config dan credentials file AWS SDKs dan alat
<code>aws.defaultsMode</code>	Default konfigurasi cerdas

Nama pengaturan	Detail
<code>aws.disableEc2MetadataV1</code>	Penyedia kredensi IMDS
<code>aws.disableHostPrefixInjection</code>	Injeksi awalan host
<code>aws.disableRequestCompression</code>	Minta kompresi
<code>aws.disableS3ExpressAuth</code>	Otentikasi sesi S3 Express One Zone
<code>aws.ec2MetadataServiceEndpoint</code>	Penyedia kredensi IMDS
<code>aws.ec2MetadataEndpointMode</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>	Menggunakan <code>credentials</code> file bersama <code>config</code> dan untuk mengkonfigurasi AWS SDKs dan alat secara global
<code>aws.region</code>	Wilayah AWS
<code>aws.requestChecksumCalculation</code>	Perlindungan Integritas Data untuk Amazon S3
<code>aws.requestMinCompressionSizeBytes</code>	Minta kompresi
<code>aws.responseChecksumValidation</code>	Perlindungan Integritas Data untuk Amazon S3
<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

Nama pengaturan	Detail
<code>aws.s3UseArnRegion</code>	Titik akses Amazon S3
<code>aws.secretsAccessKey</code>	AWS kunci akses
<code>aws.sessionToken</code>	AWS kunci akses
<code>aws.sharedCredentialsFile</code>	Menemukan dan mengubah lokasi berbagi config dan credentials file AWS SDKs dan alat
<code>aws.useDualstackEndpoint</code>	Dual-stack dan titik akhir FIPS
<code>aws.useFipsEndpoint</code>	Dual-stack dan titik akhir FIPS
<code>aws.webIdentityTokenFile</code>	Asumsikan penyedia kredensi peran
<code>sdk.ua.appId</code>	ID Aplikasi

AWS SDKs dan Penyedia kredensi standar Alat

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

⚠ Important

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

Topik

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

Memahami rantai penyedia kredensi

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

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

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 kredensi peran	Masuk menggunakan penyedia identitas eksternal (IDP) yang terkenal, seperti Login with Amazon, Facebook, Google, atau IDP lain yang kompatibel dengan OpenID Connect (OIDC). Asumsikan izin peran IAM menggunakan JSON Web Token (JWT) from (). AWS Security Token Service AWS STS
Penyedia kredensi login	Dapatkan kredensial untuk sesi konsol baru atau yang sudah ada yang Anda masuki.
Penyedia kredensi Pusat Identitas IAM	Dapatkan kredensial dari. AWS IAM Identity Center
Asumsikan penyedia kredensi peran	Dapatkan akses ke sumber daya lain dengan mengasumsikan izin peran IAM. (Ambil dan kemudian gunakan kredensi sementara untuk peran).
Penyedia kredensi kontainer	Amazon Elastic Container Service (Amazon ECS) dan kredensi Amazon Elastic Kubernetes Service (Amazon EKS). Penyedia kredensi kontainer mengambil kredensi untuk aplikasi kontainer pelanggan.
Penyedia kredensi proses	Penyedia kredensi khusus. Dapatkan kredensial Anda dari sumber atau proses eksternal, termasuk IAM 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 [Menggunakan credentials file bersama config dan untuk mengkonfigurasi AWS SDKs dan alat secara global](#). Untuk informasi selengkapnya, lihat [Prioritas pengaturan](#).

Rantai penyedia kredensi khusus SDK dan khusus alat

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

- [AWS CLI](#)
- [SDK for C++](#)
- [SDK for Go](#)
- [SDK for Java](#)
- [SDK untuk JavaScript](#)
- [SDK para Kotlin](#)
- [SDK for .NET](#)
- [SDK for PHP](#)
- [SDK untuk Python \(Boto3\)](#)
- [SDK for Ruby](#)
- [SDK untuk Rust](#)
- [SDK para Swift](#)
- [Alat untuk PowerShell](#)


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 kredensi 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 kredensi 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 kredensi untuk mengautentikasi pengguna. Anda menerima nilai ini sebagai bagian dari kredensi 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 [Menggunakan kredensial jangka pendek untuk AWS SDKs mengautentikasi dan alat](#).

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

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

Linux/macOS contoh pengaturan variabel lingkungan melalui baris perintah:

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

Contoh Windows pengaturan variabel lingkungan melalui baris perintah:

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

Support oleh AWS SDKs dan alat

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

SDK	Di Catatan atau informasi lebih lanjut
AWS CLI v2	Ya

SDK	Di	Catatan atau informasi lebih lanjut
SDK for C++	Ya	<code>configfile</code> bersama tidak didukung.
SDK for Go V2 (1.x)	Ya	
SDK for Go 1.x (V1)	Ya	Untuk menggunakan pengaturan <code>config file</code> 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 4.x	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 para Swift	Ya	
Alat untuk PowerShell V5	Ya	
Alat untuk PowerShell V4	Ya	Variabel lingkungan tidak didukung.

Penyedia kredensi login

Anda dapat [menggunakan kredensial masuk Konsol AWS Manajemen yang ada untuk memperoleh kredensial](#) jangka pendek yang dapat digunakan untuk akses terprogram. Setelah Anda

menyelesaikan alur otentikasi berbasis browser, buat AWS kredensial sementara yang berfungsi di seluruh alat pengembangan lokal seperti CLI AWS, Alat untuk dan. AWS PowerShell AWS SDKs

Untuk menghasilkan kredensial ini, jalankan `aws login` perintah di AWS CLI, atau `Invoke-AWSLogin cmdlet` di Tools for. AWS PowerShell Kredensial jangka pendek yang dihasilkan akan di-cache secara lokal, di mana mereka dapat digunakan kembali oleh. AWS SDKs Kredensial jangka pendek kedaluwarsa dalam 15 menit, tetapi CLI dan SDKs akan secara otomatis menyegarkannya sesuai kebutuhan hingga 12 jam. Ketika token penyegaran kedaluwarsa, Anda akan diminta untuk masuk lagi melalui CLI atau. PowerShell

Perintah login akan memperbarui profil yang Anda tentukan dengan `login_session` pengaturan, yang menyimpan identitas sesi konsol manajemen yang Anda pilih selama alur kerja login.

```
[profile console]
login_session = arn:aws:iam::0123456789012:user/username
region = us-west-2
```

Secara default, kredensial jangka pendek dan token penyegaran disimpan dalam file JSON di `~/.aws/login/cache` direktori di Linux dan macOS, atau di Windows. `%USERPROFILE%\aws\login\cache` Nama file didasarkan pada nama sesi login. Anda dapat mengganti direktori dengan mengatur variabel `AWS_LOGIN_CACHE_DIRECTORY` lingkungan.

Pengaturan Penyedia Masuk

Konfigurasi fungsi ini dengan menggunakan yang berikut:

AWS_LOGIN_CACHE_DIRECTORY- variabel lingkungan

Direktori alternatif di mana CLI dan SDKs akan menyimpan kredensial cache yang dipetakan ke profil sesi login.

Nilai default: `~/.aws/login/cache` di Linux dan macOS, atau `%USERPROFILE%\aws\login\cache` di Windows.

Support oleh AWS SDKs dan alat

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

SDK	Di	Catatan atau informasi lebih lanjut
AWS CLI v2	Ya	
SDK for C++	Ya	
SDK for Go V2 (1.x)	Tida	
SDK for Go 1.x (V1)	Ya	
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 4.x	Ya	
SDK for .NET 3.x	Ya	
SDK for PHP 3.x	Ya	
SDK untuk Python (Boto3)	Ya	Membutuhkan CRT
SDK for Ruby 3.x	Ya	
SDK untuk Rust	Ya	
Alat untuk PowerShell V5	Ya	
Alat untuk PowerShell V4	Tida	

Asumsikan penyedia kredensi peran

Note

Untuk bantuan dalam memahami tata letak halaman pengaturan, atau dalam menafsirkan tabel Support by AWS SDKs and tools berikut, lihat [Memahami halaman pengaturan panduan ini](#).

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 [Dengan asumsi peran dengan AWS kredensi untuk mengautentikasi dan alat AWS SDKs](#) di panduan ini.

Asumsikan pengaturan penyedia kredensi peran

Konfigurasi fungsi ini dengan menggunakan yang berikut:

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 Amazon ECS atau peran IAM yang dilampirkan ke wadah Amazon EKS 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 pengenal 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. Untuk informasi selengkapnya tentang MFA, lihat [Autentikasi AWS multi-faktor di IAM di Panduan Pengguna IAM](#).

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. Format ARN adalah:
`arn:aws:iam::account-id:mfa/mfa-device-name`

Contoh pengaturan ini dalam config file:

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

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

role_arn- Pengaturan AWS **config** file bersama, **AWS_ROLE_ARN**- variabel lingkungan, **aws.roleArn**- Properti 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_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. Untuk detailnya, lihat [elemen CloudTrail UserIdentity](#) di AWS CloudTrail Panduan Pengguna.

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

Nilai yang valid: Disediakan untuk `RoleSessionName` parameter 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.
```

Contoh pengaturan ini dalam `config` file:

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

source_profile- Pengaturan AWS `config` file bersama

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

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

Nilai default: Tidak ada.

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

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

Contoh pengaturan ini dalam file konfigurasi:

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

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

Pada contoh sebelumnya, A profil memberi tahu SDK atau alat untuk secara otomatis mencari kredensial untuk profil yang ditautkan. B Dalam hal ini, B profil menggunakan alat bantu kredensial yang disediakan oleh [Menggunakan Peran IAM Di Mana Saja untuk mengautentikasi dan AWS SDKs alat](#) untuk mendapatkan kredensial untuk SDK. AWS Kredensial sementara tersebut kemudian digunakan oleh kode Anda 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. Setiap tindakan yang diambil oleh profil A memiliki nama sesi peran yang disertakan dalam CloudTrail log.

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

```
[profile A]
```

```
source_profile = B
role_arn = arn:aws:iam::123456789012:role/RoleA
role_session_name = ProfileARoleSession

[profile B]
credential_source=Ec2InstanceMetadata
```

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

web_identity_token_file- Pengaturan AWS **config** file bersama,
AWS_WEB_IDENTITY_TOKEN_FILE- variabel lingkungan, **aws.webIdentityTokenFile**- Properti 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.

Support oleh AWS SDKs dan alat

Berikut ini SDKs mendukung fitur dan pengaturan yang dijelaskan dalam topik ini. Setiap pengecualian sebagian dicatat. Setiap pengaturan properti sistem JVM didukung oleh AWS SDK untuk Java dan satu-satunya. AWS SDK untuk 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	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	Parsi	<code>mfa_serial</code> tidak didukung. <code>duration_seconds</code> tidak didukung.
SDK for Java 1.x	Parsi	<code>credential_source</code> tidak didukung. <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	
SDK for .NET 4.x	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 para Swift	Ya	
Alat untuk PowerShell V5	Ya	
Alat untuk PowerShell V4	Ya	

Penyedia kredensi kontainer

Note

Untuk bantuan dalam memahami tata letak halaman pengaturan, atau dalam menafsirkan tabel Support by AWS SDKs and tools berikut, lihat [Memahami halaman pengaturan panduan ini](#).

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

Jika Anda menggunakan Amazon ECS, kami sarankan Anda menggunakan tugas Peran IAM untuk meningkatkan isolasi kredensi, otorisasi, dan auditabilitas. Saat dikonfigurasi, Amazon ECS menetapkan variabel `AWS_CONTAINER_CREDENTIALS_RELATIVE_URI` lingkungan yang digunakan SDKs dan alat untuk mendapatkan kredensi. Untuk mengonfigurasi Amazon ECS untuk fungsionalitas ini, lihat [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-kredensi aplikasi Anda. Untuk mempelajari selengkapnya tentang Identitas Pod [Amazon EKS](#), lihat [Identitas Pod](#) Amazon EKS di Panduan Pengguna Amazon EKS. Saat dikonfigurasi, Amazon EKS menetapkan variabel `AWS_CONTAINER_CREDENTIALS_FULL_URI` dan `AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE` lingkungan yang digunakan SDKs dan alat untuk mendapatkan kredensi. 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
```

Support oleh AWS SDKs dan alat

Berikut ini SDKs mendukung fitur dan pengaturan yang dijelaskan dalam topik ini. Setiap pengecualian sebagian dicatat. Setiap pengaturan properti sistem JVM didukung oleh AWS SDK untuk Java dan satu-satunya. AWS SDK untuk 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 Ketika Lambda SnapStart diaktifkan, <code>AWS_CONTAINER_CREDENTIALS_FULL_URI</code> dan secara otomatis <code>AWS_CONTAINER_AUTHORIZATION_TOKEN</code> digunakan untuk otentikasi.
SDK for Java 1.x	Ya Ketika Lambda SnapStart diaktifkan, <code>AWS_CONTAINER_CREDENTIALS_FULL_URI</code> dan secara otomatis

SDK	Di	Catatan atau informasi lebih lanjut
		AWS_CONTAINER_AUTHORIZATION_TOKEN digunakan untuk otentikasi.
SDK untuk 3.x JavaScript	Ya	
SDK untuk 2.x JavaScript	Ya	
SDK para Kotlin	Ya	
SDK for .NET 4.x	Ya	Ketika Lambda SnapStart diaktifkan, AWS_CONTAINER_CREDENTIALS_FULL_URI dan secara otomatis AWS_CONTAINER_AUTHORIZATION_TOKEN digunakan untuk otentikasi.
SDK for .NET 3.x	Ya	Ketika Lambda SnapStart diaktifkan, AWS_CONTAINER_CREDENTIALS_FULL_URI dan secara otomatis AWS_CONTAINER_AUTHORIZATION_TOKEN digunakan untuk otentikasi.
SDK for PHP 3.x	Ya	
SDK untuk Python (Boto3)	Ya	Ketika Lambda SnapStart diaktifkan, AWS_CONTAINER_CREDENTIALS_FULL_URI dan secara otomatis AWS_CONTAINER_AUTHORIZATION_TOKEN digunakan untuk otentikasi.
SDK for Ruby 3.x	Ya	
SDK untuk Rust	Ya	
SDK para Swift	Ya	
Alat untuk PowerShell V5	Ya	
Alat untuk PowerShell V4	Ya	

Penyedia kredensi Pusat Identitas IAM

Note

Untuk bantuan dalam memahami tata letak halaman pengaturan, atau dalam menafsirkan tabel Support by AWS SDKs and tools berikut, lihat [Memahami halaman pengaturan panduan ini](#).

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 mengotentikasi 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. [Bagaimana otentikasi IAM Identity Center diselesaikan untuk AWS SDKs dan alat](#)

Ada dua cara untuk mengkonfigurasi IAM Identity Center melalui config file:

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

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

Dua panduan berikut berisi informasi tambahan tentang IAM Identity Center:

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

Untuk menyelam lebih dalam tentang cara SDKs dan alat menggunakan dan menyegarkan kredensial menggunakan konfigurasi ini, lihat [Bagaimana otentikasi IAM Identity Center diselesaikan untuk AWS SDKs dan alat](#)

Prasyarat

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

Note

Atau, untuk prasyarat lengkap dan konfigurasi config file bersama yang diperlukan yang dirinci di halaman ini, lihat petunjuk yang dipandu untuk pengaturan. [Menggunakan IAM Identity Center untuk mengautentikasi AWS SDK dan alat](#)

Konfigurasi penyedia token SSO

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-sessionBagian config file digunakan untuk mengelompokkan variabel konfigurasi untuk memperoleh token akses SSO, yang kemudian dapat digunakan untuk memperoleh AWS kredensial. Untuk detail lebih lanjut tentang bagian ini dalam config file, lihat [Format file konfigurasi](#).

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

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

[sso-session my-sso]
sso_region = us-east-1
```

```
sso_start_url = https://my-sso-portal.awsapps.com/start
sso_registration_scopes = sso:account:access
```

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

`sso_account_id` dan `sso_role_name` tidak diperlukan untuk semua skenario konfigurasi 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 Layanan AWS panduan individual untuk menentukan apakah layanan 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. Contoh sebelumnya ditetapkan `sso_registration_scopes` untuk menyediakan akses yang diperlukan untuk daftar akun dan peran.

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

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

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

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

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

Konfigurasi lama yang tidak dapat disegarkan

Penyegaran token otomatis tidak didukung menggunakan konfigurasi lama yang tidak dapat disegarkan. Kami merekomendasikan menggunakan [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

Konfigurasikan fungsi ini dengan menggunakan yang berikut:

sso_start_url- Pengaturan AWS **config** file bersama

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

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

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

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 Services 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` Aplikasi dapat meminta satu atau lebih cakupan, dan token akses yang dikeluarkan untuk aplikasi terbatas pada cakupan yang diberikan. Cakupan minimum `sso:account:access` harus

diberikan untuk mendapatkan token penyegaran kembali dari layanan IAM Identity Center. Untuk daftar opsi cakupan akses yang tersedia, lihat [Cakupan akses](#) di Panduan AWS IAM Identity Center Pengguna.

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

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

Support oleh AWS SDKs dan alat

Berikut ini SDKs mendukung fitur dan pengaturan yang dijelaskan dalam topik ini. Setiap pengecualian sebagian dicatat. Setiap pengaturan properti sistem JVM didukung oleh AWS SDK untuk Java dan satu-satunya. AWS SDK untuk 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 4.x	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	Parsi	Konfigurasi lama yang tidak dapat disegarkan saja.
SDK para Swift	Ya	
Alat untuk PowerShell V5	Ya	
Alat untuk PowerShell V4	Ya	

Penyedia kredensi IMDS

Note

Untuk bantuan dalam memahami tata letak halaman pengaturan, atau dalam menafsirkan tabel Support by AWS SDKs and tools berikut, lihat [Memahami halaman pengaturan panduan ini](#).

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 [Bekerja dengan metadata instans](#) di EC2 Panduan Pengguna Amazon. Amazon EC2 menyediakan endpoint lokal yang tersedia untuk instans yang dapat memberikan berbagai bit informasi ke instans. Jika instance memiliki peran yang dilampirkan, itu dapat memberikan serangkaian kredensial yang valid untuk peran itu. Endpoint SDKs dapat menggunakan endpoint tersebut untuk menyelesaikan kredensi sebagai bagian dari rantai penyedia [kredensi default](#) mereka. Instance Metadata Service Version 2 (IMDSv2), versi 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 ini:

AWS_EC2_METADATA_DISABLED- variabel lingkungan

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

Nilai default:`false`.

Nilai yang valid:

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

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

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

Note

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

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 ini mengesampingkan lokasi default yang AWS SDKs dan alat akan mencari metadata EC2 instans Amazon.

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

Nilai yang valid: 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. [Memahami 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 Amazon EC2 Instance Metadata Service (IMDS) untuk mengambil kredensial untuk peran. AWS Perilaku ini dapat dinonaktifkan dengan menyetel variabel **AWS_EC2_METADATA_DISABLED** lingkungan ke `true`. Hal ini mencegah aktivitas jaringan yang tidak perlu dan meningkatkan keamanan pada jaringan yang tidak tepercaya di mana Layanan Metadata EC2 Instans Amazon dapat ditiru.

Note

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

Menonaktifkan penggunaan kredensi Amazon IMDS EC2

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.

Support oleh AWS SDKs dan alat

Berikut ini SDKs mendukung fitur dan pengaturan yang dijelaskan dalam topik ini. Setiap pengecualian sebagian dicatat. Setiap pengaturan properti sistem JVM didukung oleh AWS SDK untuk Java dan satu-satunya. AWS SDK untuk 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	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 IMDSv1 fallback.
SDK for .NET 4.x	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	Tidak menggunakan IMDSv1 fallback.
SDK para Swift	Ya	
Alat untuk PowerShell V5	Ya	Anda dapat menonaktifkan IMDSv1 fallback secara eksplisit dalam kode menggunakan. <code>[Amazon.Util].EC2In</code>

SDK	Di Catatan atau informasi lebih lanjut <pre>InstanceMetadata>::EC2MetadataV1Disabled = \$true</pre>
Alat untuk PowerShell V4	Ya Anda dapat menonaktifkan IMDSv1 fallback secara eksplisit dalam kode menggunakan. [Amazon.Util.EC2InstanceMetadata>::EC2MetadataV1Disabled = \$true

Penyedia kredensi proses

Note

Untuk bantuan dalam memahami tata letak halaman pengaturan, atau dalam menafsirkan tabel Support by AWS SDKs and tools berikut, lihat [Memahami halaman pengaturan panduan ini](#).

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

Misalnya, IAM Roles Anywhere digunakan `credential_process` untuk mendapatkan kredensial sementara atas nama aplikasi Anda. `credential_process` Untuk mengonfigurasi penggunaan ini, lihat [Menggunakan Peran IAM Di Mana Saja untuk mengautentikasi dan AWS SDKs alat](#).

Note

Berikut ini menjelaskan metode sumber kredensial dari proses eksternal dan dapat digunakan jika Anda menjalankan perangkat lunak di luar. AWS Jika Anda membangun sumber daya AWS komputasi, gunakan penyedia kredensi lainnya. Jika menggunakan opsi ini, Anda harus memastikan bahwa file konfigurasi dikunci sebanyak mungkin menggunakan praktik terbaik keamanan untuk sistem operasi Anda. Konfirmasikan bahwa alat kredensi khusus Anda tidak menulis informasi rahasia apa pun `StdErr`, karena SDKs 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 kredensi otentikasi untuk digunakan. Pengaturan menentukan nama program/command 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.

Note

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

Menentukan jalur ke program kredensial

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 menemukan program dalam folder yang ditentukan oleh variabel PATH lingkungan. Jalur bervariasi tergantung pada sistem operasi:

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

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

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

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

- Dapat ditentukan dalam profil khusus:

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

Output yang valid dari program kredensial

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

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

Note

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

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

Support oleh AWS SDKs dan alat

Berikut ini SDKs mendukung fitur dan pengaturan yang dijelaskan dalam topik ini. Setiap pengecualian sebagian dicatat. Setiap pengaturan properti sistem JVM didukung oleh AWS SDK untuk Java dan satu-satunya. AWS SDK untuk 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	Ya	
SDK untuk 3.x JavaScript	Ya	
SDK untuk 2.x JavaScript	Ya	
SDK para Kotlin	Ya	
SDK for .NET 4.x	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 para Swift	Ya	
Alat untuk PowerShell V5	Ya	
Alat untuk PowerShell V4	Ya	

AWS SDKs dan fitur standar Alat

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

⚠ Important

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

Topik

- [Titik akhir berbasis akun](#)
- [ID Aplikasi](#)
- [EC2 Metadata contoh Amazon](#)
- [Titik akses Amazon S3](#)
- [Titik Akses Multi-Wilayah Amazon S3](#)
- [Otentikasi sesi S3 Express One Zone](#)
- [Skema otentikasi](#)
- [Wilayah AWS](#)
- [AWS STS Titik akhir regional](#)
- [Perlindungan Integritas Data untuk Amazon S3](#)
- [Dual-stack dan titik akhir FIPS](#)
- [Penemuan titik akhir](#)
- [Pengaturan konfigurasi umum](#)
- [Injeksi awalan host](#)
- [Klien IMDS](#)
- [Coba lagi perilaku](#)
- [Minta kompresi](#)
- [Titik akhir khusus layanan](#)
- [Default konfigurasi cerdas](#)

Titik akhir berbasis akun

Note

Untuk bantuan dalam memahami tata letak halaman pengaturan, atau dalam menafsirkan tabel Support by AWS SDKs and tools berikut, lihat [Memahami halaman pengaturan panduan ini](#).

Titik akhir berbasis akun membantu memastikan kinerja dan skalabilitas tinggi dengan menggunakan Akun AWS ID Anda untuk merutekan permintaan layanan yang mendukung fitur ini. Bila Anda menggunakan AWS SDK dan layanan yang mendukung endpoint berbasis akun, klien SDK akan membuat dan menggunakan endpoint berbasis akun, bukan endpoint regional. Jika ID akun tidak terlihat oleh klien SDK, klien akan menggunakan titik akhir regional. Endpoint berbasis akun mengambil bentuk `https://<account-id>.ddb.<region>.amazonaws.com`, di mana `<account-id>` dan `<region>` adalah ID Anda Akun AWS dan Wilayah AWS

Konfigurasi fungsi ini dengan menggunakan yang berikut:

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

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

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

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

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

Nilai default: `preferred`

Nilai yang valid:

- **preferred**— Titik akhir harus menyertakan ID akun jika tersedia.
- **disabled**— Titik akhir yang diselesaikan tidak menyertakan ID akun.

- **required**— Titik akhir harus menyertakan ID akun. Jika ID akun tidak tersedia, SDK akan memunculkan kesalahan.

Support oleh AWS SDKs dan alat

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

SDK	Didu	Dirilis dalam versi SDK	Catatan atau informasi lebih lanjut
AWS CLI v2	Ya	2.25.0	
AWS CLI v1	Ya	1.38.0	
SDK for C++	Tidak		
SDK for Go V2 (1.x)	Ya	v1.35.0	
SDK for Go 1.x (V1)	Tidak		
SDK for Java 2.x	Ya	v2.28.4	
SDK for Java 1.x	Ya	v1.12.771	
SDK untuk 3.x JavaScript	Ya	v3.656.0	
SDK untuk 2.x JavaScript	Tidak		
SDK para Kotlin	Ya	v1.3.37	
SDK for .NET 4.x	Ya	4.0.0	
SDK for .NET 3.x	Tidak		

SDK	Didu	Dirilis dalam versi SDK	Catatan atau informasi lebih lanjut
SDK for PHP 3.x	Ya	v3.318.0	
SDK untuk Python (Boto3)	Ya	1.37.0	
SDK for Ruby 3.x	Ya	v1.123.0	
SDK untuk Rust	Ya	rilis-2025-04-24	
SDK para Swift	Ya	1.2.0	
Alat untuk PowerShell V5	Tidak		
Alat untuk PowerShell V4	Tidak		

ID Aplikasi

Note

Untuk bantuan dalam memahami tata letak halaman pengaturan, atau dalam menafsirkan tabel Support by AWS SDKs and tools berikut, lihat [Memahami halaman pengaturan panduan ini](#).

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 SDKs dan layanan tidak menggunakan atau menafsirkan nilai ini selain untuk memunculkannya kembali dalam komunikasi pelanggan. Misalnya, nilai ini dapat dimasukkan dalam email operasional atau Dasbor AWS Health 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, **sdk.ua.appId**- 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.

Support oleh AWS SDKs dan alat

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

SDK	Di Catatan atau informasi lebih lanjut
AWS CLI v2	Ya

SDK	Di	Catatan atau informasi lebih lanjut
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 for Java 1.x	Tida	
SDK untuk 3.x JavaScript	Ya	
SDK untuk 2.x JavaScript	Tida	
SDK para Kotlin	Ya	Properti sistem JVM adalah. <code>aws.userAgentAppId</code>
SDK for .NET 4.x	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 para Swift	Ya	
Alat untuk PowerShell V5	Ya	
Alat untuk PowerShell V4	Ya	

EC2 Metadata contoh Amazon

Note

Untuk bantuan dalam memahami tata letak halaman pengaturan, atau dalam menafsirkan tabel Support by AWS SDKs and tools berikut, lihat [Memahami halaman pengaturan panduan ini](#).

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

Konfigurasi fungsi ini dengan menggunakan yang berikut ini:

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
```

Support oleh AWS SDKs dan alat

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

SDK	Di	Catatan atau informasi lebih lanjut
AWS CLI v2	Ya	
SDK for C++	Tida	
SDK for Go V2 (1.x)	Tida	
SDK for Go 1.x (V1)	Tida	
SDK for Java 2.x	Parsi	Hanya AWS_METADATA_SERVICE_TIMEOUT didukung.
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 4.x	Tida	

SDK	Di Catatan atau informasi lebih lanjut
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
SDK para Swift	Tida
Alat untuk PowerShell V5	Tida
Alat untuk PowerShell V4	Tida

Titik akses Amazon S3

Note

Untuk bantuan dalam memahami tata letak halaman pengaturan, atau dalam menafsirkan tabel Support by AWS SDKs and tools berikut, lihat [Memahami halaman pengaturan panduan ini](#).

Layanan Amazon S3 menyediakan titik akses sebagai cara alternatif untuk berinteraksi dengan bucket Amazon S3. Titik akses memiliki kebijakan dan konfigurasi unik yang dapat diterapkan padanya, bukan langsung ke bucket. Dengan AWS SDKs, Anda dapat menggunakan access point Amazon Resource Names (ARNs) 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 selengkapnya tentang jalur akses Amazon S3 dan ARNs, lihat [Menggunakan titik akses](#) di Panduan Pengguna Amazon S3.

Konfigurasi fungsi ini dengan menggunakan yang berikut:

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.

Support oleh AWS SDKs dan alat

Berikut ini SDKs mendukung fitur dan pengaturan yang dijelaskan dalam topik ini. Setiap pengecualian sebagian dicatat. Setiap pengaturan properti sistem JVM didukung oleh AWS SDK untuk Java dan satu-satunya. AWS SDK untuk 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	Di	Catatan atau informasi lebih lanjut
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 4.x	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	
SDK para Swift	Tida	
Alat untuk PowerShell V5	Ya	Tidak mengikuti prioritas standar; nilai config file bersama lebih diutamakan daripada variabel lingkungan.
Alat untuk PowerShell V4	Ya	Tidak mengikuti prioritas standar; nilai config file bersama lebih diutamakan daripada variabel lingkungan.

Titik Akses Multi-Wilayah Amazon S3

Note

Untuk bantuan dalam memahami tata letak halaman pengaturan, atau dalam menafsirkan tabel Support by AWS SDKs and tools berikut, lihat [Memahami halaman pengaturan panduan ini](#).

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

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

Algoritma Sigv4a adalah implementasi penandatanganan yang digunakan untuk menandatangani permintaan Wilayah global. Algoritma ini diperoleh oleh SDK melalui ketergantungan pada [AWS Pustaka 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.

Support oleh AWS SDKs dan alat

Berikut ini SDKs mendukung fitur dan pengaturan yang dijelaskan dalam topik ini. Setiap pengecualian sebagian dicatat. Setiap pengaturan properti sistem JVM didukung oleh AWS SDK untuk Java dan satu-satunya. AWS SDK untuk 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 4.x	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 para Swift	Tida
Alat untuk PowerShell V5	Ya

SDK	Di	Catatan atau informasi lebih lanjut
Alat untuk PowerShell V4	Ya	

Otentikasi sesi S3 Express One Zone

Note

Untuk bantuan dalam memahami tata letak halaman pengaturan, atau dalam menafsirkan tabel Support by AWS SDKs and tools berikut, lihat [Memahami halaman pengaturan panduan ini](#).

S3 Express One Zone adalah kelas penyimpanan berkinerja tinggi Amazon S3 yang menyediakan latensi milidetik satu digit untuk data yang sering diakses. Saat Anda menggunakan bucket S3 Express One Zone, AWS SDKs dan alat secara otomatis menggunakan otentikasi berbasis sesi yang dioptimalkan untuk otorisasi permintaan data latensi rendah. Anda menggunakan token sesi dengan operasi Zonal (tingkat objek) untuk mendistribusikan latensi yang terkait dengan otorisasi atas sejumlah permintaan dalam satu sesi, mengurangi overhead otentikasi dan meningkatkan kinerja permintaan secara keseluruhan.

Bucket S3 Express One Zone menggunakan format penamaan tertentu yang menyertakan ID Availability Zone, seperti. `bucket-name--usw2-az1--x-s3` Saat SDK mendeteksi pola penamaan ini, SDK secara otomatis merutekan permintaan ke titik akhir S3 Express One Zone yang sesuai dan menerapkan alur otentikasi yang dioptimalkan. Autentikasi sesi membuat kredensial sementara khusus ember yang menyediakan akses latensi rendah ke bucket Anda dan di-cache serta di-refresh secara otomatis oleh SDK. Lihat [S3 Express One Zone](#) di Panduan Pengguna Amazon S3 untuk mempelajari lebih lanjut.

Secara default, otentikasi sesi diaktifkan untuk bucket S3 Express One Zone.

Konfigurasi fungsi ini dengan menggunakan yang berikut ini:

s3_disable_express_session_auth- Pengaturan AWS **config** file bersama, **AWS_S3_DISABLE_EXPRESS_SESSION_AUTH**- variabel lingkungan, **aws.disableS3ExpressAuth**- Properti sistem JVM: hanya Java/Kotlin

Mengontrol apakah otentikasi sesi S3 Express One Zone dinonaktifkan. Saat disetel ke `true`, SDK menggunakan otentikasi SiGv4 standar untuk bucket S3 Express One Zone alih-alih otentikasi sesi.

Nilai default: `false`

Nilai yang valid:

- **true**— Nonaktifkan otentikasi sesi S3 Express One Zone.
- **false**— Aktifkan otentikasi sesi S3 Express One Zone.

Contoh pengaturan nilai ini dalam `config` file:

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

Linux/macOS contoh pengaturan variabel lingkungan melalui baris perintah:

```
export AWS_S3_DISABLE_EXPRESS_SESSION_AUTH=true
```

Contoh Windows pengaturan variabel lingkungan melalui baris perintah:

```
setx AWS_S3_DISABLE_EXPRESS_SESSION_AUTH true
```

Support oleh AWS SDKs dan alat

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

SDK	Diduk	Catatan atau informasi lebih lanjut
AWS CLI v2	Tidak	
AWS CLI v1	Tidak	

SDK	Diduk	Catatan atau informasi lebih lanjut
SDK for C++	Ya	
SDK for Go V2 (1.x)	Ya	
SDK for Go 1.x (V1)	Tidak	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	Tidak	
SDK untuk 3.x JavaScript	Ya	
SDK untuk 2.x JavaScript	Tidak	
SDK para Kotlin	Ya	Properti sistem JVM adalah. <code>aws.s3DisableExpressionSessionAuth</code>
SDK for .NET 4.x	Ya	
SDK for .NET 3.x	Ya	
SDK for PHP 3.x	Ya	
SDK untuk Python (Boto3)	Tidak	
SDK for Ruby 3.x	Ya	
SDK untuk Rust	Ya	
SDK para Swift	Ya	
Alat untuk PowerShell V5	Ya	

SDK	Diduk	Catatan atau informasi lebih lanjut
Alat untuk PowerShell V4	Ya	

Skema otentikasi

Note

Untuk bantuan dalam memahami tata letak halaman pengaturan, atau dalam menafsirkan tabel Support by AWS SDKs and tools berikut, lihat [Memahami halaman pengaturan panduan ini](#).

AWS layanan mendukung beberapa skema otentikasi, seperti AWS Sigv4 (SigV4) dan Signature Version 4a (AWS Sigv4a). Secara default, SDKs pilih skema otentikasi berdasarkan definisi model layanan dan prioritaskan skema yang memberikan kompatibilitas terbaik. Namun, Anda dapat mengonfigurasi skema otentikasi pilihan Anda untuk mengoptimalkan persyaratan tertentu.

Tidak seperti SigV4, permintaan yang ditandatangani dengan Sigv4a valid dalam beberapa wilayah. Wilayah AWS Sigv4a menyediakan ketersediaan yang ditingkatkan melalui penandatanganan permintaan lintas wilayah, yang memungkinkan failover otomatis ke wilayah cadangan selama gangguan regional. Ini sangat bermanfaat untuk layanan global seperti AWS Identity and Access Management atau Amazon CloudFront.

Untuk informasi selengkapnya tentang dua skema autentikasi ini, lihat [Versi AWS Tanda Tangan 4 untuk permintaan API](#) di Panduan Pengguna IAM.

Konfigurasi fungsi ini dengan menggunakan yang berikut ini:

auth_scheme_preference- Pengaturan AWS **config** file bersama, **AWS_AUTH_SCHEME_PREFERENCE**- variabel lingkungan, **aws.authSchemePreference**- Properti sistem JVM: hanya Java/Kotlin

Menentukan daftar dipisahkan koma skema otentikasi pilihan dalam urutan prioritas. Ketika layanan mendukung beberapa skema otentikasi, SDK mencoba menggunakan skema dari daftar ini dalam urutan yang ditentukan, kembali ke perilaku default jika tidak ada skema pilihan yang tersedia.

Nilai default: Tidak ada.

Nilai yang valid: Daftar dipisahkan koma dari satu atau lebih dari berikut ini:

- **sigv4**— Versi Tanda Tangan 4 (kinerja tercepat, wilayah tunggal)
- **sigv4a**- Signature Version 4a (ketersediaan yang ditingkatkan, dukungan lintas wilayah, memiliki kinerja penandatanganan yang lebih lambat daripada SigV4)
- **httpBearerAuth**— Otentikasi token Pembawa HTTP

Karakter spasi dan tab di antara nama skema diabaikan.

Contoh pengaturan nilai ini dalam `config` file untuk memilih Sigv4a:

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

sigv4a_signing_region_set- Pengaturan AWS `config` file bersama, **AWS_SIGV4A_SIGNING_REGION_SET**- variabel lingkungan

Menentukan daftar dipisahkan koma untuk penandatanganan multi-wilayah Wilayah AWS Sigv4a. Ini digunakan sebagai Region default yang ditetapkan untuk permintaan jika Sigv4a adalah skema otentikasi yang dipilih.

Nilai default: Ditentukan oleh permintaan.

Nilai yang valid: Daftar dipisahkan koma dari Wilayah AWS Karakter spasi dan tab antar Wilayah diabaikan.

Support oleh AWS SDKs dan alat

Berikut ini SDKs mendukung fitur dan pengaturan yang dijelaskan dalam topik ini. Setiap pengecualian sebagian dicatat. Setiap pengaturan properti sistem JVM didukung oleh AWS SDK untuk Java dan satu-satunya. AWS SDK untuk 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 4.x	Ya	
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	Ya	
SDK para Swift	Ya	
Alat untuk PowerShell V5	Ya	
Alat untuk PowerShell V4	Tida	

Wilayah AWS

Note

Untuk bantuan dalam memahami tata letak halaman pengaturan, atau dalam menafsirkan tabel Support by AWS SDKs and tools berikut, lihat [Memahami halaman pengaturan panduan ini](#).

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

Dengan Wilayah AWS, Anda dapat mengakses Layanan AWS yang secara fisik berada di wilayah geografis tertentu. Ini dapat berguna untuk menjaga data dan aplikasi Anda berjalan dekat dengan tempat Anda dan pengguna Anda akan mengaksesnya. Wilayah memberikan toleransi kesalahan, stabilitas, serta ketahanan, dan juga dapat mengurangi latensi. Dengan Wilayah, Anda dapat membuat sumber daya redundan yang tetap tersedia dan tidak terpengaruh oleh pemadaman Regional.

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

Referensi Umum AWS Berisi informasi tentang hal berikut:

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

Membuat klien layanan

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

Jika tidak ada Region yang secara eksplisit ditentukan untuk klien dalam kode itu sendiri, klien default menggunakan Region yang diatur melalui pengaturan berikut. `region` Namun, Region aktif untuk klien dapat secara eksplisit diatur untuk setiap objek klien individu. Mengatur Wilayah dengan cara ini lebih diutamakan daripada pengaturan global apa pun untuk klien layanan tertentu. Wilayah alternatif ditentukan selama instantiasi klien tersebut, khusus untuk 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 SDKs memiliki objek “konfigurasi” yang tersedia untuk mengatur Wilayah default dari dalam kode aplikasi. Untuk detailnya, lihat panduan developer AWS SDK spesifik Anda.

Support oleh AWS SDKs dan alat

Berikut ini SDKs mendukung fitur dan pengaturan yang dijelaskan dalam topik ini. Setiap pengecualian sebagian dicatat. Setiap pengaturan properti sistem JVM didukung oleh AWS SDK untuk Java dan satu-satunya. AWS SDK untuk 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 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 4.x	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
SDK para Swift	Ya

SDK	Di	Catatan atau informasi lebih lanjut
Alat untuk PowerShell V5	Ya	
Alat untuk PowerShell V4	Ya	

AWS STS Titik akhir regional

Note

Untuk bantuan dalam memahami tata letak halaman pengaturan, atau dalam menafsirkan tabel Support by AWS SDKs and tools berikut, lihat [Memahami halaman pengaturan panduan ini](#).

AWS Security Token Service (AWS STS) tersedia baik sebagai layanan global maupun Regional. Beberapa AWS SDKs dan CLIs menggunakan endpoint layanan global (`https://sts.amazonaws.com`) secara default, sementara beberapa menggunakan endpoint layanan Regional (`https://sts.{region_identifier}.{partition_domain}`). Di Wilayah yang [diaktifkan secara default](#), permintaan ke titik akhir AWS STS global secara otomatis ditayangkan di Wilayah yang sama tempat permintaan berasal. Di Wilayah keikutsertaan, permintaan ke titik akhir AWS STS global dilayani oleh satu Wilayah AWS, US East (Virginia N.). Untuk informasi selengkapnya tentang AWS STS titik akhir, lihat [Titik Akhir](#) di Referensi AWS Security Token Service API atau [Kelola AWS STS Wilayah AWS dalam AWS Identity and Access Management](#) Panduan Pengguna.

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

Note

AWS telah membuat perubahan pada AWS Security Token Service (AWS STS) titik akhir global (`https://sts.amazonaws.com`) di Wilayah yang [diaktifkan secara default](#) untuk meningkatkan ketahanan dan kinerjanya. AWS STS permintaan ke titik akhir global secara


otomatis disajikan Wilayah AWS sama dengan beban kerja Anda. Perubahan ini tidak akan diterapkan ke Wilayah keikutsertaan. Kami menyarankan Anda menggunakan titik akhir AWS STS regional yang sesuai. Untuk informasi selengkapnya, lihat [perubahan titik akhir AWS STS global](#) di Panduan AWS Identity and Access Management Pengguna.

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

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

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:`regional`, lihat pengecualian dalam tabel berikut.

 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`.
- **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 `Regionalsts.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
```

Support oleh AWS SDKs dan alat

Note

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

Tabel berikut ini merangkum, untuk SDK atau alat Anda:

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

Tabel ini juga menggunakan nilai-nilai berikut:

- Titik akhir global: `https://sts.amazonaws.com`.
- Titik akhir regional: Berdasarkan konfigurasi yang [Wilayah AWS](#) digunakan oleh aplikasi Anda.
- **us-east-1**(Regional): Menggunakan titik akhir `us-east-1` Wilayah tetapi dengan token sesi yang lebih panjang daripada permintaan global biasa.

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

SDK		Nilai pengaturan default	Target klien layanan default STS Endpoint	Perilaku fallback klien layanan	Catatan atau informasi lebih lanjut
SDK untuk 2.x JavaScript	Y	legacy	Titik akhir global	Titik akhir global	
SDK para Kotlin	Tid	N/A	Titik akhir Regional	Titik akhir global	
SDK for .NET 4.x	Tid	N/A	Titik akhir Regional	us-east-1 (Regional)	
SDK for .NET 3.x	Y	regional	Titik akhir global	Titik akhir global	
SDK for PHP 3.x	Y	regional	Titik akhir global	Kegagalan permintaan	
SDK untuk Python (Boto3)	Y	regional	Titik akhir global	Titik akhir global	
SDK for Ruby 3.x	Y	regional	Titik akhir Regional	Kegagalan permintaan	
SDK untuk Rust	Tid	N/A	Titik akhir Regional	Kegagalan permintaan	
SDK para Swift	Tid	N/A	Titik akhir Regional	Kegagalan permintaan	
Alat untuk PowerShell V5	Y	regional	Titik akhir global	Titik akhir global	

SDK	Nilai pengaturan default	Target klien layanan default STS Endpoint	Perilaku fallback klien layanan	Catatan atau informasi lebih lanjut
Alat untuk PowerShell V4	Y regional	Titik akhir global	Titik akhir global	

Perlindungan Integritas Data untuk Amazon S3

Note

Untuk bantuan dalam memahami tata letak halaman pengaturan, atau dalam menafsirkan tabel Support by AWS SDKs and tools berikut, lihat [Memahami halaman pengaturan panduan ini](#).

Untuk beberapa waktu, AWS SDKs telah mendukung pemeriksaan integritas data saat mengunggah data ke atau mengunduh data dari Amazon Simple Storage Service. Sebelumnya, pemeriksaan ini adalah opt-in. Sekarang, kami telah mengaktifkan pemeriksaan ini secara default, menggunakan algoritma berbasis CRC seperti CRC32 atau NVME. CRC64 Meskipun setiap SDK atau alat memiliki algoritma default, Anda dapat memilih algoritma yang berbeda. Anda juga dapat terus menyediakan checksum yang telah dihitung sebelumnya secara manual untuk diunggah jika Anda mau. Perilaku konsisten di seluruh unggahan, unggahan multibagian, unduhan, dan mode enkripsi menyederhanakan pemeriksaan integritas sisi klien.

Versi terbaru dari kami AWS SDKs dan AWS CLI secara otomatis menghitung [checksum berbasis pemeriksaan redundansi siklik \(CRC\)](#) untuk setiap unggahan dan mengirimkannya ke Amazon S3. Amazon S3 secara independen menghitung checksum di sisi server dan memvalidasinya terhadap nilai yang diberikan sebelum menyimpan objek dan checksumnya secara tahan lama dalam metadata objek. Dengan menyimpan checksum dalam metadata di samping objek, ketika objek diunduh, checksum yang sama dapat secara otomatis dikembalikan dan digunakan untuk memvalidasi unduhan juga. Anda juga dapat memverifikasi checksum yang disimpan dalam metadata objek kapan saja.

Untuk mempelajari lebih lanjut tentang operasi checksum, unggahan multibagian, atau daftar algoritme checksum yang didukung, lihat [Memeriksa integritas objek di Amazon S3 di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon](#).

Unggahan multipart:

Amazon S3 juga memberi pengembang checksum objek lengkap yang konsisten di seluruh unggahan satu bagian dan multibagian.

Saat mengunggah file di beberapa bagian, SDKs menghitung checksum untuk setiap bagian. Amazon S3 menggunakan checksum ini untuk memverifikasi integritas setiap bagian melalui API. `UploadPart` Selain itu, Amazon S3 memvalidasi seluruh ukuran file dan checksum saat Anda memanggil API. `CompleteMultipartUpload`

Jika SDK Anda memiliki Manajer Transfer Amazon S3 untuk membantu unggahan multibagian, checksum akan divalidasi untuk komponen menggunakan algoritme default khusus SDK yang ditemukan dalam tabel. [Support oleh AWS SDKs dan alat](#) Anda dapat ikut serta ke checksum objek lengkap dengan mengatur pengaturan `checksum_type` ke `FULL_OBJECT` atau dengan memilih untuk menggunakan algoritma CRC64 NVME.

Jika Anda menggunakan versi SDK yang lebih lama atau AWS CLI:

Jika aplikasi Anda menggunakan versi SDK atau alat sebelum Desember 2024, Amazon S3 masih menghitung CRC64 checksum NVME pada objek baru dan menyimpannya di metadata objek untuk referensi future. Anda nantinya dapat membandingkan CRC yang disimpan dengan CRC yang dihitung di sisi Anda dan memverifikasi transmisi jaringan sudah benar. Selain itu, Anda masih dapat memperluas perlindungan integritas secara manual dengan menyediakan checksum yang telah dihitung sebelumnya dengan `UploadPart` permintaan `PutObject` atau Anda, yang merupakan teknik standar untuk menangani ini dalam versi yang lebih lama.

Konfigurasi fungsi ini dengan menggunakan yang berikut ini:

`request_checksum_calculation`- Pengaturan AWS `config` file bersama, **`AWS_REQUEST_CHECKSUM_CALCULATION`**- variabel lingkungan, **`aws.requestChecksumCalculation`**- Properti sistem JVM: hanya Java/Kotlin

Secara default, pengguna memilih untuk menghitung checksum permintaan saat mengirim permintaan. Pengguna dapat memilih salah satu [algoritma checksum yang tersedia](#) sebagai bagian dari membangun permintaan. Jika tidak, algoritma default khusus SDK digunakan. Lihat [Support oleh AWS SDKs dan alat](#) tabel untuk algoritme default untuk setiap SDK atau alat.

Nilai default: `WHEN_SUPPORTED`

Nilai yang valid:

- **WHEN_SUPPORTED**— Validasi Checksum dilakukan pada semua muatan permintaan saat didukung oleh operasi API, seperti transfer data ke Amazon S3.
- **WHEN_REQUIRED**— Validasi Checksum dilakukan hanya bila diperlukan oleh operasi API.

response_checksum_validation- Pengaturan AWS `config` file

bersama, **AWS_RESPONSE_CHECKSUM_VALIDATION**- variabel lingkungan,

aws.responseChecksumValidation- Properti sistem JVM: hanya Java/Kotlin

Secara default, pengguna ikut serta dalam validasi checksum respons saat mengirim permintaan. Checksum dihitung untuk muatan respons dan dibandingkan dengan header respons checksum. Jika validasi checksum gagal, kesalahan akan muncul ke pengguna saat payload dibaca.

Header respons checksum juga menunjukkan algoritma untuk checksum. Klien Amazon S3 mencoba memvalidasi checksum respons untuk semua operasi API Amazon S3 yang mendukung checksum. Namun, jika SDK belum menerapkan algoritma checksum yang ditentukan maka validasi ini dilewati.

Nilai default: `WHEN_SUPPORTED`

Nilai yang valid:

- **WHEN_SUPPORTED**— Validasi Checksum dilakukan pada semua muatan respons saat didukung oleh operasi API, seperti transfer data ke Amazon S3.
- **WHEN_REQUIRED**— Validasi Checksum dilakukan hanya ketika didukung oleh operasi API dan pemanggil telah secara eksplisit mengaktifkan checksum untuk operasi. Misalnya, ketika `GetObject` API Amazon S3 dipanggil dan `ChecksumMode` parameter disetel ke diaktifkan.

Support oleh AWS SDKs dan alat

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

Note

Dalam tabel berikut, 'CRT' mengacu pada [AWS Pustaka Runtime Umum \(CRT\)](#) dan mungkin perlu menambahkan ketergantungan tambahan ke proyek Anda.

SDK	Diduk	Algoritma checksum default	Algoritma checksum yang didukung	Catatan atau informasi lebih lanjut
AWS CLI v2	Ya	CRC64NVME	CRC64NVME,, CRC32 C CRC32,, SHA1 SHA256	Untuk AWS CLI v1, algoritma default dan algoritma yang didukung akan identik dengan Python (Boto3).
SDK for C++	Ya	CRC64NVME	CRC64NVME,, CRC32 C CRC32,, SHA1 SHA256	
SDK for Go V2 (1.x)	Ya	CRC32	CRC64NVME,, CRC32 C CRC32,, SHA1 SHA256	
SDK for Go 1.x (V1)	Tidak			
SDK for Java 2.x	Ya	CRC32	CRC64NVME (hanya melalui CRT),, C, CRC32, CRC32 SHA1 SHA256	
SDK for Java 1.x	Tidak			
SDK untuk 3.x JavaScript	Ya	CRC32	CRC32, CRC32 C, SHA1, SHA256	

SDK	Diduk	Algoritma checksum default	Algoritma checksum yang didukung	Catatan atau informasi lebih lanjut
SDK untuk 2.x JavaScript	Tidak			
SDK para Kotlin	Ya	CRC32	CRC32, CRC32 C, SHA1, SHA256	
SDK for .NET 4.x	Ya	CRC32	CRC32, CRC32 C, SHA1, SHA256	
SDK for .NET 3.x	Ya	CRC32	CRC32, CRC32 C, SHA1, SHA256	
SDK for PHP 3.x	Ya	CRC32	CRC32, CRC32 C (hanya melalui CRT),, SHA1 SHA256	aws_crt ekstensi diperlukan untuk menggunakan CRC32 C.
SDK untuk Python (Boto3)	Ya	CRC32	CRC64NVME (hanya melalui CRT), CRC32, CRC32 C (hanya melalui CRT),, SHA1 SHA256	
SDK for Ruby 3.x	Ya	CRC32	CRC64NVME (hanya melalui CRT), CRC32, CRC32 C (hanya melalui CRT),, SHA1 SHA256	
SDK untuk Rust	Ya	CRC32	CRC64NVME,, CRC32 C CRC32,, SHA1 SHA256	
SDK para Swift	Ya	CRC32	CRC64NVME,, CRC32 C CRC32,, SHA1 SHA256	Ketergantungan CRT diperlukan untuk semua algoritma.

SDK	Diduk	Algoritma checksum default	Algoritma checksum yang didukung	Catatan atau informasi lebih lanjut
Alat untuk PowerShell V5	Ya	CRC32	CRC32, CRC32 C, SHA1, SHA256	
Alat untuk PowerShell V4	Ya	CRC32	CRC32, CRC32 C, SHA1, SHA256	

Dual-stack dan titik akhir FIPS

Note

Untuk bantuan dalam memahami tata letak halaman pengaturan, atau dalam menafsirkan tabel Support by AWS SDKs and tools berikut, lihat [Memahami halaman pengaturan panduan ini](#).

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 keduanya IPv4 dan IPv6 lalu lintas, lihat Menggunakan [titik akhir tumpukan ganda Amazon S3](#) di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon. Titik akhir dual-stack tersedia untuk beberapa layanan di beberapa wilayah.

Nilai default: `false`

Nilai yang valid:

- **true** 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 divalidasi terhadap FIPS 140. 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 Amazon API. 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.

Support oleh AWS SDKs dan alat

Berikut ini SDKs mendukung fitur dan pengaturan yang dijelaskan dalam topik ini. Setiap pengecualian sebagian dicatat. Setiap pengaturan properti sistem JVM didukung oleh AWS SDK untuk Java dan satu-satunya. AWS SDK untuk 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	Tida	
SDK untuk 3.x JavaScript	Ya	
SDK untuk 2.x JavaScript	Ya	
SDK para Kotlin	Ya	
SDK for .NET 4.x	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 para Swift	Ya	
Alat untuk PowerShell V5	Ya	
Alat untuk PowerShell V4	Ya	

Penemuan titik akhir

Note

Untuk bantuan dalam memahami tata letak halaman pengaturan, atau dalam menafsirkan tabel Support by AWS SDKs and tools berikut, lihat [Memahami halaman pengaturan panduan ini](#).

SDKs menggunakan penemuan titik akhir untuk mengakses titik akhir layanan (URLs untuk mengakses berbagai sumber daya), sambil tetap mempertahankan fleksibilitas AWS untuk mengubah sesuai kebutuhan URLs. 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.

Support oleh AWS SDKs dan alat

Berikut ini SDKs mendukung fitur dan pengaturan yang dijelaskan dalam topik ini. Setiap pengecualian sebagian dicatat. Setiap pengaturan properti sistem JVM didukung oleh AWS SDK untuk Java dan satu-satunya. AWS SDK untuk 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 4.x	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	Parsi	Didukung hanya untuk Timestream.
SDK para Swift	Tida	
Alat untuk PowerShell V5	Ya	
Alat untuk PowerShell V4	Ya	

Pengaturan konfigurasi umum

Note

Untuk bantuan dalam memahami tata letak halaman pengaturan, atau dalam menafsirkan tabel Support by AWS SDKs and tools berikut, lihat [Memahami halaman pengaturan panduan ini](#).

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

Konfigurasi 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 SSL/TLS koneksi.

Nilai default: tidak ada

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

Contoh pengaturan nilai ini dalam config file:

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

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

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

Linux/macOS contoh pengaturan variabel lingkungan melalui baris perintah:

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

Contoh Windows pengaturan variabel lingkungan melalui baris perintah:

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

output- Pengaturan AWS **config** file bersama

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

Nilai default: json

Nilai yang valid:

- **json**— Output diformat sebagai string [JSON](#).
- **yaml**— Output diformat sebagai string [YAMM](#).
- **yaml-stream**— Output dialirkan dan diformat sebagai string [YAMG](#). 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.

Support oleh AWS SDKs dan alat

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

SDK	Di Catatan atau informasi lebih lanjut
AWS CLI v2	Parsi <code>api_versions</code> tidak didukung.
SDK for C++	Ya

SDK	Di	Catatan atau informasi lebih lanjut
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 para Kotlin	Tida	
SDK for .NET 4.x	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	
SDK para Swift	Tida	
Alat untuk PowerShell V5	Tida	
Alat untuk PowerShell V4	Tida	

Injeksi awalan host

Note

Untuk bantuan dalam memahami tata letak halaman pengaturan, atau dalam menafsirkan tabel Support by AWS SDKs and tools berikut, lihat [Memahami halaman pengaturan panduan ini](#).

Injeksi awalan host adalah fitur yang AWS SDKs secara otomatis menambahkan awalan ke nama host titik akhir layanan untuk operasi API tertentu. Awalan ini dapat berupa string statis atau nilai dinamis yang menyertakan data dari parameter permintaan Anda.

Misalnya, saat menggunakan Amazon Simple Storage Service untuk melakukan tindakan pada objek atau bucket Amazon S3, SDK menggantikan nama bucket dan Akun AWS ID Anda di titik akhir API final.

Meskipun perilaku ini diperlukan untuk titik akhir AWS layanan normal, ini dapat menyebabkan masalah saat menggunakan titik akhir khusus seperti titik akhir VPC atau alat pengujian lokal. Dalam kasus ini, Anda mungkin perlu menonaktifkan injeksi awalan host.

Konfigurasi fungsi ini dengan menggunakan yang berikut ini:

disable_host_prefix_injection- Pengaturan AWS **config** file bersama, **AWS_DISABLE_HOST_PREFIX_INJECTION**- variabel lingkungan, **aws.disableHostPrefixInjection**- Properti sistem JVM: hanya Java/Kotlin

Pengaturan ini mengontrol apakah SDK atau alat akan memodifikasi nama host endpoint dengan mengawali awalan host seperti yang didefinisikan dalam objek atau variabel klien SDK Anda.

Nilai default: `false`

Nilai yang valid:

- **true**— Nonaktifkan injeksi awalan host. SDK tidak akan mengubah nama host endpoint.
- **false**- Aktifkan injeksi awalan host. SDK akan menambahkan awalan host ke nama host endpoint.

Contoh pengaturan nilai ini dalam `config` file:

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

Linux/macOS contoh pengaturan variabel lingkungan melalui baris perintah:

```
export AWS_DISABLE_HOST_PREFIX_INJECTION=true
```

Contoh Windows pengaturan variabel lingkungan melalui baris perintah:

```
setx AWS_DISABLE_HOST_PREFIX_INJECTION true
```

Contoh injeksi awalan host

Tabel contoh berikut menunjukkan bagaimana SDKs memodifikasi titik akhir saat injeksi awalan host diaktifkan dan dinonaktifkan.

- Awalan host: Template string properti awalan host ditetapkan pada objek klien SDK atau variabel dalam kode.
- Input: Masukan tambahan ditetapkan pada objek klien SDK atau variabel dalam kode.
- Titik akhir klien: Titik akhir turunan klien.
- Nilai pengaturan: Nilai yang diselesaikan untuk pengaturan sebelumnya.
- Titik akhir yang dihasilkan: Titik akhir yang dihasilkan yang digunakan klien SDK untuk melakukan panggilan API.

Awalan host	Masukan	Titik akhir klien	Menetapkan nilai	Titik akhir yang dihasilkan
"data."	{}	"https://service.us-west-2.amazonaws.com"	false	"https://data.service.us-west-2.amazonaws.com"
"{Bucket} - {AccountId}."	Ember: "amzn-s3-demo-bucket1",:"	"https://service.us-west-2.amazonaws.com"	false	"https://amzn-s3-demo-bucket1-123456789012.se"

Awalan host	Masukan	Titik akhir klien	Menetapkan nilai	Titik akhir yang dihasilkan
	123456789012" AccountId			rvic.us- west-2.am zonaws.com"
"data."	{	"https://override. us-west-2 .amazonaw s.com"(se bagai titik akhir override)	true	"https://override. us-west-2 .amazonaw s.com"

Support oleh AWS SDKs dan alat

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

SDK	Di	Catatan atau informasi lebih lanjut
AWS CLI v2	Ya	
SDK for C++	Tida	Pengaturan tidak didukung, tetapi dapat dikonfigurasi dalam kode pada klien menggunakan: enableHostPrefixInjection .
SDK for Go V2 (1.x)	Tida	Dapat dinonaktifkan menggunakan middleware .
SDK for Go 1.x (V1)	Tida	
SDK for Java 2.x	Tida	Pengaturan tidak didukung, tetapi dapat dikonfigurasi dalam kode pada klien menggunakan: SdkAdvancedClientOption.DISABLE_HOST_PREFIX_INJECTION .

SDK	Di	Catatan atau informasi lebih lanjut
SDK for Java 1.x	Tida	Pengaturan tidak didukung, tetapi dapat dikonfigurasi dalam kode pada klien menggunakan: withDisableHostPrefixInjection .
SDK untuk 3.x JavaScript	Tida	Pengaturan tidak didukung, tetapi dapat dikonfigurasi dalam kode pada klien menggunakan: disableHostPrefix .
SDK untuk 2.x JavaScript	Tida	Pengaturan tidak didukung, tetapi dapat dikonfigurasi dalam kode pada klien menggunakan: hostPrefixEnabled .
SDK para Kotlin	Tida	
SDK for .NET 4.x	Tida	Pengaturan tidak didukung, tetapi dapat dikonfigurasi dalam kode pada klien menggunakan: DisableHostPrefixInjection .
SDK for .NET 3.x	Tida	Pengaturan tidak didukung, tetapi dapat dikonfigurasi dalam kode pada klien menggunakan: DisableHostPrefixInjection .
SDK for PHP 3.x	Tida	Pengaturan tidak didukung, tetapi dapat dikonfigurasi dalam kode pada klien menggunakan: disable_host_prefix_injection .
SDK untuk Python (Boto3)	Ya	Dapat dikonfigurasi dalam kode pada klien menggunakan: inject_host_prefix .
SDK for Ruby 3.x	Tida	Pengaturan tidak didukung, tetapi dapat dikonfigurasi dalam kode pada klien menggunakan: disable_host_prefix_injection .
SDK untuk Rust	Tida	
SDK para Swift	Tida	

SDK	Di	Catatan atau informasi lebih lanjut
Alat untuk PowerShell V5	Tida	Pengaturan tidak didukung, tetapi dapat disertakan dalam cmdlet tertentu menggunakan parameter. <code>-ClientConfig @{DisableHostPrefixInjection = \$true}</code>
Alat untuk PowerShell V4	Tida	Pengaturan tidak didukung, tetapi dapat disertakan dalam cmdlet tertentu menggunakan parameter. <code>-ClientConfig @{DisableHostPrefixInjection = \$true}</code>

Klien IMDS

Note

Untuk bantuan dalam memahami tata letak halaman pengaturan, atau dalam menafsirkan tabel Support by AWS SDKs and tools berikut, lihat [Memahami halaman pengaturan panduan ini](#).

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

Konfigurasi fungsi ini dengan menggunakan yang berikut ini:

retries- anggota objek konfigurasi klien

Jumlah upaya coba lagi tambahan untuk setiap permintaan yang gagal.

Nilai default: 3

Nilai yang valid: Angka lebih besar dari 0.

port- anggota objek konfigurasi klien

Port untuk titik akhir.

Nilai default: 80

Nilai yang valid: Nomor.

token_ttl- anggota objek konfigurasi klien

TTL 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 SDKs. 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: I/O Aliran 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.

Support oleh AWS SDKs dan alat

Berikut ini SDKs mendukung fitur dan pengaturan yang dijelaskan dalam topik ini. Setiap pengecualian sebagian dicatat. Setiap pengaturan properti sistem JVM didukung oleh AWS SDK untuk Java dan satu-satunya. AWS SDK untuk 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)	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	Tida
SDK for .NET 4.x	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 para Swift	Ya
Alat untuk PowerShell V5	Ya
Alat untuk PowerShell V4	Ya

Coba lagi perilaku

Note

Untuk bantuan dalam memahami tata letak halaman pengaturan, atau dalam menafsirkan tabel Support by AWS SDKs and tools berikut, lihat [Memahami halaman pengaturan panduan ini](#).

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

Konfigurasi fungsi ini dengan menggunakan yang berikut:

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

Nilai yang valid:

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

`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.

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.
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 Region-and-table pasangan. 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

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

- Bucket token coba lagi digunakan untuk menentukan apakah SDK harus menonaktifkan sementara percobaan ulang untuk melindungi layanan hulu dan hilir selama pemadaman. Token diperoleh dari bucket sebelum percobaan ulang dicoba, dan token dikembalikan ke bucket saat permintaan berhasil. Jika bucket kosong saat percobaan ulang dicoba, SDK tidak akan mencoba lagi permintaan tersebut.
- Bucket token rate permintaan hanya digunakan dalam mode adaptive coba lagi untuk menentukan tingkat pengiriman permintaan. Token diperoleh dari bucket sebelum permintaan dikirim, dan token dikembalikan ke bucket pada tingkat yang ditentukan secara dinamis berdasarkan respons pelambatan yang dikembalikan oleh layanan.

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

```
MakeSDKRequest() {
  attempts = 0
  loop {
    GetSendToken()
    response = SendHTTPRequest()
    RequestBookkeeping(response)
    if not Retryable(response)
      return response
    attempts += 1
    if attempts >= MAX_ATTEMPTS:
      return response
    if not HasRetryQuota(response)
      return response
    delay = ExponentialBackoff(attempts)
    sleep(delay)
  }
}
```

```
}
```

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

GetSendToken:

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

SendHttpRequest:

Langkah ini mengirimkan permintaan ke AWS. Sebagian besar AWS SDKs menggunakan perpustakaan HTTP yang menggunakan kumpulan koneksi untuk menggunakan kembali koneksi yang ada saat membuat permintaan HTTP. Umumnya, koneksi digunakan kembali jika permintaan gagal karena kesalahan pelambatan tetapi tidak jika permintaan gagal karena kesalahan sementara.

RequestBookkeeping:

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

Retryable:

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

- Kode 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:

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

HasRetryQuota

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

ExponentialBackoff

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

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

Dalam algoritma sebelumnya, nilai-nilai berikut berlaku:

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

`r` = 2

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

Support oleh AWS SDKs dan alat

Berikut ini SDKs mendukung fitur dan pengaturan yang dijelaskan dalam topik ini. Setiap pengecualian sebagian dicatat. Setiap pengaturan properti sistem JVM didukung oleh AWS SDK untuk Java dan satu-satunya. AWS SDK untuk 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	Di	Catatan atau informasi lebih lanjut
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 4.x	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 para Swift	Ya	
Alat untuk PowerShell V5	Ya	
Alat untuk PowerShell V4	Ya	

Minta kompresi

Note

Untuk bantuan dalam memahami tata letak halaman pengaturan, atau dalam menafsirkan tabel Support by AWS SDKs and tools berikut, lihat [Memahami halaman pengaturan panduan ini](#).

AWS SDKs dan alat dapat secara otomatis mengompres 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:

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.

Support oleh AWS SDKs dan alat

Berikut ini SDKs mendukung fitur dan pengaturan yang dijelaskan dalam topik ini. Setiap pengecualian sebagian dicatat. Setiap pengaturan properti sistem JVM didukung oleh AWS SDK untuk Java dan satu-satunya. AWS SDK untuk 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 4.x	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	
SDK para Swift	Tida	
Alat untuk PowerShell V5	Ya	
Alat untuk PowerShell V4	Ya	

Titik akhir khusus layanan

Note

Untuk bantuan dalam memahami tata letak halaman pengaturan, atau dalam menafsirkan tabel Support by AWS SDKs and tools berikut, lihat [Memahami halaman pengaturan panduan ini](#).

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:

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 titik akhir khusus layanan untuk Amazon URLs S3 dan AWS Elastic Beanstalk 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. A 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.

Support oleh AWS SDKs dan alat

Berikut ini SDKs mendukung fitur dan pengaturan yang dijelaskan dalam topik ini. Setiap pengecualian sebagian dicatat. Setiap pengaturan properti sistem JVM didukung oleh AWS SDK untuk Java dan satu-satunya. AWS SDK untuk 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	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 4.x	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 para Swift	Ya	
Alat untuk PowerShell V5	Ya	
Alat untuk PowerShell V4	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	Keterangan
	Ki Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE> pe la ur fil be Al co
AccessAnalyzer	ac AWS_ENDPOINT_URL_ACCESSANALYZER ly
Account	ac AWS_ENDPOINT_URL_ACCOUNT
ACM	ac AWS_ENDPOINT_URL_ACM
ACM PCA	ac AWS_ENDPOINT_URL_ACM_PCA
Alexa For Business	af AWS_ENDPOINT_URL_ALEXA_FOR_BUSINESS _l
amp	ar AWS_ENDPOINT_URL_AMP
Amplify	ar AWS_ENDPOINT_URL_AMPLIFY
AmplifyBackend	ar AWS_ENDPOINT_URL_AMPLIFYBACKEND cl
AmplifyUIBuilder	ar AWS_ENDPOINT_URL_AMPLIFYUIBUILDER bt
API Gateway	ag AWS_ENDPOINT_URL_API_GATEWAY ay
ApiGatewayManagemen tApi	ag AWS_ENDPOINT_URL_APIGATEWAYMANAGEMENTAPI yr nt

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

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

serviceId	Ki Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE>
Backup Gateway	b: AWS_ENDPOINT_URL_BACKUP_GATEWAY t:
BackupStorage	b: AWS_ENDPOINT_URL_BACKUPSTORAGE r:
Batch	b: AWS_ENDPOINT_URL_BATCH
BCM Data Exports	b: AWS_ENDPOINT_URL_BCM_DATA_EXPORTS e:
Bedrock	b: AWS_ENDPOINT_URL_BEDROCK
Bedrock Agent	b: AWS_ENDPOINT_URL_BEDROCK_AGENT g:
Bedrock Agent Runtime	b: AWS_ENDPOINT_URL_BEDROCK_AGENT_RUNTIME g: i:
Bedrock Runtime	b: AWS_ENDPOINT_URL_BEDROCK_RUNTIME u:
billingconductor	b: AWS_ENDPOINT_URL_BILLINGCONDUCTOR n:
Braket	b: AWS_ENDPOINT_URL_BRAKET
Budgets	b: AWS_ENDPOINT_URL_BUDGETS

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

serviceId	Ki Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE> pe la ur fil be Al co
CleanRoomsML	c: AWS_ENDPOINT_URL_CLEANROOMSML sr
Cloud9	c: AWS_ENDPOINT_URL_CLOUD9
CloudControl	c: AWS_ENDPOINT_URL_CLOUDCONTROL r()
CloudDirectory	c: AWS_ENDPOINT_URL_CLOUDDIRECTORY cl
CloudFormation	c: AWS_ENDPOINT_URL_CLOUDFORMATION at
CloudFront	c: AWS_ENDPOINT_URL_CLOUDFRONT t
CloudFront 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

serviceId	Ki Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE> pe la ur fil be A co
CloudSearch Domain	c: AWS_ENDPOINT_URL_CLOUDSEARCH_DOMAIN cl
CloudTrail	c: AWS_ENDPOINT_URL_CLOUDTRAIL l
CloudTrail Data	c: AWS_ENDPOINT_URL_CLOUDTRAIL_DATA l_
CloudWatch	c: AWS_ENDPOINT_URL_CLOUDWATCH h
codeartifact	co AWS_ENDPOINT_URL_CODEARTIFACT a
CodeBuild	co AWS_ENDPOINT_URL_CODEBUILD
CodeCatalyst	co AWS_ENDPOINT_URL_CODECATALYST y:
CodeCommit	co AWS_ENDPOINT_URL_CODECOMMIT t
CodeDeploy	co AWS_ENDPOINT_URL_CODEDEPLOY y
CodeGuru Reviewer	co AWS_ENDPOINT_URL_CODEGURU_REVIEWER I

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

serviceId	Kategori	Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE>
Comprehend	comprehend	AWS_ENDPOINT_URL_COMPREHEND
ComprehendMedical	comprehendmedical	AWS_ENDPOINT_URL_COMPREHENDMEDICAL
Compute Optimizer	computeoptimizer	AWS_ENDPOINT_URL_COMPUTE_OPTIMIZER
Config Service	config	AWS_ENDPOINT_URL_CONFIG_SERVICE
Connect	connect	AWS_ENDPOINT_URL_CONNECT
Connect Contact Lens	connectcontactlens	AWS_ENDPOINT_URL_CONNECT_CONTACT_LENS
ConnectCampaigns	connectcampaigns	AWS_ENDPOINT_URL_CONNECTCAMPAIGNS
ConnectCases	connectcases	AWS_ENDPOINT_URL_CONNECTCASES
ConnectParticipant	connectparticipant	AWS_ENDPOINT_URL_CONNECTPARTICIPANT
ControlTower	controltower	AWS_ENDPOINT_URL_CONTROLTOWER

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

serviceId	Ki	Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE>
Device Farm	d:	AWS_ENDPOINT_URL_DEVICE_FARM
DevOps Guru	d:	AWS_ENDPOINT_URL_DEVOPS_GURU
Direct Connect	d:	AWS_ENDPOINT_URL_DIRECT_CONNECT
Application Discovery Service	a:	AWS_ENDPOINT_URL_APPLICATION_DISCOVERY_SERVICE
DLM	d:	AWS_ENDPOINT_URL_DLM
Database Migration Service	d:	AWS_ENDPOINT_URL_DATABASE_MIGRATION_SERVICE
DocDB	d:	AWS_ENDPOINT_URL_DOCDB
DocDB Elastic	d:	AWS_ENDPOINT_URL_DOCDB_ELASTIC
drs	d:	AWS_ENDPOINT_URL_DRS
Directory Service	d:	AWS_ENDPOINT_URL_DIRECTORY_SERVICE

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

serviceId	Ki Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE> pe la ur fil be Al co
Elastic Beanstalk	e: AWS_ENDPOINT_URL_ELASTIC_BEANSTALK e:
Elastic Transcoder	e: AWS_ENDPOINT_URL_ELASTIC_TRANSCODER r:
Elastic Load Balancing	e: AWS_ENDPOINT_URL_ELASTIC_LOAD_BALANCING o: c:
Elastic Load Balancing v2	e: AWS_ENDPOINT_URL_ELASTIC_LOAD_BALANCING_V2 o: c:
EMR	er AWS_ENDPOINT_URL_EMR
EMR containers	er AWS_ENDPOINT_URL_EMR_CONTAINERS i:
EMR Serverless	er AWS_ENDPOINT_URL_EMR_SERVERLESS r:
EntityResolution	er AWS_ENDPOINT_URL_ENTITYRESOLUTION o:
Elasticsearch Service	e: AWS_ENDPOINT_URL_ELASTICSEARCH_SERVICE a: i:

serviceId	Kategori	Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE>
EventBridge	event	AWS_ENDPOINT_URL_EVENTBRIDGE
Evidently	event	AWS_ENDPOINT_URL_EVIDENTLY
finspace	financial	AWS_ENDPOINT_URL_FINSPLACE
finspace data	financial	AWS_ENDPOINT_URL_FINSPLACE_DATA
Firehose	financial	AWS_ENDPOINT_URL_FIREHOSE
fis	financial	AWS_ENDPOINT_URL_FIS
FMS	financial	AWS_ENDPOINT_URL_FMS
forecast	financial	AWS_ENDPOINT_URL_FORECAST
forecastquery	financial	AWS_ENDPOINT_URL_FORECASTQUERY
FraudDetector	financial	AWS_ENDPOINT_URL_FRAUDETECTOR
FreeTier	financial	AWS_ENDPOINT_URL_FREETIER
FSx	financial	AWS_ENDPOINT_URL_FSX
GameLift	game	AWS_ENDPOINT_URL_GAMELIFT

serviceId	Kategori	Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE>
Glacier	g:	AWS_ENDPOINT_URL_GLACIER
Global Accelerator	g:	AWS_ENDPOINT_URL_GLOBAL_ACCELERATOR
Glue	g:	AWS_ENDPOINT_URL_GLUE
grafana	g:	AWS_ENDPOINT_URL_GRAFANA
Greengrass	g:	AWS_ENDPOINT_URL_GREENGRASS
GreengrassV2	g:	AWS_ENDPOINT_URL_GREENGRASSV2
GroundStation	g:	AWS_ENDPOINT_URL_GROUNDSTATION
GuardDuty	g:	AWS_ENDPOINT_URL_GUARDDUTY
Health	h:	AWS_ENDPOINT_URL_HEALTH
HealthLake	h:	AWS_ENDPOINT_URL_HEALTHLAKE
Honeycode	h:	AWS_ENDPOINT_URL_HONEYCODE

serviceId	Keterangan
	Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE>
IAM	<code>AWS_ENDPOINT_URL_IAM</code>
identitystore	<code>AWS_ENDPOINT_URL_IDENTITYSTORE</code>
imagebuilder	<code>AWS_ENDPOINT_URL_IMAGEBUILDER</code>
ImportExport	<code>AWS_ENDPOINT_URL_IMPORTEXPORT</code>
Inspector	<code>AWS_ENDPOINT_URL_INSPECTOR</code>
Inspector Scan	<code>AWS_ENDPOINT_URL_INSPECTOR_SCAN</code>
Inspector2	<code>AWS_ENDPOINT_URL_INSPECTOR2</code>
InternetMonitor	<code>AWS_ENDPOINT_URL_INTERNETMONITOR</code>
IoT	<code>AWS_ENDPOINT_URL_IOT</code>
IoT Data Plane	<code>AWS_ENDPOINT_URL_IOT_DATA_PLANE</code>

serviceId	Ki Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE> pe la ur fil be Al co
IoT Jobs Data Plane	ic AWS_ENDPOINT_URL_IOT_JOBS_DATA_PLANE di e
IoT 1Click Devices Service	ic AWS_ENDPOINT_URL_IOT_1CLICK_DEVICES_ k_ SERVICE _:
IoT 1Click Projects	ic AWS_ENDPOINT_URL_IOT_1CLICK_PROJECTS k_ s
IoTAnalytics	ic AWS_ENDPOINT_URL_IOTANALYTICS ic
IotDeviceAdvisor	ic AWS_ENDPOINT_URL_IOTDEVICEADVISOR ac
IoT Events	ic AWS_ENDPOINT_URL_IOT_EVENTS s
IoT Events Data	ic AWS_ENDPOINT_URL_IOT_EVENTS_DATA s_
IoTFleetHub	ic AWS_ENDPOINT_URL_IOTFLEETHUB ul
IoTFleetWise	ic AWS_ENDPOINT_URL_IOTFLEETWISE is

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

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

serviceId	Ki Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE> pe la ur fil be Al co
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
Lex Runtime Service	l: AWS_ENDPOINT_URL_LEX_RUNTIME_SERVICE m e
Lex Models V2	l: AWS_ENDPOINT_URL_LEX_MODELS_V2 s_
Lex Runtime V2	l: AWS_ENDPOINT_URL_LEX_RUNTIME_V2 m

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

serviceId	Ki Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE> pe la ur fil be Al co
m2	m: AWS_ENDPOINT_URL_M2
Machine Learning	m: AWS_ENDPOINT_URL_MACHINE_LEARNING e:
Macie2	m: AWS_ENDPOINT_URL_MACIE2
ManagedBlockchain	m: AWS_ENDPOINT_URL_MANAGEDBLOCKCHAIN o:
ManagedBlockchain Query	m: AWS_ENDPOINT_URL_MANAGEDBLOCKCHAIN_QUERY o: q:
Marketplace Agreement	m: AWS_ENDPOINT_URL_MARKETPLACE_AGREEMENT C: e:
Marketplace Catalog	m: AWS_ENDPOINT_URL_MARKETPLACE_CATALOG C: g
Marketplace Deployment	m: AWS_ENDPOINT_URL_MARKETPLACE_DEPLOYMENT C: m:

serviceId	Kategori	Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE>
Marketplace Entitlement Service	marketplace	AWS_ENDPOINT_URL_MARKETPLACE_ENTITLEMENT_SERVICE
Marketplace Commerce Analytics	marketplace	AWS_ENDPOINT_URL_MARKETPLACE_COMMERCE_ANALYTICS
MediaConnect	mediaconnect	AWS_ENDPOINT_URL_MEDIACONNECT
MediaConvert	mediaconvert	AWS_ENDPOINT_URL_MEDIACONVERT
MediaLive	medialive	AWS_ENDPOINT_URL_MEDIALIVE
MediaPackage	mediapackage	AWS_ENDPOINT_URL_MEDIAPACKAGE
MediaPackage Vod	mediapackagevod	AWS_ENDPOINT_URL_MEDIAPACKAGE_VOD
MediaPackageV2	mediapackagev2	AWS_ENDPOINT_URL_MEDIAPACKAGEV2

serviceId	Ki Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE> pe la ur fil be Al co
MediaStore	me AWS_ENDPOINT_URL_MEDIASTORE e
MediaStore Data	me AWS_ENDPOINT_URL_MEDIASTORE_DATA e_
MediaTailor	me AWS_ENDPOINT_URL_MEDIATAILOR O:
Medical Imaging	me AWS_ENDPOINT_URL_MEDICAL_IMAGING m:
MemoryDB	me AWS_ENDPOINT_URL_MEMORYDB
Marketplace Metering	m: AWS_ENDPOINT_URL_MARKETPLACE_METERING C n:
Migration Hub	m: AWS_ENDPOINT_URL_MIGRATION_HUB _f
mgn	m: AWS_ENDPOINT_URL_MGN
Migration Hub Refactor Spaces	m: AWS_ENDPOINT_URL_MIGRATION_HUB_REFACTOR_SPACES c e:

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

serviceId	Kategori	Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE>
NetworkManager	networkmanager	AWS_ENDPOINT_URL_NETWORKMANAGER
NetworkMonitor	networkmonitor	AWS_ENDPOINT_URL_NETWORKMONITOR
nimble	nimble	AWS_ENDPOINT_URL_NIMBLE
OAM	oam	AWS_ENDPOINT_URL_OAM
Omics	omics	AWS_ENDPOINT_URL_OMICS
OpenSearch	opensearch	AWS_ENDPOINT_URL_OPENSEARCH
OpenSearchServerless	opensearchserverless	AWS_ENDPOINT_URL_OPENSEARCHSERVERLESS
OpsWorks	opsworks	AWS_ENDPOINT_URL_OPSWORKS
OpsWorksCM	opsworkscm	AWS_ENDPOINT_URL_OPSWORKSCM
Organizations	organizations	AWS_ENDPOINT_URL_ORGANIZATIONS
OSIS	osis	AWS_ENDPOINT_URL_OSIS
Outposts	outposts	AWS_ENDPOINT_URL_OUTPOSTS

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

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

serviceId	Key	Value
	<code>AWS_ENDPOINT_URL_<SERVICE></code>	Variabel lingkungan <code>AWS_ENDPOINT_URL_<SERVICE></code>
QLDB Session	<code>aws_qlldb_session_endpoint_url</code>	<code>AWS_ENDPOINT_URL_QLDB_SESSION</code>
QuickSight	<code>aws_quicksight_endpoint_url</code>	<code>AWS_ENDPOINT_URL_QUICKSIGHT</code>
RAM	<code>aws_ram_endpoint_url</code>	<code>AWS_ENDPOINT_URL_RAM</code>
rbin	<code>aws_rbin_endpoint_url</code>	<code>AWS_ENDPOINT_URL_RBIN</code>
RDS	<code>aws_rds_endpoint_url</code>	<code>AWS_ENDPOINT_URL_RDS</code>
RDS Data	<code>aws_rds_data_endpoint_url</code>	<code>AWS_ENDPOINT_URL_RDS_DATA</code>
Redshift	<code>aws_redshift_endpoint_url</code>	<code>AWS_ENDPOINT_URL_REDSHIFT</code>
Redshift Data	<code>aws_redshift_data_endpoint_url</code>	<code>AWS_ENDPOINT_URL_REDSHIFT_DATA</code>
Redshift Serverless	<code>aws_redshift_serverless_endpoint_url</code>	<code>AWS_ENDPOINT_URL_REDSHIFT_SERVERLESS</code>
Rekognition	<code>aws_rekognition_endpoint_url</code>	<code>AWS_ENDPOINT_URL_REKOGNITION</code>
repostspace	<code>aws_repostspace_endpoint_url</code>	<code>AWS_ENDPOINT_URL_REPOSTSPACE</code>

serviceId	Ki Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE> pe la ur fil be A co
resiliencehub	r AWS_ENDPOINT_URL_RESILIENCEHUB el
Resource Explorer 2	r AWS_ENDPOINT_URL_RESOURCE_EXPLORER_2 e: 2
Resource Groups	r AWS_ENDPOINT_URL_RESOURCE_GROUPS g:
Resource Groups Tagging API	r AWS_ENDPOINT_URL_RESOURCE_GROUPS_TAG g: GING_API g:
RoboMaker	r AWS_ENDPOINT_URL_ROBOMAKER
RolesAnywhere	r AWS_ENDPOINT_URL_ROLESEANYWHERE h:
Route 53	r AWS_ENDPOINT_URL_ROUTE_53
Route53 Recovery Cluster	r AWS_ENDPOINT_URL_ROUTE53_RECOVERY_CLUSTER e: lu

serviceId	Kategori	Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE>
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	Service	AWS_ENDPOINT_URL_S3
S3 Control	Service	AWS_ENDPOINT_URL_S3_CONTROL
S3Outposts	Service	AWS_ENDPOINT_URL_S3OUTPOSTS
SageMaker	Service	AWS_ENDPOINT_URL_SAGEMAKER

serviceId	Ki Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE>
SageMaker A2I Runtime	s: AWS_ENDPOINT_URL_SAGEMAKER_A2I_RUNTIME _i ir
Sagemaker Edge	s: AWS_ENDPOINT_URL_SAGEMAKER_EDGE _e
SageMaker FeatureStore Runtime	s: AWS_ENDPOINT_URL_SAGEMAKER_FEATURESTORE_RUNTIME _f ir
SageMaker Geospatial	s: AWS_ENDPOINT_URL_SAGEMAKER_GEOSPATIAL _g a:
SageMaker Metrics	s: AWS_ENDPOINT_URL_SAGEMAKER_METRICS _m
SageMaker Runtime	s: AWS_ENDPOINT_URL_SAGEMAKER_RUNTIME _r
savingsplans	s: AWS_ENDPOINT_URL_SAVINGSPLANS a:
Scheduler	s: AWS_ENDPOINT_URL_SCHEDULER
schemas	s: AWS_ENDPOINT_URL_SCHEMAS

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

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

serviceId	Ki Variabel lingkungan AWS_ENDPOINT_URL_<SERVICE> pe la ur fil be Al co
Ssm Sap	s: AWS_ENDPOINT_URL_SSM_SAP
SSO	s: AWS_ENDPOINT_URL_SSO
SSO Admin	s: AWS_ENDPOINT_URL_SSO_ADMIN
SSO OIDC	s: AWS_ENDPOINT_URL_SSO_OIDC
SFN	s: AWS_ENDPOINT_URL_SFN
Storage Gateway	s: AWS_ENDPOINT_URL_STORAGE_GATEWAY at
STS	s: AWS_ENDPOINT_URL_STS
SupplyChain	s: AWS_ENDPOINT_URL_SUPPLYCHAIN in
Support	s: AWS_ENDPOINT_URL_SUPPORT
Support App	s: AWS_ENDPOINT_URL_SUPPORT_APP PI
SWF	s: AWS_ENDPOINT_URL_SWF
synthetics	s: AWS_ENDPOINT_URL_SYNTHETICS S
Textract	t: AWS_ENDPOINT_URL_TEXTRACT

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

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

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

Default konfigurasi cerdas

Note

Untuk bantuan dalam memahami tata letak halaman pengaturan, atau dalam menafsirkan tabel Support by AWS SDKs and tools berikut, lihat [Memahami halaman pengaturan panduan ini](#).

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

Konfigurasi fungsi ini dengan menggunakan yang berikut 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 setelah 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 `legacy`, 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 dengan perkembangan praktik terbaik.

Nilai default: `legacy`

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

Nilai yang valid:

- `legacy`— Menyediakan pengaturan default yang bervariasi menurut SDK dan ada sebelum 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](#), 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 pemilih `defaults_mode`:

- `retryMode`— Menentukan bagaimana SDK mencoba ulang. Lihat [Coba lagi perilaku](#).
- `stsRegionalEndpoints`— Menentukan bagaimana SDK menentukan Layanan AWS titik akhir yang digunakannya untuk berbicara dengan (). AWS Security Token Service AWS STS Lihat [AWS STS Titik akhir regional](#).
- `s3UsEast1RegionalEndpoints`— Menentukan 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).

Support oleh AWS SDKs dan alat

Berikut ini SDKs mendukung fitur dan pengaturan yang dijelaskan dalam topik ini. Setiap pengecualian sebagian dicatat. Setiap pengaturan properti sistem JVM didukung oleh AWS SDK untuk Java dan satu-satunya. AWS SDK untuk 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 4.x	Ya	Parameter tidak dioptimalkan: <code>connectTimeoutInMillis</code> , <code>tlsNegotiationTimeoutInMillis</code> .
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	Didukung	Catatan atau informasi lebih lanjut
SDK untuk Python (Boto3)	Ya	Parameter tidak dioptimalkan: <code>tlsNegotiationTimeoutInMillis</code> .
SDK for Ruby 3.x	Ya	
SDK untuk Rust	Tidak	
SDK para Swift	Tidak	
Alat untuk PowerShell V5	Ya	Parameter tidak dioptimalkan: <code>connectTimeoutInMilliseconds</code> , <code>tlsNegotiationTimeoutInMilliseconds</code> .
Alat untuk PowerShell V4	Ya	Parameter tidak dioptimalkan: <code>connectTimeoutInMilliseconds</code> , <code>tlsNegotiationTimeoutInMilliseconds</code> .

AWS Pustaka Runtime Umum (CRT)

Pustaka AWS Common Runtime (CRT) adalah pustaka dasar dari SDKs 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 dibagi di semua SDKs menyediakan penggunaan kembali kode yang lebih baik, optimasi, dan akurasi. 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 (, SHA256 HMAC) MD5 SHA256, penandatanganan, AES
- [awslabs/aws-c-common](#): Struktur data dasar, tipe threading/synchronization primitif, 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 pustaka C99 untuk berkomunikasi dengan layanan Amazon S3, yang dirancang untuk memaksimalkan throughput pada instans Amazon dengan bandwidth tinggi EC2
- [awslabs/aws-c-sdkutils](#): Pustaka utilitas untuk mengurai dan mengelola profil AWS
- [awslabs/aws-checksums](#): Perangkat keras lintas platform dipercepat CRC32c dan CRC32 dengan mundur ke implementasi perangkat lunak yang efisien
- [awslabs/aws-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 SDKs kecuali Go dan Rust.

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.

Diagram ketergantungan CRT menunjukkan bagaimana pustaka CRT individu saling terkait satu sama lain.

AWS SDKs dan kebijakan pemeliharaan alat

Ikhtisar

Dokumen ini menguraikan kebijakan pemeliharaan untuk AWS Software Development Kits (SDKs) dan Tools, termasuk Mobile dan IoT, dan dependensi yang SDKs mendasarinya. AWS secara teratur menyediakan AWS SDKs dan Tools dengan pembaruan yang mungkin berisi dukungan untuk fitur baru atau yang diperbarui AWS APIs, baru, penyempurnaan, perbaikan bug, patch 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 mayor SDKs dan Tools terdiri dari 5 fase, yang diuraikan di bawah ini.

- Pratinjau Pengembang (Tahap 0) - Selama fase ini, tidak didukung, tidak boleh digunakan dalam lingkungan produksi, dan dimaksudkan untuk akses awal dan tujuan umpan balik saja. SDKs 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, SDKs 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.

Garis waktu kebijakan pemeliharaan

Siklus hidup ketergantungan

Sebagian besar AWS SDKs 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 ketergantungan tersebut. 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 SDKs, menguraikan jalur ke end-of-support dan menautkan ke dokumentasi SDK.

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

AWS SDKs dan siklus hidup versi Tools

Tabel 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 SDKs dan Alat serta dependensi yang mendasarinya, lihat. [Kebijakan pemeliharaan](#)

SDK	Versi utama	Fase Saat Ini	Tanggal Ketersediaan Umum	Catatan
AWS CLI	1.x	Pengumuman Pemeliharaan	9/2/2013	Lihat pengumuman untuk detail dan tanggal
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	Akhir dukungan	11/19/2015	
SDK for Java	1.x	Akhir dukungan	25/03/2010	
SDK for Java	2.x	Ketersediaan Umum	11/20/2018	
SDK untuk JavaScript	1.x	Akhir dukungan	5/6/2013	
SDK untuk JavaScript	2.x	Akhir dukungan	6/19/2014	

SDK	Versi utama	Fase Saat Ini	Tanggal Ketersediaan Umum	Catatan
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 .NET	4.x	Ketersediaan Umum	4/28/2025	
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 utama	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	Ketersediaan Umum	9/17/2024	
Alat untuk PowerShell	2.x	Akhir dukungan	11/8/2013	
Alat untuk PowerShell	3.x	Akhir dukungan	29/7/2015	
Alat untuk PowerShell	4.x	Ketersediaan Umum	11/21/2019	
Alat untuk PowerShell	5.x	Ketersediaan Umum	6/23/2025	

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

Riwayat dokumen untuk AWS SDKs dan Panduan Referensi Alat

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

Perubahan	Deskripsi	Tanggal
Menambahkan pengaturan S3 Express One Zone baru	Menambahkan pengaturan S3 Express One Zone baru untuk menonaktifkan otentikasi sesi.	Oktober 13, 2025
Menambahkan pohon keputusan otentikasi baru	Menambahkan pohon keputusan baru untuk membantu dalam keputusan otentikasi antar opsi.	September 23, 2025
Menambahkan fitur skema otentikasi baru	Menambahkan fitur skema otentikasi baru. Pembaruan untuk titik akhir AWS STS Regional.	Agustus 18, 2025
Menambahkan versi baru Alat untuk PowerShell	Menambahkan versi terbaru Alat untuk PowerShell dukungan ke semua referensi Pengaturan Kompatibilitas dengan AWS SDKs tabel. Menambahkan fitur injeksi awalan Host.	Juni 23, 2025
Pembaruan judul halaman	Lebih banyak judul, judul tabel, abstrak, dan pembaruan SEO.	Maret 5, 2025
Pembaruan judul halaman	Memperbarui konten untuk menggunakan judul yang lebih deskriptif.	Februari 24, 2025

Menambahkan Swift SDK ke referensi Pengaturan	Menambahkan dukungan Swift SDK ke semua referensi Pengaturan Kompatibilitas dengan AWS SDKs tabel.	September 17, 2024
SDK for Java 1.x properti sistem	Tambahkan detail tentang pengaturan konfigurasi sistem JVM yang didukung oleh 1.x. AWS SDK untuk 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 IMDSv1 fallback.	Desember 29, 2023
Minta kompresi	Menambahkan pengaturan untuk fitur kompresi permintaan.	27 Desember 2023
Tabel kompatibilitas	Tabel kompatibilitas untuk SDK dan fitur alat diperbarui untuk menyertakan SDK untuk Kotlin, SDK untuk Rust, dan. Alat AWS untuk PowerShell	Desember 10, 2023
Pembaruan otentikasi	Pembaruan untuk metode otentikasi yang didukung untuk SDKs dan alat.	Juli 1, 2023

Pembaruan praktik terbaik IAM	Memperbarui panduan untuk menyelaraskan dengan praktik terbaik IAM. Untuk informasi lebih lanjut, 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 kredensi 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

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