Panduan Referensi

AWS SDKs dan Tools

AWS SDKs dan Tools: Panduan Referensi

Table of Contents

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 alatmencakup 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:
• SDK atau alat apa pun	AWS SDKs dan kebijakan pemeliharaan alat
• 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:
• 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	
• 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 for Rust • AWS SDK for Swift • AWS Tools for Windows PowerShell	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 kredensional 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

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

`credentials`File 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/alat saat profil itu digunakan. File-file ini “dibagikan” karena nilai mempengaruhi aplikasi, proses, atau SDKs lingkungan lokal untuk pengguna.

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

Profil

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

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

> **Note**
>
> `[default]`hanyalah profil yang tidak disebutkan namanya. Profil ini dinamai `default` karena merupakan profil default yang digunakan oleh SDK jika pengguna tidak menentukan profil. Itu tidak memberikan nilai default yang diwariskan ke profil lain. Jika Anda menyetel sesuatu di `[default]` profil dan Anda tidak mengaturnya di profil bernama, maka nilainya tidak disetel saat Anda menggunakan profil bernama.

Menetapkan profil bernama

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

Konfigurasikan 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

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

`config`File 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 pengenal `profile` 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 kredensil 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 pengenal. Layanan AWS Pengenal didasarkan pada model API `serviceId` dengan mengganti semua spasi dengan garis bawah dan huruf kecil semua huruf. Untuk daftar semua kunci pengenal layanan yang akan digunakan di `services` bagian ini, lihatPengidentifikasi untuk titik akhir khusus layanan. Kunci pengenal layanan diikuti oleh pengaturan bersarang dengan masing-masing pada barisnya sendiri dan menjorok oleh dua spasi.

Contoh berikut menggunakan `services` definisi untuk mengonfigurasi titik akhir yang akan digunakan untuk permintaan yang dibuat hanya untuk Amazon DynamoDB layanan. `"local-dynamodb"servicesBagian` ini direferensikan di `profile` bagian dengan nama menggunakan `services` kunci. Kunci Layanan AWS pengenal adalah`dynamodb`. Subbagian Amazon DynamoDB layanan dimulai pada telepon`dynamodb =` . Baris yang segera mengikuti yang menjorok termasuk dalam ayat itu dan berlaku untuk layanan itu.

```
[profile dev]
```

```
services = local-dynamodb

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

Untuk informasi selengkapnya tentang konfigurasi titik akhir kustom, lihatTitik akhir khusus layanan.

Format file kredensial

Aturan untuk `credentials` file umumnya identik dengan aturan untuk `config` file, kecuali bahwa bagian profil tidak dimulai dengan kata`profile`. Gunakan hanya nama profil itu sendiri di antara tanda kurung siku. Contoh berikut menunjukkan `credentials` file dengan bagian profil bernama bernama`foo`.

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

Hanya pengaturan berikut yang dianggap “rahasia” atau sensitif yang dapat disimpan dalam `credentials` file:`aws_access_key_id`,`aws_secret_access_key`, dan`aws_session_token`. Meskipun pengaturan ini dapat ditempatkan di `config` file bersama, kami sarankan Anda menyimpan nilai-nilai sensitif ini dalam `credentials` file terpisah. Dengan cara ini, Anda dapat memberikan izin terpisah untuk setiap file, jika perlu.

Contoh berikut menunjukkan `credentials` file dasar dengan `[default]` profil. Ini mengatur `aws_access_key_id`,`aws_secret_access_key`, dan pengaturan `aws_session_token` global.

```
[default]
aws_access_key_id=AKIAIOSFODNN7EXAMPLE
aws_secret_access_key=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
aws_session_token=IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZVERYLONGSTR1
```

Terlepas dari apakah Anda menggunakan profil bernama atau "`default`" dalam `credentials` file Anda, pengaturan apa pun di sini akan digabungkan dengan pengaturan apa pun dari `config` file Anda yang menggunakan nama profil yang sama. Jika ada kredensil di kedua file untuk profil yang berbagi nama yang sama, kunci dalam file kredensil diutamakan.

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	`~/.aws/config` `~/.aws/credentials`
Windows	`%USERPROFILE%\.aws\config` `%USERPROFILE%\.aws\credentials`

Resolusi direktori rumah

~hanya digunakan untuk resolusi direktori home ketika:

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

Saat menentukan direktori home, variabel-variabel berikut diperiksa:

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

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

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

Ubah lokasi default file-file ini

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

Menggunakan variabel lingkungan

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

- `config`variabel lingkungan file: **`AWS_CONFIG_FILE`**
- `credentials`variabel lingkungan file: **`AWS_SHARED_CREDENTIALS_FILE`**

Linux/macOS

Anda dapat menentukan lokasi alternatif dengan menjalankan perintah ekspor berikut di Linux atau macOS.

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

Windows

Anda dapat menentukan lokasi alternatif dengan menjalankan perintah setx berikut di Windows.

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

Untuk informasi selengkapnya tentang mengonfigurasi sistem Anda menggunakan variabel lingkungan, lihatMenggunakan 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:

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

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

Menggunakan variabel lingkungan untuk mengkonfigurasi AWS SDKs dan alat secara global

Variabel lingkungan menyediakan cara lain untuk menentukan opsi konfigurasi dan kredensil 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, lihatDaftar 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...zrkuWJOgQs8IZZaIv2BXIa2R4Olgk
$ 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...zrkuWJOgQs8IZZaIv2BXIa2R4Olgk
C:\> setx AWS_REGION us-west-2
```

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

PowerShell

```
PS C:\> $Env:AWS_ACCESS_KEY_ID="AKIAIOSFODNN7EXAMPLE"
PS C:\> $Env:AWS_SECRET_ACCESS_KEY="wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY"
PS C:
\> $Env:AWS_SESSION_TOKEN="AQoEXAMPLEH4aoAH0gNCAPy...truncated...zrkuWJOgQs8IZZaIv2BXIa2R4Ol
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 JavaAWS SDK untuk Kotlin

Properti sistem JVM menyediakan cara lain untuk menentukan opsi konfigurasi dan kredensil 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 mengautentikasi. 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 otentikasi untuk kode yang berjalan secara lokal (tidak masuk) 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
- Menggunakan Peran IAM Di Mana Saja untuk mengautentikasi dan AWS SDKs alat— Anda dapat menggunakan IAM Roles Anywhere untuk mendapatkan kredensil 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.

Opsi otentikasi untuk kode yang berjalan dalam lingkungan AWS

Jika kode Anda berjalan AWS, kredensi dapat dibuat secara otomatis tersedia untuk aplikasi Anda. Misalnya, jika aplikasi Anda di-host di Amazon Elastic Compute Cloud, dan 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 CloudShelluntuk 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 kredensil AWS keamanan sementara secara dinamis dengan menggunakan federasi identitas web.

Dengan federasi identitas web, Anda tidak perlu membuat kode masuk khusus atau mengelola identitas pengguna Anda sendiri. Sebagai gantinya, pengguna aplikasi dapat masuk menggunakan penyedia identitas eksternal (IDP) yang terkenal, seperti Login with Amazon, Facebook, Google, atau iDP yang kompatibel dengan OpenID Connect (OIDC) lainnya. Mereka dapat menerima token otentikasi, dan kemudian menukar token itu dengan kredensil 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, lihatDengan 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.

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 Webmemiliki dasar-dasar dasar sebagai berikut:

- Memahami dan mendapatkan AWS kredensil Anda — Akses opsi kunci dan praktik manajemen untuk akses konsol dan program.

Plugin IAM Identity Center tepercaya 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 tepercaya, 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 IAM Identity Center untuk mengautentikasi AWS SDK dan alat

AWS IAM Identity Center adalah metode yang direkomendasikan untuk memberikan AWS kredensil saat mengembangkan AWS aplikasi pada layanan AWS non-komputasi. Misalnya, ini akan menjadi sesuatu seperti lingkungan pengembangan lokal Anda. Jika Anda mengembangkan AWS sumber daya, seperti Amazon Elastic Compute Cloud (Amazon EC2) atau AWS Cloud9, kami sarankan untuk mendapatkan kredensil dari layanan tersebut.

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

- Portal AWS akses adalah lokasi web tempat Anda masuk secara manual ke Pusat Identitas IAM. Format URL adalah `d-xxxxxxxxxx.awsapps.com/start` atau*`your_subdomain`*`.awsapps.com/start`. Saat masuk ke portal AWS akses, Anda dapat melihat Akun AWS dan peran yang telah dikonfigurasi untuk pengguna tersebut. Prosedur ini menggunakan portal AWS akses untuk mendapatkan nilai konfigurasi yang Anda butuhkan untuk proses otentikasi SDK/alat.
- AWS CLI Ini digunakan untuk mengonfigurasi SDK atau alat Anda untuk menggunakan autentikasi IAM Identity Center untuk panggilan API yang dibuat oleh kode Anda. Proses satu kali ini memperbarui AWS `config` file bersama Anda, yang kemudian digunakan oleh SDK atau alat saat Anda menjalankan kode.

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.

Konfigurasikan akses terprogram menggunakan IAM Identity Center

Langkah 1: Buat akses dan pilih set izin yang sesuai

Pilih salah satu metode berikut untuk mengakses AWS kredensional 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 Konfigurasikan 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 diatur 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 kredensional 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, lihatProfil.

4. (Opsional) Pada prompt AWS CLI perintah, konfirmasikan identitas sesi aktif dengan menjalankan `aws sts get-caller-identity` perintah. Respons harus menunjukkan set izin IAM Identity Center yang Anda konfigurasikan.
5. Jika Anda menggunakan AWS SDK, buat aplikasi untuk SDK Anda di lingkungan pengembangan Anda.

    a. Untuk beberapa SDKs, paket tambahan seperti `SSO` dan `SSOOIDC` 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 punAWS kunci akses. Anda harus menghapus kredensi statis apa pun sebelum SDK atau alat akan menggunakan kredensil Pusat Identitas IAM karena diutamakan. Memahami rantai penyedia kredensi

Untuk mempelajari lebih dalam bagaimana alat SDKs dan menggunakan dan menyegarkan kredensional menggunakan konfigurasi ini, lihat. Bagaimana otentikasi IAM Identity Center diselesaikan untuk AWS SDKs dan alat

Untuk mengonfigurasi pengaturan 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 IAM Identity Center — Konfigurasikan 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 kompatibi litas 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`https://xxx.awsapps.com/start` . Anda masuk ke portal ini menggunak an kredenal 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 kredensional	AWS kredensialnya kredensi sigv4	Kredensi yang sebenarny a digunakan SDK untuk sebagian besar Layanan AWS panggilan (khususny a, 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 mendapatk an kredensialnya, seperti kelas atau modul yang menyediakan fungsionalitas.

Memahami resolusi kredensi SDK untuk Layanan AWS

API Pusat Identitas IAM menukar kredenal token pembawa untuk kredenal sigv4. Sebagian besar Layanan AWS adalah sigv4 APIs, dengan beberapa pengecualian seperti dan. Amazon CodeWhisperer Amazon CodeCatalyst Berikut ini menjelaskan proses resolusi kredenal 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 kredensional 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 kredensil untuk aplikasi Anda.

- Proses resolusi kredenal selesai selama runtime ketika klien dibuat.

Untuk mengambil kredensil 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 kredenal 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 getRoleCredentialsIdentitas 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 kredenal 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 kredensi 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 kredenal set izin tidak di-cache. Kode Anda tidak akan dapat membuat klien baru dan menyelesaikan proses resolusi kredensi sampai Anda memiliki token akses yang valid.

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

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

Anda dapat menggunakan IAM Roles Anywhere untuk mendapatkan kredenal 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 kredenal keamanan sementara dari IAM Roles Anywhere, gunakan alat bantu kredenal yang disediakan oleh IAM Roles Anywhere. Alat kredensi mengimplementasikan proses penandatanganan untuk IAM Roles Anywhere.

Untuk petunjuk mengunduh alat bantu kredensi, lihat Memperoleh kredenal keamanan sementara dari AWS Identity and Access Management Peran Di Mana Saja di Panduan Pengguna IAM Roles Anywhere.

Untuk menggunakan kredenal 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 kredensi 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. AWS Management Console Contoh berikut menunjukkan `config` file bersama yang mengonfigurasi pengambilan kredensional 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 kredenal 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 AndaWilayah 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 kredensional dari atau. `credential_source` `source_profile`

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

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

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

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

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

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

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

Untuk detail tentang semua pengaturan penyedia kredensi peran asumsi, lihat Asumsikan penyedia kredensi peran di panduan ini.

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

Dengan asumsi peran melibatkan penggunaan seperangkat kredensil 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 AndaWilayah 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 klaster 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: Konfigurasikan SDK atau alat

Konfigurasikan 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 kredensil 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 kredenal sementara alat secara langsung, lihat. Menggunakan kredensi jangka pendek untuk AWS SDKs mengautentikasi dan alat

Gunakan kredensi 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 Anda 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 kredensional, 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 kredensi jangka pendek yang diambil dari portal akses AWS

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

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

3. Simpan file tersebut. File sekarang `~/.aws/credentials` harus ada di sistem pengembangan lokal Anda. File ini berisi profil [default] yang digunakan SDK atau alat jika profil bernama tertentu tidak ditentukan.
4. Masuk ke portal AWS akses.
5. Ikuti petunjuk ini untuk penyegaran kredenal manual untuk menyalin kredensi peran IAM dari portal akses. AWS

    a. Untuk langkah 4 dalam petunjuk terkait, pilih nama peran IAM yang memberikan akses untuk kebutuhan pengembangan Anda. Peran ini biasanya memiliki nama seperti PowerUserAccessatau Pengembang.

    b. Untuk langkah 7 dalam instruksi yang ditautkan, pilih opsi Tambahkan profil ke file AWS kredensial Anda secara manual dan salin isinya.

6. Tempelkan kredenal yang disalin ke file lokal Anda. `credentials` Nama profil yang dihasilkan tidak diperlukan jika Anda menggunakan `default` profil. File Anda harus menyerupai yang berikut ini.

```
[default]
aws_access_key_id=AKIAIOSFODNN7EXAMPLE
aws_secret_access_key=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
aws_session_token=IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZVERYLONG
```

7. Simpan file `credentials`.

Saat SDK membuat klien layanan, SDK akan mengakses kredensi sementara ini dan menggunakannya untuk setiap permintaan. Pengaturan untuk peran IAM yang dipilih pada langkah 5a menentukan berapa lama kredensi sementara valid. Durasi maksimum adalah dua belas jam.

Setelah kredensi sementara kedaluwarsa, ulangi langkah 4 hingga 7.

Menggunakan kredensi jangka panjang untuk AWS SDKs mengautentikasi dan alat

> **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.

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 kredensional

Peringatan untuk kredensional

- 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 kredenal apa pun yang disimpan dalam AWS `credentials` file bersama disimpan dalam teks biasa.

Panduan tambahan untuk mengelola kredensional dengan aman

Untuk diskusi umum tentang cara mengelola AWS kredensional 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 EC2 instans Amazon.

Prasyarat: Buat akun AWS

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

1. Buat akun.

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

2. Buat pengguna administratif.

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

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

Tak satu pun dari akun ini sesuai untuk melakukan pengembangan pada AWS atau untuk menjalankan aplikasi di AWS. Sebagai praktik terbaik, Anda perlu membuat pengguna, set izin, atau

peran layanan yang sesuai untuk tugas-tugas ini. Untuk informasi selengkapnya, lihat Menerapkan izin hak istimewa terkecil di Panduan Pengguna IAM.

Langkah 1: Buat pengguna IAM Anda

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

Langkah 2: Dapatkan kunci akses Anda

1. Di panel navigasi konsol IAM, pilih Pengguna dan kemudian pilih pengguna **`User name`** yang Anda buat sebelumnya.
2. Pada halaman pengguna, pilih halaman Security credentials. Kemudian, di bawah tombol Access, pilih Create Access Key.
3. Untuk Buat kunci akses Langkah 1, pilih Command Line Interface (CLI) atau kode Lokal. Kedua opsi menghasilkan jenis kunci yang sama untuk digunakan dengan kedua AWS CLI dan 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 kredensitas 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 kredensional. Ini juga dapat ditetapkan sebagai variabel lingkungan, lihat AWS kunci akses untuk nama variabel lingkungan. Ini adalah cara untuk membantu Anda memulai, tetapi kami sarankan Anda beralih ke IAM Identity Center atau kredensi sementara lainnya sesegera mungkin. Setelah Anda beralih dari penggunaan kredensi jangka panjang, ingatlah untuk menghapus kredensi ini dari file bersama. `credentials`

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

    a. Masuk ke AWS Management Console dan buka EC2 konsol Amazon di https://console.aws.amazon.com/ec2/.

    b. Pilih instans.

    c. Pilih Tindakan, Keamanan, 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 kredensil yang ditingkatkan identitas.
- **`applicationArn`**: String pengenal unik untuk klien atau aplikasi. Nilai ini adalah aplikasi ARN yang memiliki OAuth hibah yang dikonfigurasi.
- **`applicationRoleArn`**: (Opsional) Peran IAM ARN yang akan diasumsikan `AssumeRoleWithWebIdentity` sehingga OIDC AWS STS dan klien dapat di-bootstrap tanpa penyedia kredensional default. Jika ini tidak disediakan, nilai `accessRoleArn` parameter akan digunakan.
- **`ssoOidcClient`**: (Opsional) Klien SSO OIDC, seperti `SsoOidcClient`untuk Java atau Javascript, dengan `client-sso-oidc`konfigurasi yang ditentukan pelanggan. Jika tidak disediakan, klien OIDC yang menggunakan konfigurasi default 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 konfigurasi default dipakai dan digunakan.

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>1.0.0</version>
</dependency>
```

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

Kode contoh berikut menunjukkan cara membuat instance plugin propagasi identitas tepercaya dan kemudian menambahkan plugin ke instance klien layanan.

Contoh ini menggunakan `S3Client` sebagai Layanan AWS klien yang dipilih untuk menunjukkan memperoleh token IAM Identity Center. Namun, Layanan AWS yang lain yang mendukung TIP akan serupa.

```
StsClient client = StsClient.builder()
    .region(Region.US_EAST_1)
    .credentialsProvider(AnonymousCredentialsProvider.create()).build();

TrustedIdentityPropagationPlugin trustedIdentityPropagationPlugin =
 TrustedIdentityPropagationPlugin.builder()
    .stsClient(client)
    .webTokenProvider(() -> idToken)
    .applicationArn(idcApplicationArn)
    .accessRoleArn(accessRoleArn)
    .ssoOidcClient(SsoOidcClient.builder().region(Region.US_EAST_1).build())
    .build();

S3Client s3Client =

 S3Client.builder().region(Region.US_EAST_1).addPlugin(trustedIdentityPropagationPlugin)
```

```
        .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": "^1.0.0"
  },
```

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

Kode contoh berikut menunjukkan cara membuat instance plugin propagasi identitas tepercaya dan kemudian menambahkan plugin ke instance klien layanan.

Contoh ini menggunakan `S3Client` sebagai Layanan AWS klien yang dipilih untuk menunjukkan memperoleh token IAM Identity Center. Namun, Layanan AWS yang lain yang mendukung TIP akan serupa.

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

// Plugin configurations, please refer to the documentation on each of these fields.
const applicationRoleArn = 'YOUR_APPLICATION_ROLE_ARN';
const accessRoleArn = 'YOUR_ACCESS_ROLE_ARN';
const applicationArn = 'YOUR_APPLICATION_ARN';

const s3Client = new S3Client({
 region,
 extensions: [
  TrustedIdentityPropagationExtension.create({
   webTokenProvider: async () => {
    return 'ID_TOKEN_FROM_YOUR_IDENTITY_PROVIDER';
```

```
    },
    applicationRoleArn,
    accessRoleArn,
    applicationArn,
   }),
 ],
});
```

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 kredensional untuk digunakan untuk permintaan Anda, untuk mempertahankan pengaturan untuk digunakan dengan setiap layanan, dan untuk mendapatkan nilai yang akan digunakan untuk pengaturan global.

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

- AWS SDKs dan Penyedia kredensi standar Alat— Penyedia kredensi umum distandarisasi di beberapa. SDKs
- AWS SDKs dan fitur standar Alat— Fitur umum distandarisasi di beberapa. SDKs

Membuat klien layanan

Untuk mengakses secara terprogram Layanan AWS, SDKs gunakan 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. `config`File 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 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) setelan JVM yang dapat digunakan di luar kode Anda untuk mengonfigurasi fitur. Setiap topik tertaut di setiap daftar membawa Anda ke halaman pengaturan yang sesuai.

- `Config`daftar pengaturan file
- `Credentials`daftar pengaturan file
- Daftar variabel lingkungan
- 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.

`Config`daftar 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	
`account_id_endpoint_mode`	Titik akhir berbasis akun	
`api_versions`	Pengaturan konfigurasi umum	
`auth_scheme_preference`	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_min_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_multiregion_access_points`	Titik Akses Multi-Wilayah Amazon S3
`s3_use_arn_region`	Titik akses Amazon S3
`sdk_ua_app_id`	ID Aplikasi
`sigv4a_signing_region_set`	Skema otentikasi
`source_profile`	Asumsikan penyedia kredensi peran

Nama pengaturan	Detail
`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

`Credentials`daftar 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

Nama pengaturan	Detail	
`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	
`AWS_CONFIG_FILE`	Menemukan dan mengubah lokasi berbagi `config` dan `credentials` file AWS SDKs dan alat	
`AWS_CONTAINER_AUTH`	Penyedia kredensi kontainer	

Nama pengaturan	Detail	
ORIZATION _TOKEN		
AWS_CONTA INER_AUTH ORIZATION _TOKEN_FILE	Penyedia kredensi kontainer	
AWS_CONTA INER_CRED ENTIALS_F ULL_URI	Penyedia kredensi kontainer	
AWS_CONTA INER_CRED ENTIALS_R ELATIVE_URI	Penyedia kredensi kontainer	
AWS_DEFAU LTS_MODE	Default konfigurasi cerdas	
AWS_DISAB LE_HOST_P REFIX_INJ ECTION	Injeksi awalan host	
AWS_DISAB LE_REQUES T_COMPRESSION	Minta kompresi	
AWS_EC2_M ETADATA_D ISABLED	Penyedia kredensi IMDS	

Nama pengaturan	Detail
AWS_EC2_METADATA_SERVICE_ENDPOINT	Penyedia kredensi IMDS
AWS_EC2_METADATA_SERVICE_ENDPOINT_MODE	Penyedia kredensi IMDS
AWS_EC2_METADATA_V1_DISABLED	Penyedia kredensi IMDS
AWS_ENABLE_ENDPOINT_DISCOVERY	Penemuan titik akhir
AWS_ENDPOINT_URL	Titik akhir khusus layanan
AWS_ENDPOINT_URL_<SERVICE>	Titik akhir khusus layanan
AWS_IGNORE_CONFIGURED_ENDPOINT_URLS	Titik akhir khusus layanan
AWS_MAX_ATTEMPTS	Coba lagi perilaku

Nama pengaturan	Detail
AWS_METADATA_SERVICE_NUM_ATTEMPTS	EC2 Metadata contoh Amazon
AWS_METADATA_SERVICE_TIMEOUT	EC2 Metadata contoh Amazon
AWS_PROFILE	Menggunakan `credentials` file bersama `config` dan untuk mengkonfigurasi AWS SDKs dan alat secara global
AWS_REGION	Wilayah AWS
AWS_REQUEST_CHECKSUM_CALCULATION	Perlindungan Integritas Data untuk Amazon S3
AWS_REQUEST_MIN_COMPRESSION_SIZE_BYTES	Minta kompresi
AWS_RESPONSE_CHECKSUM_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_DISABLE_MULTIREGION_ACCESS_POINTS`	Titik Akses Multi-Wilayah Amazon S3
`AWS_S3_USE_ARN_REGION`	Titik akses Amazon S3
`AWS_SDK_UA_APP_ID`	ID Aplikasi
`AWS_SECRET_ACCESS_KEY`	AWS kunci akses
`AWS_SESSION_TOKEN`	AWS kunci akses
`AWS_SHARED_CREDENTIALS_FILE`	Menemukan dan mengubah lokasi berbagi `config` dan `credentials` file AWS SDKs dan alat
`AWS_SIGV4A_SIGNING_REGION_SET`	Skema otentikasi
`AWS_STS_REGIONAL_ENDPOINTS`	AWS STS Titik akhir regional
`AWS_USE_DUALSTACK_ENDPOINT`	Dual-stack dan titik akhir FIPS
`AWS_USE_FIPS_ENDPOINT`	Dual-stack dan titik akhir FIPS

Nama pengaturan	Detail
`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
`aws.accessKeyId`	AWS kunci akses
`aws.accountId`	Titik akhir berbasis akun
`aws.accountIdEndpointMode`	Titik akhir berbasis akun
`aws.authSchemePreference`	Skema otentikasi
`aws.configFile`	Menemukan dan mengubah lokasi berbagi `config` dan `credentials` file AWS SDKs dan alat
`aws.defaultsMode`	Default konfigurasi cerdas
`aws.disableEc2MetadataV1`	Penyedia kredensi IMDS

Nama pengaturan	Detail	
`aws.disableHostPrefixInjection`	Injeksi awalan host	
`aws.disableRequestCompression`	Minta kompresi	
`aws.ec2MetadataServiceEndpoint`	Penyedia kredensi IMDS	
`aws.ec2MetadataServiceEndpointMode`	Penyedia kredensi IMDS	
`aws.endpointDiscoveryEnabled`	Penemuan titik akhir	
`aws.endpointUrl`	Titik akhir khusus layanan	
`aws.endpointUrl<ServiceName>`	Titik akhir khusus layanan	
`aws.ignoreConfiguredEndpointUrls`	Titik akhir khusus layanan	
`aws.maxAttempts`	Coba lagi perilaku	
`aws.profile`	Menggunakan `credentials` file bersama `config` dan untuk mengkonfigurasi AWS SDKs dan alat secara global	

Nama pengaturan	Detail	
`aws.region`	Wilayah AWS	
`aws.requestChecksumCalculation`	Perlindungan Integritas Data untuk Amazon S3	
`aws.requestMinCompressionSizeBytes`	Minta kompresi	
`aws.responseChecksumValidation`	Perlindungan Integritas Data untuk Amazon S3	
`aws.retryMode`	Coba lagi perilaku	
`aws.roleArn`	Asumsikan penyedia kredensi peran	
`aws.roleSessionName`	Asumsikan penyedia kredensi peran	
`aws.s3DisableMultiRegionAccessPoints`	Titik Akses Multi-Wilayah Amazon S3	
`aws.s3UseArnRegion`	Titik akses Amazon S3	
`aws.secretAccessKey`	AWS kunci akses	
`aws.sessionToken`	AWS kunci akses	

Nama pengaturan	Detail
`aws.sharedCredentialsFile`	Menemukan dan mengubah lokasi berbagi `config` dan `credentials` file AWS SDKs dan alat
`aws.useDualstackEndpoint`	Dual-stack dan titik akhir FIPS
`aws.useFipsEndpoint`	Dual-stack dan titik akhir FIPS
`aws.webIdentityTokenFile`	Asumsikan penyedia kredensi peran
`sdk.ua.appId`	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
- 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 kredensil yang valid untuk digunakan untuk membuat permintaan ke file. Layanan AWS Setelah kredensi 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 kredensil secara otomatis ketika mereka kedaluwarsa. Rantai penyedia kredensi bawaan memberi aplikasi Anda kemampuan untuk menyegarkan kredensil 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 (sepertiAWS_ACCESS_KEY_ID , danAWS_SECRET_ACCESS_KEY).
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 Pusat Identitas IAM	Dapatkan kredensil dari. AWS IAM Identity Center

Penyedia kredensi	Deskripsi
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 kredensil Anda dari sumber atau proses eksternal, termasuk Peran IAM Di Mana Saja.
Penyedia kredensi IMDS	Kredensi profil instans Amazon Elastic Compute Cloud (Amazon EC2). Kaitkan peran IAM dengan setiap EC2 instans Anda. Kredensi sementara untuk peran itu tersedia untuk kode yang berjalan dalam instance. Kredensialnya dikirimkan melalui layanan EC2 metadata Amazon.

Untuk setiap langkah dalam rantai, ada beberapa cara untuk menetapkan nilai pengaturan. Menetapkan nilai yang ditentukan dalam kode selalu diutamakan. Namun, ada juga Variabel-variabel lingkungan danMenggunakan `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 kredensional Anda. AWS SDK secara otomatis menggunakan AWS kredenal 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 kredenal sementara ini, AWS dapat memperpanjang validitas kredensialnya sehingga beban kerja Anda tidak terpengaruh.

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

Untuk mempelajari lebih lanjut tentang AWS kredensial dan menggunakan kunci akses, lihat kredenal AWS keamanan dan Mengelola kunci akses untuk pengguna IAM di Panduan Pengguna IAM.

Konfigurasikan 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 kredenal keamanan sementara. Namun, kami menyarankan Anda untuk selalu menggunakan kredenal keamanan sementara alih-alih kredensi jangka panjang. Untuk rekomendasi keamanan, lihat Praktik terbaik keamanan di IAM.

Untuk petunjuk tentang cara mendapatkan nilai-nilai ini, lihatMenggunakan kredensi 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...zrkuWJOgQs8IZZaIv2BXIa2R4Olgk
```

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...zrkuWJOgQs8IZZaIv2BXIa2R4Olgk
```

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...zrkuWJOgQs8IZZaIv2BXIa2R4Olgk
```

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

Asumsikan penyedia kredensi peran

> **Note**
>
> Untuk bantuan dalam memahami tata letak halaman pengaturan, atau dalam menafsirkan tabel Support by AWS SDKs and tools berikut, lihatMemahami halaman pengaturan panduan ini.

Dengan asumsi peran melibatkan penggunaan seperangkat kredensil 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

Konfigurasikan fungsi ini dengan menggunakan yang berikut:

`credential_source`- Pengaturan AWS **`config`** file bersama

Digunakan dalam EC2 instans Amazon atau penampung Amazon Elastic Container Service untuk menentukan tempat SDK atau alat dapat menemukan kredenal 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 kredenal sumber dari variabel lingkungan dan. `AWS_ACCESS_KEY_IDAWS_SECRET_ACCESS_KEY`
- Ec2 InstanceMetadata — Menentukan bahwa SDK atau alat adalah menggunakan peran IAM yang dilampirkan ke profil EC2 instance untuk mendapatkan kredensi sumber.
- EcsContainer— Menentukan bahwa SDK atau alat adalah menggunakan peran IAM yang dilampirkan ke wadah ECS untuk mendapatkan kredensi sumber.

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

Contoh pengaturan ini dalam `config` file untuk menunjukkan bahwa kredensil 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 (sepertiGAHT12345678), 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 kredensil yang memiliki izin untuk mengambil peran dalam profil ini.
- `credential_source`— Untuk menggunakan kredenal yang diidentifikasi oleh variabel lingkungan saat ini atau kredenal yang dilampirkan ke profil EC2 instans Amazon, 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, lihatBerbagi `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 kredensil. 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 kredensil untuk profil yang ditautkan. B Dalam hal ini, B profil menggunakan alat bantu kredenal yang disediakan oleh Menggunakan Peran IAM Di Mana Saja untuk mengautentikasi dan AWS SDKs alat untuk mendapatkan kredensil untuk SDK. AWS Kredensi 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 kredensil dari EC2 instans Amazon untuk mengambil peran yang ditentukan dan akan memperbarui kredensialnya secara otomatis.

`web_identity_token_file`- Pengaturan AWS **`config`** file bersama, **`AWS_WEB_IDENTITY_TOKEN_FILE`**- variabel lingkungan, **`aws.webIdentityTokenFile`**- Properti 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	`credential_source` tidak didukung. `duration_seconds` tidak didukung. `mfa_serial` tidak didukung.
SDK for Go V2 (1.x)	Ya	
SDK for Go 1.x (V1)	Ya	Untuk menggunakan pengaturan `config` file bersama, Anda harus mengaktifkan pemuatan dari file konfigurasi; lihat Sesi.
SDK for Java 2.x	Parsi	`mfa_serial` tidak didukung. `duration_seconds` tidak didukung.
SDK for Java 1.x	Parsi	`credential_source` tidak didukung. `mfa_serial` tidak didukung. Properti sistem JVM tidak didukung.
SDK untuk 3.x JavaScript	Ya	
SDK untuk 2.x JavaScript	Parsi	`credential_source` tidak didukung.
SDK para Kotlin	Ya	
SDK for .NET 4.x	Ya	
SDK for .NET 3.x	Ya	

SDK	Di	Catatan atau informasi lebih lanjut
SDK for PHP 3.x	Ya	
SDK untuk Python (Boto3)	Ya	
SDK for Ruby 3.x	Ya	
SDK untuk Rust	Ya	
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, lihatMemahami 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 klaster Amazon EKS di situs web Blog. AWS

Konfigurasikan 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 kredensional. 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 kredensional. 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, `AWS_CONTAINER_CREDENTIALS_FULL_URI` dan secara otomatis `AWS_CONTAINER_AUTHORIZATION_TOKEN` digunakan untuk otentikasi.
SDK for Java 1.x	Ya	Ketika Lambda SnapStart diaktifkan, `AWS_CONTAINER_CREDENTIALS_FULL_URI` dan secara otomatis `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

SDK	Di	Catatan atau informasi lebih lanjut
		`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, lihatMemahami 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 mengautentikasi dengan IAM Identity Center, portal mengembalikan kredensi jangka pendek untuk peran IAM yang terkait dengan pengguna tersebut. Untuk mempelajari cara SDK mendapatkan kredensi sementara dari konfigurasi dan menggunakannya untuk Layanan

AWS permintaan, lihat. 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 kredensional 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-session`Bagian `config` file digunakan untuk mengelompokkan variabel konfigurasi untuk memperoleh token akses SSO, yang kemudian dapat digunakan untuk memperoleh AWS kredensional. Untuk detail lebih lanjut tentang bagian ini dalam `config` file, lihatFormat file konfigurasi.

Contoh `config` file bersama berikut mengonfigurasi SDK atau alat menggunakan `dev` profil untuk meminta kredenal 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 kredensional. `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 kredensi 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 file`sso-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 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 ListAccountsAPI 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 ListAccountRolesAPI 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	Di	Catatan atau informasi lebih lanjut
SDK for Go 1.x (V1)	Ya	Untuk menggunakan pengaturan `config` file bersama, Anda harus mengaktifkan pemuatan dari file konfigurasi; lihat Sesi.
SDK for Java 2.x	Ya	Nilai konfigurasi juga didukung dalam `credentials` 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 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, lihatMemahami 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 kredenal 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.

Konfigurasikan 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 tabelSupport 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 kredenal yang valid, SDK akan mencoba menggunakan Amazon EC2 Instance Metadata Service (IMDS) untuk mengambil kredenal 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 `com.amazonaws.sdk.disableEc2MetadataV1` sebagai pengganti

SDK	Di	Catatan atau informasi lebih lanjut
		`aws.disableEc2MetadataV1` ; `aws.ec2MetadataServiceEndpoint` dan `aws.ec2MetadataServiceEndpointMode` 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. `[Amazon.Util.EC2InstanceMetadata]::EC2MetadataV1Disabled = $true`
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, lihatMemahami halaman pengaturan panduan ini.

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

Misalnya, IAM Roles Anywhere digunakan `credential_process` untuk mendapatkan kredensyal sementara atas nama aplikasi Anda. `credential_process`Untuk mengonfigurasi penggunaan ini, lihatMenggunakan Peran IAM Di Mana Saja untuk mengautentikasi dan AWS SDKs alat.

> **Note**
>
> Berikut ini menjelaskan metode sumber kredensional 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.

Konfigurasikan 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 kredensyal 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 kredensyal

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

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

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

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

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

`Expiration`Kuncinya adalah stempel RFC3339 waktu yang diformat. Jika `Expiration` kunci tidak ada dalam output alat, SDK 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 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	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	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
- 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, lihatMemahami 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. Saat Anda menggunakan AWS SDK dan layanan yang mendukung titik akhir berbasis akun, klien SDK akan membuat dan menggunakan titik akhir berbasis akun, bukan titik akhir 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

Konfigurasikan 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	Didu	Dirilis dalam versi SDK	Catatan atau informasi lebih lanjut
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 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	Tidak		
SDK para Swift	Ya	1.2.0	
Alat untuk PowerShell V5	Tidak		

SDK	Didu	Dirilis dalam versi SDK	Catatan atau informasi lebih lanjut
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, lihatMemahami 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 AWS Health Dashboard untuk mengidentifikasi secara unik aplikasi mana yang terkait dengan notifikasi.

Konfigurasikan 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 for C++	Ya	`config`file 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	Di	Catatan atau informasi lebih lanjut
SDK untuk 2.x JavaScript	Tida	
SDK para Kotlin	Ya	Properti sistem JVM adalah. `aws.userAgentAppId`
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	Tida	
Alat untuk PowerShell V4	Tida	

EC2 Metadata contoh Amazon

> **Note**
>
> Untuk bantuan dalam memahami tata letak halaman pengaturan, atau dalam menafsirkan tabel Support by AWS SDKs and tools berikut, lihatMemahami 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.

Konfigurasikan 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 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, lihatMemahami 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.

Konfigurasikan 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 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	Di	Catatan atau informasi lebih lanjut
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, lihatMemahami 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)

Konfigurasikan 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	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	Tida	
Alat untuk PowerShell V5	Ya	
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, lihatMemahami 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 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.

Konfigurasikan 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++	Tida	
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	Tida	
SDK for .NET 3.x	Tida	

SDK	Di	Catatan atau informasi lebih lanjut
SDK for PHP 3.x	Ya	
SDK untuk Python (Boto3)	Ya	
SDK for Ruby 3.x	Ya	
SDK untuk Rust	Ya	
SDK para Swift	Ya	
Alat untuk PowerShell V5	Tida	
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, lihatMemahami 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 AWSBerisi 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).

Konfigurasikan 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 `AWS_REGION` sebelum nilai apa pun di `AWS_DEFAULT_REGION` (kedua variabel diperiksa).
AWS CLI v1	Ya	AWS CLI v1 menggunakan variabel lingkungan bernama `AWS_DEFAULT_REGION` 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	Di	Catatan atau informasi lebih lanjut
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 `AWS_DEFAULT_REGION` untuk tujuan ini.
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 STS Titik akhir regional

> **Note**
>
> Untuk bantuan dalam memahami tata letak halaman pengaturan, atau dalam menafsirkan tabel Support by AWS SDKs and tools berikut, lihatMemahami 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 STSWilayah AWS dalam AWS Identity and Access Management Panduan Pengguna.

Ini adalah praktik AWS terbaik untuk menggunakan titik akhir Regional bila memungkinkan dan untuk mengkonfigurasi AndaWilayah 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 Regional`sts.us-west-2.amazonaws.com`, bukan titik `sts.amazonaws.com` akhir global. Untuk mengirim permintaan ke titik akhir global saat pengaturan ini diaktifkan, Anda dapat mengatur Wilayah ke`aws-global`.

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

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

Linux/macOS contoh pengaturan variabel lingkungan melalui baris perintah:

```
export AWS_STS_REGIONAL_ENDPOINTS=regional
```

Contoh Windows pengaturan variabel lingkungan melalui baris perintah:

```
setx AWS_STS_REGIONAL_ENDPOINTS regional
```

Support oleh AWS SDKs dan alat

> **Note**
>
> Ini adalah praktik AWS terbaik untuk menggunakan titik akhir Regional bila memungkinkan dan untuk mengkonfigurasi AndaWilayah 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	Mendukung pengaturan	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		Nilai pengaturan default	Target klien layanan default STS Endpoint	Perilaku fallback klien layanan	Catatan atau informasi lebih lanjut
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	Kegagalan permintaan	
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		Nilai pengaturan default	Target klien layanan default STS Endpoint	Perilaku fallback klien layanan	Catatan atau informasi lebih lanjut
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	
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, lihatMemahami 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.

Konfigurasikan 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 respons 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	Diduk	Algoritma checksum default	Algoritma checksum yang didukung	Catatan atau informasi lebih lanjut
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 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	`awscrt`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	Diduk	Algoritma checksum default	Algoritma checksum yang didukung	Catatan atau informasi lebih lanjut
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.
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, lihatMemahami halaman pengaturan panduan ini.

Konfigurasikan fungsi ini dengan menggunakan yang berikut:

`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 sesuai dengan FIPS 140-2. Jika pengaturan ini diaktifkan dan titik akhir FIPS tidak ada untuk layanan di Anda Wilayah AWS, AWS panggilan mungkin gagal. Titik akhir khusus layanandan `--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 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	Di	Catatan atau informasi lebih lanjut
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, lihatMemahami 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`

Konfigurasikan 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 `AWS_ENDPOINT_DISCOVERY_ENABLED` 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	Di	Catatan atau informasi lebih lanjut
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	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, lihatMemahami halaman pengaturan panduan ini.

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

Konfigurasikan fungsi ini dengan menggunakan yang berikut ini:

`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-alat.

Nilai default: `json`

Nilai yang valid:

- **`json`**— Output diformat sebagai string JSON.
- **`yaml`**— Output diformat sebagai string YAMG.
- **`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, sepertigrep,sed, atauawk.
- **`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	`api_versions` tidak didukung.
SDK for C++	Ya	
SDK for Go V2 (1.x)	Parsi	`api_versions` dan `parameter_validation` tidak didukung.
SDK for Go 1.x (V1)	Parsi	`api_versions` dan `parameter_validation` 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	Di	Catatan atau informasi lebih lanjut
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, lihatMemahami 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.

Konfigurasikan 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.	false	"https://data.service.us-we

Awalan host	Masukan	Titik akhir klien	Menetapkan nilai	Titik akhir yang dihasilkan
		amazonaws.com"		st-2.amazonaws.com"
“{Bucket} - {AccountId}.”	Ember: “amzn-s3-demo-bucket1",:" 123456789012" AccountId	"https://service.us-west-2.amazonaws.com"	false	"https://amzn-s3-demo-bucket1-123456789012.service.us-west-2.amazonaws.com"
“data.”	{}	"https://override.us-west-2.amazonaws.com"(sebagai titik akhir override)	true	"https://override.us-west-2.amazonaws.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	Di	Catatan atau informasi lebih lanjut
SDK for Java 2.x	Tida	Pengaturan tidak didukung, tetapi dapat dikonfigurasi dalam kode pada klien menggunakan: `SdkAdvancedClientOption.DISABLE_HOST_PREFIX_INJECTION`.
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	Di	Catatan atau informasi lebih lanjut
SDK para Swift	Tida	
Alat untuk PowerShell V5	Tida	Pengaturan tidak didukung, tetapi dapat disertakan dalam cmdlet tertentu menggunakan parameter. `-ClientConfig @{DisableHostPrefixInjection = $true}`
Alat untuk PowerShell V4	Tida	Pengaturan tidak didukung, tetapi dapat disertakan dalam cmdlet tertentu menggunakan parameter. `-ClientConfig @{DisableHostPrefixInjection = $true}`

Klien IMDS

> **Note**
>
> Untuk bantuan dalam memahami tata letak halaman pengaturan, atau dalam menafsirkan tabel Support by AWS SDKs and tools berikut, lihatMemahami 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.

Konfigurasikan 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` sama`IPv4`, maka titik akhir default adalah. `http://169.254.169.254` Jika `endpoint_mode` sama`IPv6`, 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, lihatMemahami 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

Konfigurasikan 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**
>
> `adaptive`Mode 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: 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 peningkat an 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 permintaan`i`:

```
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 <= b <= 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 `com.amazonaws.sdk.maxAttempts` sebagai penggantiaws.maxAttempts ; gunakan `com.amazonaws.sdk.retryMode` sebagai pengganti. `aws.retryMode`
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, lihatMemahami 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.

Konfigurasikan 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 memampatkan muatan sebelum mengirim permintaan.

Nilai default: `false`

Nilai yang valid:

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

`request_min_compression_size_bytes`- 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, lihatMemahami 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.

Konfigurasikan 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 pengenal 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 pengenal, 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.

Konfigurasikan 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
```

Konfigurasikan 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 pengenal 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, lihatFormat 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 telepon`elastic_beanstalk =` . Untuk daftar semua kunci pengenal layanan yang akan digunakan di `services` bagian ini, lihatPengidentifikasi 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
```

Konfigurasikan titik akhir di profil menggunakan kredensi berbasis peran

Jika profil Anda memiliki kredensyal 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, lihatAsumsikan 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 Anda 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 pengenal layanan dalam `services` bagian file bersama`config`.
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++	Tida	
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, lihatTitik akhir khusus layanan.

serviceId	Ku pe la ur fil be AV **co**	Variabel lingkungan **AWS_ENDPOINT_URL_<SERVICE>**
AccessAnalyzer	ac ly	AWS_ENDPOINT_URL_ACCESSANALYZER
Account	ac	AWS_ENDPOINT_URL_ACCOUNT
ACM	ac	AWS_ENDPOINT_URL_ACM
ACM PCA	ac	AWS_ENDPOINT_URL_ACM_PCA
Alexa For Business	a: _F	AWS_ENDPOINT_URL_ALEXA_FOR_BUSINESS
amp	ar	AWS_ENDPOINT_URL_AMP
Amplify	ar	AWS_ENDPOINT_URL_AMPLIFY
AmplifyBackend	ar cl	AWS_ENDPOINT_URL_AMPLIFYBACKEND
AmplifyUIBuilder	ar bu	AWS_ENDPOINT_URL_AMPLIFYUIBUILDER
API Gateway	ap ay	AWS_ENDPOINT_URL_API_GATEWAY
ApiGatewayManagementApi	ap yr nt	AWS_ENDPOINT_URL_APIGATEWAYMANAGEMENTAPI

serviceId	Ku pe la ur fil be AV **co**	Variabel lingkungan **AWS_ENDPOINT_URL_<SERVICE>**
ApiGatewayV2	ap y\	AWS_ENDPOINT_URL_APIGATEWAYV2
AppConfig	ap	AWS_ENDPOINT_URL_APPCONFIG
AppConfigData	ap da	AWS_ENDPOINT_URL_APPCONFIGDATA
AppFabric	ap	AWS_ENDPOINT_URL_APPFABRIC
Appflow	ap	AWS_ENDPOINT_URL_APPFLOW
AppIntegrations	ap at	AWS_ENDPOINT_URL_APPINTEGRATIONS
Application Auto Scaling	ap or ca	AWS_ENDPOINT_URL_APPLICATION_AUTO_SCALING
Application Insights	ap or ts	AWS_ENDPOINT_URL_APPLICATION_INSIGHTS
ApplicationCostProfiler	ap or f:	AWS_ENDPOINT_URL_APPLICATIONCOSTPROFILER
App Mesh	ap	AWS_ENDPOINT_URL_APP_MESH

serviceId	Ku pe la ur fil be AV **co**	Variabel lingkungan **AWS_ENDPOINT_URL_<SERVICE>**
AppRunner	ap	AWS_ENDPOINT_URL_APPRUNNER
AppStream	ap	AWS_ENDPOINT_URL_APPSTREAM
AppSync	ap	AWS_ENDPOINT_URL_APPSYNC
ARC Zonal Shift	ar _s	AWS_ENDPOINT_URL_ARC_ZONAL_SHIFT
Artifact	ar	AWS_ENDPOINT_URL_ARTIFACT
Athena	at	AWS_ENDPOINT_URL_ATHENA
AuditManager	au ge	AWS_ENDPOINT_URL_AUDITMANAGER
Auto Scaling	au in	AWS_ENDPOINT_URL_AUTO_SCALING
Auto Scaling Plans	au in	AWS_ENDPOINT_URL_AUTO_SCALING_PLANS
b2bi	b2	AWS_ENDPOINT_URL_B2BI
Backup	ba	AWS_ENDPOINT_URL_BACKUP

serviceId	Ku pe la ur fil be AV **co**	Variabel lingkungan **AWS_ENDPOINT_URL_<SERVICE>**
Backup Gateway	ba te	AWS_ENDPOINT_URL_BACKUP_GATEWAY
BackupStorage	ba ra	AWS_ENDPOINT_URL_BACKUPSTORAGE
Batch	ba	AWS_ENDPOINT_URL_BATCH
BCM Data Exports	bc ex	AWS_ENDPOINT_URL_BCM_DATA_EXPORTS
Bedrock	be	AWS_ENDPOINT_URL_BEDROCK
Bedrock Agent	be ge	AWS_ENDPOINT_URL_BEDROCK_AGENT
Bedrock Agent Runtime	be ge ir	AWS_ENDPOINT_URL_BEDROCK_AGENT_RUNTIME
Bedrock Runtime	be ur	AWS_ENDPOINT_URL_BEDROCK_RUNTIME
billingconductor	bi nc	AWS_ENDPOINT_URL_BILLINGCONDUCTOR
Braket	bi	AWS_ENDPOINT_URL_BRAKET
Budgets	bu	AWS_ENDPOINT_URL_BUDGETS

serviceId	Ku pe la ur fil be AV **co**	Variabel lingkungan **AWS_ENDPOINT_URL_<SERVICE>**
Cost Explorer	co ot	AWS_ENDPOINT_URL_COST_EXPLORER
chatbot	ch	AWS_ENDPOINT_URL_CHATBOT
Chime	ch	AWS_ENDPOINT_URL_CHIME
Chime SDK Identity	ch _i	AWS_ENDPOINT_URL_CHIME_SDK_IDENTITY
Chime SDK Media Pipelines	ch _m pe	AWS_ENDPOINT_URL_CHIME_SDK_MEDIA_PIPELINES
Chime SDK Meetings	ch _m	AWS_ENDPOINT_URL_CHIME_SDK_MEETINGS
Chime SDK Messaging	ch _m g	AWS_ENDPOINT_URL_CHIME_SDK_MESSAGING
Chime SDK Voice	ch _v	AWS_ENDPOINT_URL_CHIME_SDK_VOICE
CleanRooms	cl s	AWS_ENDPOINT_URL_CLEANROOMS

serviceId	Ku pe la ur fil be AV **co**	Variabel lingkungan **AWS_ENDPOINT_URL_<SERVICE>**
CleanRoomsML	cl sr	AWS_ENDPOINT_URL_CLEANROOMSML
Cloud9	cl	AWS_ENDPOINT_URL_CLOUD9
CloudControl	cl rc	AWS_ENDPOINT_URL_CLOUDCONTROL
CloudDirectory	cl cl	AWS_ENDPOINT_URL_CLOUDDIRECTORY
CloudFormation	cl af	AWS_ENDPOINT_URL_CLOUDFORMATION
CloudFront	cl t	AWS_ENDPOINT_URL_CLOUDFRONT
CloudFront KeyValueStore	cl t_ es	AWS_ENDPOINT_URL_CLOUDFRONT_KEYVALUESTORE
CloudHSM	cl	AWS_ENDPOINT_URL_CLOUDHSM
CloudHSM V2	cl v2	AWS_ENDPOINT_URL_CLOUDHSM_V2
CloudSearch	cl ch	AWS_ENDPOINT_URL_CLOUDSEARCH

serviceId	Ku pe la ur fil be AV **co**	Variabel lingkungan **AWS_ENDPOINT_URL_<SERVICE>**
CloudSearch Domain	c: ch	AWS_ENDPOINT_URL_CLOUDSEARCH_DOMAIN
CloudTrail	c: l	AWS_ENDPOINT_URL_CLOUDTRAIL
CloudTrail Data	c: l_	AWS_ENDPOINT_URL_CLOUDTRAIL_DATA
CloudWatch	c: h	AWS_ENDPOINT_URL_CLOUDWATCH
codeartifact	co ac	AWS_ENDPOINT_URL_CODEARTIFACT
CodeBuild	co	AWS_ENDPOINT_URL_CODEBUILD
CodeCatalyst	co ys	AWS_ENDPOINT_URL_CODECATALYST
CodeCommit	co t	AWS_ENDPOINT_URL_CODECOMMIT
CodeDeploy	co y	AWS_ENDPOINT_URL_CODEDEPLOY
CodeGuru Reviewer	co re	AWS_ENDPOINT_URL_CODEGURU_REVIEWER

serviceId	Ku pe la ur fil be AV **co**	Variabel lingkungan **AWS_ENDPOINT_URL_<SERVICE>**
CodeGuru Security	co se	AWS_ENDPOINT_URL_CODEGURU_SECURITY
CodeGuruProfiler	co ro	AWS_ENDPOINT_URL_CODEGURUPROFILER
CodePipeline	co ir	AWS_ENDPOINT_URL_CODEPIPELINE
CodeStar	co	AWS_ENDPOINT_URL_CODESTAR
CodeStar connections	co co ns	AWS_ENDPOINT_URL_CODESTAR_CONNECTIONS
codestar notifications	co no ic	AWS_ENDPOINT_URL_CODESTAR_NOTIFICATIONS
Cognito Identity	co de	AWS_ENDPOINT_URL_COGNITO_IDENTITY
Cognito Identity Provider	co de ro	AWS_ENDPOINT_URL_COGNITO_IDENTITY_PROVIDER
Cognito Sync	co yr	AWS_ENDPOINT_URL_COGNITO_SYNC

serviceId	Ku pe la ur fil be AV **co**	Variabel lingkungan **AWS_ENDPOINT_URL_<SERVICE>**
Comprehend	co d	AWS_ENDPOINT_URL_COMPREHEND
ComprehendMedical	co dr	AWS_ENDPOINT_URL_COMPREHENDMEDICAL
Compute Optimizer	co pt	AWS_ENDPOINT_URL_COMPUTE_OPTIMIZER
Config Service	co rv	AWS_ENDPOINT_URL_CONFIG_SERVICE
Connect	co	AWS_ENDPOINT_URL_CONNECT
Connect Contact Lens	co or ns	AWS_ENDPOINT_URL_CONNECT_CONTACT_LENS
ConnectCampaigns	co mp	AWS_ENDPOINT_URL_CONNECTCAMPAIGNS
ConnectCases	co se	AWS_ENDPOINT_URL_CONNECTCASES
ConnectParticipant	co rt	AWS_ENDPOINT_URL_CONNECTPARTICIPANT
ControlTower	co we	AWS_ENDPOINT_URL_CONTROLTOWER

serviceId	Ku pe la ur fil be AV **co**	Variabel lingkungan **AWS_ENDPOINT_URL_<SERVICE>**
Cost Optimization Hub	co m: hu	AWS_ENDPOINT_URL_COST_OPTIMIZATION_HUB
Cost and Usage Report Service	co us o: ce	AWS_ENDPOINT_URL_COST_AND_USAGE_REPO RT_SERVICE
Customer Profiles	cu p:	AWS_ENDPOINT_URL_CUSTOMER_PROFILES
DataBrew	da	AWS_ENDPOINT_URL_DATABREW
DataExchange	da ng	AWS_ENDPOINT_URL_DATAEXCHANGE
Data Pipeline	da l:	AWS_ENDPOINT_URL_DATA_PIPELINE
DataSync	da	AWS_ENDPOINT_URL_DATASYNC
DataZone	da	AWS_ENDPOINT_URL_DATAZONE
DAX	da	AWS_ENDPOINT_URL_DAX
Detective	de	AWS_ENDPOINT_URL_DETECTIVE

serviceId	Ku pe la ur fil be A\ **co**	Variabel lingkungan **AWS_ENDPOINT_URL_<SERVICE>**
Device Farm	de rn	AWS_ENDPOINT_URL_DEVICE_FARM
DevOps Guru	de ru	AWS_ENDPOINT_URL_DEVOPS_GURU
Direct Connect	di nn	AWS_ENDPOINT_URL_DIRECT_CONNECT
Application Discovery Service	ap or e: ce	AWS_ENDPOINT_URL_APPLICATION_DISCOVERY_SERVICE
DLM	dl	AWS_ENDPOINT_URL_DLM
Database Migration Service	da m: _s	AWS_ENDPOINT_URL_DATABASE_MIGRATION_SERVICE
DocDB	do	AWS_ENDPOINT_URL_DOCDB
DocDB Elastic	do st	AWS_ENDPOINT_URL_DOCDB_ELASTIC
drs	di	AWS_ENDPOINT_URL_DRS
Directory Service	di _s	AWS_ENDPOINT_URL_DIRECTORY_SERVICE

serviceId	Ku pe la ur fil be AV **co**	Variabel lingkungan **AWS_ENDPOINT_URL_<SERVICE>**
DynamoDB	dy	AWS_ENDPOINT_URL_DYNAMODB
DynamoDB Streams	dy st	AWS_ENDPOINT_URL_DYNAMODB_STREAMS
EBS	eb	AWS_ENDPOINT_URL_EBS
EC2	ec	AWS_ENDPOINT_URL_EC2
EC2 Instance Connect	ec no ct	AWS_ENDPOINT_URL_EC2_INSTANCE_CONNECT
ECR	ec	AWS_ENDPOINT_URL_ECR
ECR PUBLIC	ec c	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	el nf	AWS_ENDPOINT_URL_ELASTIC_INFERENCE
ElastiCache	el he	AWS_ENDPOINT_URL_ELASTICACHE

serviceId	Ku pe la ur fil be AV **co**	Variabel lingkungan **AWS_ENDPOINT_URL_<SERVICE>**
Elastic Beanstalk	el ea	AWS_ENDPOINT_URL_ELASTIC_BEANSTALK
Elastic Transcoder	el ra	AWS_ENDPOINT_URL_ELASTIC_TRANSCODER
Elastic Load Balancing	el oa c:	AWS_ENDPOINT_URL_ELASTIC_LOAD_BALANCING
Elastic Load Balancing v2	el oa c:	AWS_ENDPOINT_URL_ELASTIC_LOAD_BALANCING_V2
EMR	er	AWS_ENDPOINT_URL_EMR
EMR containers	er in	AWS_ENDPOINT_URL_EMR_CONTAINERS
EMR Serverless	er r:	AWS_ENDPOINT_URL_EMR_SERVERLESS
EntityResolution	er o:	AWS_ENDPOINT_URL_ENTITYRESOLUTION
Elasticsearch Service	el a: ic	AWS_ENDPOINT_URL_ELASTICSEARCH_SERVICE

serviceId	Ku pe la ur fil be AV **co**	Variabel lingkungan **AWS_ENDPOINT_URL_<SERVICE>**
EventBridge	ev ge	AWS_ENDPOINT_URL_EVENTBRIDGE
Evidently	ev	AWS_ENDPOINT_URL_EVIDENTLY
finspace	fi	AWS_ENDPOINT_URL_FINSPACE
finspace data	fi da	AWS_ENDPOINT_URL_FINSPACE_DATA
Firehose	fi	AWS_ENDPOINT_URL_FIREHOSE
fis	fi	AWS_ENDPOINT_URL_FIS
FMS	fr	AWS_ENDPOINT_URL_FMS
forecast	fo	AWS_ENDPOINT_URL_FORECAST
forecastquery	fo ue	AWS_ENDPOINT_URL_FORECASTQUERY
FraudDetector	fi ct	AWS_ENDPOINT_URL_FRAUDDETECTOR
FreeTier	fi	AWS_ENDPOINT_URL_FREETIER
FSx	fs	AWS_ENDPOINT_URL_FSX
GameLift	ga	AWS_ENDPOINT_URL_GAMELIFT

serviceId	Ku pe la ur fil be AV **co**	Variabel lingkungan **AWS_ENDPOINT_URL_<SERVICE>**
Glacier	gl	AWS_ENDPOINT_URL_GLACIER
Global Accelerator	gl ce	AWS_ENDPOINT_URL_GLOBAL_ACCELERATOR
Glue	gl	AWS_ENDPOINT_URL_GLUE
grafana	gr	AWS_ENDPOINT_URL_GRAFANA
Greengrass	gr s	AWS_ENDPOINT_URL_GREENGRASS
GreengrassV2	gr sv	AWS_ENDPOINT_URL_GREENGRASSV2
GroundStation	gr ti	AWS_ENDPOINT_URL_GROUNDSTATION
GuardDuty	gu	AWS_ENDPOINT_URL_GUARDDUTY
Health	he	AWS_ENDPOINT_URL_HEALTH
HealthLake	he e	AWS_ENDPOINT_URL_HEALTHLAKE
Honeycode	ho	AWS_ENDPOINT_URL_HONEYCODE

serviceId	Ku pe la ur fil be AV **co**	Variabel lingkungan **AWS_ENDPOINT_URL_<SERVICE>**
IAM	ia	AWS_ENDPOINT_URL_IAM
identitystore	id to	AWS_ENDPOINT_URL_IDENTITYSTORE
imagebuilder	im de	AWS_ENDPOINT_URL_IMAGEBUILDER
ImportExport	im ox	AWS_ENDPOINT_URL_IMPORTEXPORT
Inspector	in	AWS_ENDPOINT_URL_INSPECTOR
Inspector Scan	in _s	AWS_ENDPOINT_URL_INSPECTOR_SCAN
Inspector2	in 2	AWS_ENDPOINT_URL_INSPECTOR2
InternetMonitor	in or	AWS_ENDPOINT_URL_INTERNETMONITOR
IoT	io	AWS_ENDPOINT_URL_IOT
IoT Data Plane	io pl	AWS_ENDPOINT_URL_IOT_DATA_PLANE

serviceId	Ku pe la ur fil be AV **co**	Variabel lingkungan **AWS_ENDPOINT_URL_<SERVICE>**
IoT Jobs Data Plane	io da e	AWS_ENDPOINT_URL_IOT_JOBS_DATA_PLANE
IoT 1Click Devices Service	io k_ _s	AWS_ENDPOINT_URL_IOT_1CLICK_DEVICES_ SERVICE
IoT 1Click Projects	io k_ s	AWS_ENDPOINT_URL_IOT_1CLICK_PROJECTS
IoTAnalytics	io io	AWS_ENDPOINT_URL_IOTANALYTICS
IotDeviceAdvisor	io ad	AWS_ENDPOINT_URL_IOTDEVICEADVISOR
IoT Events	io s	AWS_ENDPOINT_URL_IOT_EVENTS
IoT Events Data	io s_	AWS_ENDPOINT_URL_IOT_EVENTS_DATA
IoTFleetHub	io ub	AWS_ENDPOINT_URL_IOTFLEETHUB
IoTFleetWise	io is	AWS_ENDPOINT_URL_IOTFLEETWISE

serviceId	Ku pe la ur fil be AV **co**	Variabel lingkungan **AWS_ENDPOINT_URL_<SERVICE>**
IoTSecureTunneling	ic tu	AWS_ENDPOINT_URL_IOTSECURETUNNELING
IoTSiteWise	ic se	AWS_ENDPOINT_URL_IOTSITEWISE
IoTThingsGraph	ic gr	AWS_ENDPOINT_URL_IOTTHINGSGRAPH
IoTTwinMaker	ic ke	AWS_ENDPOINT_URL_IOTTWINMAKER
IoT Wireless	ic es	AWS_ENDPOINT_URL_IOT_WIRELESS
ivs	iv	AWS_ENDPOINT_URL_IVS
IVS RealTime	iv ir	AWS_ENDPOINT_URL_IVS_REALTIME
ivschat	iv	AWS_ENDPOINT_URL_IVSCHAT
Kafka	ka	AWS_ENDPOINT_URL_KAFKA
KafkaConnect	ka ec	AWS_ENDPOINT_URL_KAFKACONNECT
kendra	ke	AWS_ENDPOINT_URL_KENDRA

serviceId	Ku pe la ur fil be AV **co**	Variabel lingkungan **AWS_ENDPOINT_URL_<SERVICE>**
Kendra Ranking	ke nl	AWS_ENDPOINT_URL_KENDRA_RANKING
Keyspaces	ke	AWS_ENDPOINT_URL_KEYSPACES
Kinesis	ki	AWS_ENDPOINT_URL_KINESIS
Kinesis Video Archived Media	ki id iv a	AWS_ENDPOINT_URL_KINESIS_VIDEO_ARCHI VED_MEDIA
Kinesis Video Media	ki id a	AWS_ENDPOINT_URL_KINESIS_VIDEO_MEDIA
Kinesis Video Signaling	ki id a.	AWS_ENDPOINT_URL_KINESIS_VIDEO_SIGNALING
Kinesis Video WebRTC Storage	ki id to e	AWS_ENDPOINT_URL_KINESIS_VIDEO_WEBRT C_STORAGE
Kinesis Analytics	ki na	AWS_ENDPOINT_URL_KINESIS_ANALYTICS

serviceId	Ku pe la ur fil be AV **co**	Variabel lingkungan **AWS_ENDPOINT_URL_<SERVICE>**
Kinesis Analytics V2	k: na v2	AWS_ENDPOINT_URL_KINESIS_ANALYTICS_V2
Kinesis Video	k: id	AWS_ENDPOINT_URL_KINESIS_VIDEO
KMS	kr	AWS_ENDPOINT_URL_KMS
LakeFormation	la t:	AWS_ENDPOINT_URL_LAKEFORMATION
Lambda	la	AWS_ENDPOINT_URL_LAMBDA
Launch Wizard	la za	AWS_ENDPOINT_URL_LAUNCH_WIZARD
Lex Model Building Service	le _l _s	AWS_ENDPOINT_URL_LEX_MODEL_BUILDING_ SERVICE
Lex Runtime Service	le me e	AWS_ENDPOINT_URL_LEX_RUNTIME_SERVICE
Lex Models V2	le s_	AWS_ENDPOINT_URL_LEX_MODELS_V2
Lex Runtime V2	le me	AWS_ENDPOINT_URL_LEX_RUNTIME_V2

serviceId	Ku pe la ur fil be AV **co**	Variabel lingkungan **AWS_ENDPOINT_URL_<SERVICE>**
License Manager	l: ar	AWS_ENDPOINT_URL_LICENSE_MANAGER
License Manager Linux Subscriptions	l: ar nu r:	AWS_ENDPOINT_URL_LICENSE_MANAGER_LIN UX_SUBSCRIPTIONS
License Manager User Subscriptions	l: ar e: ip	AWS_ENDPOINT_URL_LICENSE_MANAGER_USE R_SUBSCRIPTIONS
Lightsail	l:	AWS_ENDPOINT_URL_LIGHTSAIL
Location	lo	AWS_ENDPOINT_URL_LOCATION
CloudWatch Logs	c: h_	AWS_ENDPOINT_URL_CLOUDWATCH_LOGS
LookoutEquipment	lo u:	AWS_ENDPOINT_URL_LOOKOUTEQUIPMENT
LookoutMetrics	lo t:	AWS_ENDPOINT_URL_LOOKOUTMETRICS
LookoutVision	lo s:	AWS_ENDPOINT_URL_LOOKOUTVISION

serviceId	Ku pe la ur fil be AV **co**	Variabel lingkungan **AWS_ENDPOINT_URL_<SERVICE>**
m2	m2	AWS_ENDPOINT_URL_M2
Machine Learning	ma ea	AWS_ENDPOINT_URL_MACHINE_LEARNING
Macie2	ma	AWS_ENDPOINT_URL_MACIE2
ManagedBlockchain	ma oc	AWS_ENDPOINT_URL_MANAGEDBLOCKCHAIN
ManagedBlockchain Query	ma oc qu	AWS_ENDPOINT_URL_MANAGEDBLOCKCHAIN_QUERY
Marketplace Agreement	ma ce er	AWS_ENDPOINT_URL_MARKETPLACE_AGREEMENT
Marketplace Catalog	ma ce g	AWS_ENDPOINT_URL_MARKETPLACE_CATALOG
Marketplace Deployment	ma ce me	AWS_ENDPOINT_URL_MARKETPLACE_DEPLOYMENT

serviceId	Ku pe la ur fil be AV **co**	Variabel lingkungan **AWS_ENDPOINT_URL_<SERVICE>**
Marketplace Entitleme nt Service	ma ce er v:	AWS_ENDPOINT_URL_MARKETPLACE_ENTITLE MENT_SERVICE
Marketplace Commerce Analytics	ma ce ce ic	AWS_ENDPOINT_URL_MARKETPLACE_COMMERC E_ANALYTICS
MediaConnect	me ec	AWS_ENDPOINT_URL_MEDIACONNECT
MediaConvert	me e:	AWS_ENDPOINT_URL_MEDIACONVERT
MediaLive	me	AWS_ENDPOINT_URL_MEDIALIVE
MediaPackage	me ag	AWS_ENDPOINT_URL_MEDIAPACKAGE
MediaPackage Vod	me ag	AWS_ENDPOINT_URL_MEDIAPACKAGE_VOD
MediaPackageV2	me ag	AWS_ENDPOINT_URL_MEDIAPACKAGEV2

serviceId	Ku pe la ur fil be AV **co**	Variabel lingkungan **AWS_ENDPOINT_URL_<SERVICE>**
MediaStore	me e	AWS_ENDPOINT_URL_MEDIASTORE
MediaStore Data	me e_	AWS_ENDPOINT_URL_MEDIASTORE_DATA
MediaTailor	me o:	AWS_ENDPOINT_URL_MEDIATAILOR
Medical Imaging	me ma	AWS_ENDPOINT_URL_MEDICAL_IMAGING
MemoryDB	me	AWS_ENDPOINT_URL_MEMORYDB
Marketplace Metering	ma ce ng	AWS_ENDPOINT_URL_MARKETPLACE_METERING
Migration Hub	m: _I	AWS_ENDPOINT_URL_MIGRATION_HUB
mgn	mg	AWS_ENDPOINT_URL_MGN
Migration Hub Refactor Spaces	m: _I cf es	AWS_ENDPOINT_URL_MIGRATION_HUB_REFAC TOR_SPACES

serviceId	Ku pe la ur fil be AV **co**	Variabel lingkungan **AWS_ENDPOINT_URL_<SERVICE>**
MigrationHub Config	mi hu g	AWS_ENDPOINT_URL_MIGRATIONHUB_CONFIG
MigrationHubOrches trator	mi hu t:	AWS_ENDPOINT_URL_MIGRATIONHUBORCHESTRATOR
MigrationHubStrategy	mi hu gy	AWS_ENDPOINT_URL_MIGRATIONHUBSTRATEGY
Mobile	mo	AWS_ENDPOINT_URL_MOBILE
mq	mo	AWS_ENDPOINT_URL_MQ
MTurk	mt	AWS_ENDPOINT_URL_MTURK
MWAA	mw	AWS_ENDPOINT_URL_MWAA
Neptune	ne	AWS_ENDPOINT_URL_NEPTUNE
Neptune Graph	ne ra	AWS_ENDPOINT_URL_NEPTUNE_GRAPH
neptunedata	ne ta	AWS_ENDPOINT_URL_NEPTUNEDATA
Network Firewall	ne i:	AWS_ENDPOINT_URL_NETWORK_FIREWALL

serviceId	Ku pe la ur fil be AV **co**	Variabel lingkungan **AWS_ENDPOINT_URL_<SERVICE>**
NetworkManager	ne na	AWS_ENDPOINT_URL_NETWORKMANAGER
NetworkMonitor	ne n:	AWS_ENDPOINT_URL_NETWORKMONITOR
nimble	n:	AWS_ENDPOINT_URL_NIMBLE
OAM	oa	AWS_ENDPOINT_URL_OAM
Omics	or	AWS_ENDPOINT_URL_OMICS
OpenSearch	op h	AWS_ENDPOINT_URL_OPENSEARCH
OpenSearchServerless	op hs ss	AWS_ENDPOINT_URL_OPENSEARCHSERVERLESS
OpsWorks	op	AWS_ENDPOINT_URL_OPSWORKS
OpsWorksCM	op m	AWS_ENDPOINT_URL_OPSWORKSCM
Organizations	or id	AWS_ENDPOINT_URL_ORGANIZATIONS
OSIS	os	AWS_ENDPOINT_URL_OSIS
Outposts	ou	AWS_ENDPOINT_URL_OUTPOSTS

serviceId	Ku pe la ur fil be AV **co**	Variabel lingkungan **AWS_ENDPOINT_URL_<SERVICE>**
p8data	p8	AWS_ENDPOINT_URL_P8DATA
p8data	p8	AWS_ENDPOINT_URL_P8DATA
Panorama	pa	AWS_ENDPOINT_URL_PANORAMA
Payment Cryptography	pa ry hy	AWS_ENDPOINT_URL_PAYMENT_CRYPTOGRAPHY
Payment Cryptography Data	pa ry hy	AWS_ENDPOINT_URL_PAYMENT_CRYPTOGRAPHY_DATA
Pca Connector Ad	pc ct	AWS_ENDPOINT_URL_PCA_CONNECTOR_AD
Personalize	pe ze	AWS_ENDPOINT_URL_PERSONALIZE
Personalize Events	pe ze	AWS_ENDPOINT_URL_PERSONALIZE_EVENTS
Personalize Runtime	pe ze e	AWS_ENDPOINT_URL_PERSONALIZE_RUNTIME
PI	pi	AWS_ENDPOINT_URL_PI

serviceId	Ku pe la ur fil be AV **co**	Variabel lingkungan **AWS_ENDPOINT_URL_<SERVICE>**
Pinpoint	p:	AWS_ENDPOINT_URL_PINPOINT
Pinpoint Email	p: er	AWS_ENDPOINT_URL_PINPOINT_EMAIL
Pinpoint SMS Voice	p: sr	AWS_ENDPOINT_URL_PINPOINT_SMS_VOICE
Pinpoint SMS Voice V2	p: sr _v	AWS_ENDPOINT_URL_PINPOINT_SMS_VOICE_V2
Pipes	p:	AWS_ENDPOINT_URL_PIPES
Polly	po	AWS_ENDPOINT_URL_POLLY
Pricing	p:	AWS_ENDPOINT_URL_PRICING
PrivateNetworks	p: tw	AWS_ENDPOINT_URL_PRIVATENETWORKS
Proton	p:	AWS_ENDPOINT_URL_PROTON
QBusiness	qb	AWS_ENDPOINT_URL_QBUSINESS
QConnect	qc	AWS_ENDPOINT_URL_QCONNECT
QLDB	qj	AWS_ENDPOINT_URL_QLDB

serviceId	Ku pe la ur fil be AV **co**	Variabel lingkungan **AWS_ENDPOINT_URL_<SERVICE>**
QLDB Session	q: ic	AWS_ENDPOINT_URL_QLDB_SESSION
QuickSight	qu t	AWS_ENDPOINT_URL_QUICKSIGHT
RAM	ra	AWS_ENDPOINT_URL_RAM
rbin	rb	AWS_ENDPOINT_URL_RBIN
RDS	rd	AWS_ENDPOINT_URL_RDS
RDS Data	rd	AWS_ENDPOINT_URL_RDS_DATA
Redshift	re	AWS_ENDPOINT_URL_REDSHIFT
Redshift Data	re da	AWS_ENDPOINT_URL_REDSHIFT_DATA
Redshift Serverless	re se s	AWS_ENDPOINT_URL_REDSHIFT_SERVERLESS
Rekognition	re on	AWS_ENDPOINT_URL_REKOGNITION
repostspace	re ce	AWS_ENDPOINT_URL_REPOSTSPACE

serviceId	Ku pe la ur fil be AV **co**	Variabel lingkungan **AWS_ENDPOINT_URL_<SERVICE>**
resiliencehub	re eh	AWS_ENDPOINT_URL_RESILIENCEHUB
Resource Explorer 2	re ex 2	AWS_ENDPOINT_URL_RESOURCE_EXPLORER_2
Resource Groups	re gr	AWS_ENDPOINT_URL_RESOURCE_GROUPS
Resource Groups Tagging API	re gr gg	AWS_ENDPOINT_URL_RESOURCE_GROUPS_TAG GING_API
RoboMaker	ro	AWS_ENDPOINT_URL_ROBOMAKER
RolesAnywhere	ro he	AWS_ENDPOINT_URL_ROLESANYWHERE
Route 53	ro	AWS_ENDPOINT_URL_ROUTE_53
Route53 Recovery Cluster	ro ec lu	AWS_ENDPOINT_URL_ROUTE53_RECOVERY_CLUSTER

serviceId	Ku pe la ur fil be AV **co**	Variabel lingkungan **AWS_ENDPOINT_URL_<SERVICE>**
Route53 Recovery Control Config	rc ec or nf	AWS_ENDPOINT_URL_ROUTE53_RECOVERY_CONTROL_CONFIG
Route53 Recovery Readiness	rc ec ea	AWS_ENDPOINT_URL_ROUTE53_RECOVERY_READINESS
Route 53 Domains	rc do	AWS_ENDPOINT_URL_ROUTE_53_DOMAINS
Route53Resolver	rc so	AWS_ENDPOINT_URL_ROUTE53RESOLVER
RUM	ru	AWS_ENDPOINT_URL_RUM
S3	s3	AWS_ENDPOINT_URL_S3
S3 Control	s3 l	AWS_ENDPOINT_URL_S3_CONTROL
S3Outposts	s3 s	AWS_ENDPOINT_URL_S3OUTPOSTS
SageMaker	sa	AWS_ENDPOINT_URL_SAGEMAKER

serviceId	Ku pe la ur fil be AV **co**	Variabel lingkungan **AWS_ENDPOINT_URL_<SERVICE>**
SageMaker A2I Runtime	sa _a ir	AWS_ENDPOINT_URL_SAGEMAKER_A2I_RUNTIME
Sagemaker Edge	sa _e	AWS_ENDPOINT_URL_SAGEMAKER_EDGE
SageMaker FeatureSt ore Runtime	sa _f to ir	AWS_ENDPOINT_URL_SAGEMAKER_FEATUREST ORE_RUNTIME
SageMaker Geospatial	sa _g a1	AWS_ENDPOINT_URL_SAGEMAKER_GEOSPATIAL
SageMaker Metrics	sa _m	AWS_ENDPOINT_URL_SAGEMAKER_METRICS
SageMaker Runtime	sa _r	AWS_ENDPOINT_URL_SAGEMAKER_RUNTIME
savingsplans	sa ar	AWS_ENDPOINT_URL_SAVINGSPLANS
Scheduler	sc	AWS_ENDPOINT_URL_SCHEDULER
schemas	sc	AWS_ENDPOINT_URL_SCHEMAS

serviceId	Ku pe la ur fil be AV **co**	Variabel lingkungan **AWS_ENDPOINT_URL_<SERVICE>**
SimpleDB	s:	AWS_ENDPOINT_URL_SIMPLEDB
Secrets Manager	se ar	AWS_ENDPOINT_URL_SECRETS_MANAGER
SecurityHub	se ub	AWS_ENDPOINT_URL_SECURITYHUB
SecurityLake	se al	AWS_ENDPOINT_URL_SECURITYLAKE
ServerlessApplicationRepository	se sa io to	AWS_ENDPOINT_URL_SERVERLESSAPPLICATIONREPOSITORY
Service Quotas	se uo	AWS_ENDPOINT_URL_SERVICE_QUOTAS
Service Catalog	se at	AWS_ENDPOINT_URL_SERVICE_CATALOG
Service Catalog AppRegistry	se at pi	AWS_ENDPOINT_URL_SERVICE_CATALOG_APPREGISTRY
ServiceDiscovery	se so	AWS_ENDPOINT_URL_SERVICEDISCOVERY

serviceId	Ku pe la ur fil be AV **co**	Variabel lingkungan **AWS_ENDPOINT_URL_<SERVICE>**
SES	se	AWS_ENDPOINT_URL_SES
SESv2	se	AWS_ENDPOINT_URL_SESV2
Shield	sh	AWS_ENDPOINT_URL_SHIELD
signer	si	AWS_ENDPOINT_URL_SIGNER
SimSpaceWeaver	si ea	AWS_ENDPOINT_URL_SIMSPACEWEAVER
SMS	sn	AWS_ENDPOINT_URL_SMS
Snow Device Management	sn ce me	AWS_ENDPOINT_URL_SNOW_DEVICE_MANAGEMENT
Snowball	sn	AWS_ENDPOINT_URL_SNOWBALL
SNS	sn	AWS_ENDPOINT_URL_SNS
SQS	sq	AWS_ENDPOINT_URL_SQS
SSM	ss	AWS_ENDPOINT_URL_SSM
SSM Contacts	ss ct	AWS_ENDPOINT_URL_SSM_CONTACTS
SSM Incidents	ss en	AWS_ENDPOINT_URL_SSM_INCIDENTS

serviceId	Ku pe la ur fil be AV **co**	Variabel lingkungan **AWS_ENDPOINT_URL_<SERVICE>**
Ssm Sap	ss	AWS_ENDPOINT_URL_SSM_SAP
SSO	ss	AWS_ENDPOINT_URL_SSO
SSO Admin	ss	AWS_ENDPOINT_URL_SSO_ADMIN
SSO OIDC	ss	AWS_ENDPOINT_URL_SSO_OIDC
SFN	st	AWS_ENDPOINT_URL_SFN
Storage Gateway	st at	AWS_ENDPOINT_URL_STORAGE_GATEWAY
STS	st	AWS_ENDPOINT_URL_STS
SupplyChain	su in	AWS_ENDPOINT_URL_SUPPLYCHAIN
Support	su	AWS_ENDPOINT_URL_SUPPORT
Support App	su pp	AWS_ENDPOINT_URL_SUPPORT_APP
SWF	sw	AWS_ENDPOINT_URL_SWF
synthetics	sy s	AWS_ENDPOINT_URL_SYNTHETICS
Textract	te	AWS_ENDPOINT_URL_TEXTRACT

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

serviceId	Ku pe la ur fil be AV **co**	Variabel lingkungan **AWS_ENDPOINT_URL_<SERVICE>**
VPC Lattice	vp ce	AWS_ENDPOINT_URL_VPC_LATTICE
WAF	wa	AWS_ENDPOINT_URL_WAF
WAF Regional	wa na	AWS_ENDPOINT_URL_WAF_REGIONAL
WAFV2	wa	AWS_ENDPOINT_URL_WAFV2
WellArchitected	we te	AWS_ENDPOINT_URL_WELLARCHITECTED
Wisdom	wi	AWS_ENDPOINT_URL_WISDOM
WorkDocs	wo	AWS_ENDPOINT_URL_WORKDOCS
WorkLink	wo	AWS_ENDPOINT_URL_WORKLINK
WorkMail	wo	AWS_ENDPOINT_URL_WORKMAIL
WorkMailMessageFlow	wo es w	AWS_ENDPOINT_URL_WORKMAILMESSAGEFLOW
WorkSpaces	wo s	AWS_ENDPOINT_URL_WORKSPACES

serviceId	Ku pe la ur fil be AV **co**	Variabel lingkungan **AWS_ENDPOINT_URL_<SERVICE>**
WorkSpaces Thin Client	wo s_ ie	AWS_ENDPOINT_URL_WORKSPACES_THIN_CLIENT
WorkSpaces Web	wo s_	AWS_ENDPOINT_URL_WORKSPACES_WEB
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, lihatMemahami halaman pengaturan panduan ini.

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

Konfigurasikan fungsi ini dengan menggunakan yang berikut ini:

`defaults_mode`- Pengaturan AWS **`config`** file bersama, **`AWS_DEFAULTS_MODE`**- variabel lingkungan, **`aws.defaultsMode`**- Properti sistem JVM: hanya Java/Kotlin

Dengan pengaturan ini, Anda dapat memilih mode yang selaras dengan arsitektur aplikasi Anda, yang kemudian memberikan nilai default yang dioptimalkan untuk aplikasi Anda. Jika setelan AWS SDK memiliki nilai yang ditetapkan secara eksplisit, maka nilai itu selalu diutamakan. Jika

setelan AWS SDK tidak memiliki nilai yang ditetapkan secara eksplisit, dan `defaults_mode` tidak sama dengan 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 pemilihan`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
`retryMode`	`standard`	`standard`	`standard`	`standard`
`stsRegionalEndpoints`	`regional`	`regional`	`regional`	`regional`
`s3UsEast1RegionalEndpoints`	`regional`	`regional`	`regional`	`regional`
`connectTimeoutInMillis`	3100	1100	3100	30000

Parameter	**standard**modus	**in-region** modus	**cross-region** modus	**mobile**modus
`tlsNegotiationTimeoutInMillis`	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:`stsRegionalEndpoints` ,`s3UsEast1RegionalEndpoints` ,`tlsNegotiationTimeoutInMillis` .
SDK for Go V2 (1.x)	Ya	Parameter tidak dioptimalkan:`retryMode` ,`stsRegionalEndpoints` ,`s3UsEast1RegionalEndpoints` .
SDK for Go 1.x (V1)	Tidak	

SDK	Didukung	Catatan atau informasi lebih lanjut
SDK for Java 2.x	Ya	Parameter tidak dioptimal kan:`stsRegionalEndpoints` .
SDK for Java 1.x	Tidak	
SDK untuk 3.x JavaScript	Ya	Parameter tidak dioptimal kan:`stsRegionalEndpoints` ,`s3UsEast1RegionalEndpoints` ,`tlsNegotiationTimeoutInMillis` . `connectTimeoutInMillis` disebut`connectionTimeout` .
SDK untuk 2.x JavaScript	Tidak	
SDK para Kotlin	Tidak	
SDK for .NET 4.x	Ya	Parameter tidak dioptimal kan:`connectTimeoutInMillis` ,`tlsNegotiationTimeoutInMillis` .
SDK for .NET 3.x	Ya	Parameter tidak dioptimal kan:`connectTimeoutInMillis` ,`tlsNegotiationTimeoutInMillis` .
SDK for PHP 3.x	Ya	Parameter tidak dioptimal kan:`tlsNegotiationTimeoutInMillis` .

SDK	Didukung	Catatan atau informasi lebih lanjut
SDK untuk Python (Boto3)	Ya	Parameter tidak dioptimal kan:`tlsNegotiationTimeoutInMillis` .
SDK for Ruby 3.x	Ya	
SDK untuk Rust	Tidak	
SDK para Swift	Tidak	
Alat untuk PowerShell V5	Ya	Parameter tidak dioptimal kan:`connectTimeoutInMillis` ,`tlsNegotiationTimeoutInMillis` .
Alat untuk PowerShell V4	Ya	Parameter tidak dioptimal kan:`connectTimeoutInMillis` ,`tlsNegotiationTimeoutInMillis` .

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 dibagikan di semua SDKs menyediakan penggunaan kembali kode, pengoptimalan, dan akurasi yang lebih baik. Paket-paketnya adalah:

- `awslabs/aws-c-auth`: otentikasi AWS sisi klien (penyedia kredensyal standar dan penandatanganan (sigv4))
- `awslabs/aws-c-cal`: Jenis primitif kriptografi, hash (,, SHA256 HMAC) MD5 SHA256, penandatangan, AES
- `awslabs/aws-c-common`: Struktur data dasar, tipe primitif utas/sinkronisasi, manajemen buffer, fungsi terkait stdlib
- `awslabs/aws-c-compression`: Algoritma kompresi (pengkodean/decoding Huffman)
- `awslabs/aws-c-event-stream`: Pemrosesan pesan aliran acara (header, pendahuluan, payload, crc/trailer), implementasi panggilan prosedur jarak jauh (RPC) melalui aliran acara
- `awslabs/aws-c-http`: Implementasi C99 dari spesifikasi HTTP/1.1 dan HTTP/2
- `awslabs/aws-c-io`: Soket (TCP, UDP), DNS, pipa, loop acara, saluran, SSL/TLS
- `awslabs/aws-c-iot`: Implementasi C99 integrasi layanan cloud AWS IoT dengan perangkat
- `awslabs/aws-c-mqtt`: Protokol pesan standar dan ringan untuk Internet of Things (IoT)
- `awslabs/aws-c-s3`: Implementasi 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 formuliraws-crt-*, di mana '*' adalah bahasa SDK (seperti aws-crt-cppatau aws-crt-java).

Berikut ini adalah ilustrasi dependensi hierarkis dari pustaka CRT.

AWS SDKs dan kebijakan pemeliharaan alat

Gambaran Umum

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.

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	Ketersediaan Umum	9/2/2013	
AWS CLI	2.x	Ketersediaan Umum	2/10/2020	
SDK for C++	1.x	Ketersediaan Umum	9/2/2015	
SDK for Go V2	V2 1.x	Ketersediaan Umum	1/19/2021	
SDK for Go	1.x	Akhir dukungan	11/19/2015	
SDK for Java	1.x	Maintenance	25/03/2010	Lihat pengumuman untuk detail dan tanggal
SDK for Java	2.x	Ketersediaan Umum	11/20/2018	
SDK untuk JavaScript	1.x	Akhir dukungan	5/6/2013	
SDK 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 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	30 Mei 2024

	JVM yang didukung oleh 1.x. AWS SDK untuk Java	
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 kredenal 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 mencermin kan 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.