Guia de referência

AWS SDKse ferramentas

AWS SDKse ferramentas: Guia de referência

Table of Contents

AWS SDKsGuia de referência de ferramentas e ferramentas

Muitas ferramentas SDKs e ferramentas compartilham algumas funcionalidades comuns, seja por meio de especificações de design compartilhadas ou por meio de uma biblioteca compartilhada.

Este guia inclui informações sobre:

- Configuração— Como usar os `credentials` arquivos `config` compartilhados ou variáveis de ambiente para configurar suas AWS SDKs ferramentas.
- Autenticação e acesso— Estabeleça como seu código ou ferramenta se autentica AWS quando você desenvolve com Serviços da AWS.
- Referência de configurações: referência para todas as configurações padronizadas disponíveis para autenticação e configuração.
- AWS Bibliotecas Common Runtime (CRT)— Visão geral das bibliotecas compartilhadas do AWS Common Runtime (CRT) que estão disponíveis para quase todosSDKs.
- AWS Política de manutenção de SDKs e ferramentasabrange a política de manutenção e o controle de versões de kits de desenvolvimento de AWS software (SDKs) e ferramentas, incluindo dispositivos móveis e Internet das Coisas (IoT)SDKs, e suas dependências subjacentes.

Este Guia de Referência AWS SDKs e de Ferramentas tem como objetivo ser uma base de informações aplicável a várias SDKs ferramentas. O guia específico da ferramenta SDK ou que você está usando deve ser usado além de qualquer informação apresentada aqui. A seguir estão SDK as ferramentas que contêm seções relevantes do material neste guia:

Se você estiver usando:	As seções relevantes deste guia para você são:
• Qualquer SDK ferramenta	AWS Política de manutenção de SDKs e ferramentas
• AWS Cloud9 • AWS CDK • AWS Toolkit for Azure DevOps • AWS Toolkit for JetBrains	Configuração Autenticação e acesso AWS Política de manutenção de SDKs e ferramentas

Se você estiver usando:	As seções relevantes deste guia para você são:
• 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 for C++ • AWS SDK for Go • AWS SDK for Java • AWS SDK for JavaScript • AWS SDK para Kotlin • AWS SDK for .NET • AWS SDK for PHP • AWS SDK for Python (Boto3) • AWS SDK for Ruby • AWS SDK para Rust • AWS SDK for Swift • AWS Tools for Windows PowerShell	Configuração Autenticação e acesso Referência de configurações AWS Bibliotecas Common Runtime (CRT) AWS Política de manutenção de SDKs e ferramentas AWS SDKse suporte à versão Tools

Recursos para desenvolvedores

Para obter uma visão geral das ferramentas que podem ajudá-lo a desenvolver aplicativos AWS, consulte Ferramentas para desenvolver AWS. Para obter informações sobre suporte, consulte o Centro de Conhecimentos da AWS.

O Amazon Q Developer é um assistente conversacional generativo baseado em IA que pode ajudar você a entender, criar, estender e operar aplicativos. AWS Para acelerar sua construção AWS, o modelo que impulsiona o Amazon Q é aprimorado com AWS conteúdo de alta qualidade para produzir respostas mais completas, acionáveis e referenciadas. Para obter mais informações, consulte What is Amazon Q Developer no Guia do usuário do Amazon Q Developer.

Notificação de telemetria do kit de ferramentas

AWS Os kits de ferramentas do Integrated Development Environment (IDE) são plug-ins e extensões que permitem o acesso aos AWS serviços em seuIDE. IDEOs plug-ins e extensões do Amazon Q permitem assistência generativa de IA em seuIDE. Para obter informações detalhadas sobre cada um dos IDE kits de ferramentas, consulte os Guias do usuário do kit de ferramentas na tabela anterior. Para saber mais sobre como usar o Amazon Q em seuIDE, consulte Como usar o Amazon Q no IDE tópico do guia do desenvolvedor do Amazon Q.

AWS IDEOs kits de ferramentas e o Amazon Q podem coletar e armazenar dados de telemetria do lado do cliente para informar decisões sobre futuros lançamentos do Toolkit AWS e do Amazon Q. Os dados coletados quantificam seu uso do AWS Toolkit e do Amazon Q.

Para saber mais sobre os dados de telemetria coletados em todos os kits de AWS IDE ferramentas e no Amazon Q, consulte o documento commonDefinitions.json no repositório Github. `aws-toolkit-common`

Para obter informações detalhadas sobre os dados de telemetria coletados por cada um dos AWS IDE Toolkits e extensões do Amazon Q, consulte os documentos de recursos nos seguintes AWS repositórios do Toolkit: GitHub

- AWS Kit de ferramentas do Visual Studio com Amazon Q
- AWS Toolkit for Visual Studio Code e extensão Amazon Q para VS Code
- AWS Toolkit for JetBrains e o plug-in Amazon Q para JetBrains
- Amazon Q para Eclipse

Certos AWS serviços acessíveis nos AWS kits de ferramentas podem coletar dados adicionais de telemetria do lado do cliente. Para obter informações detalhadas sobre o tipo de dados coletados por cada AWS serviço individual, consulte o tópico de AWS documentação do serviço específico em que você está interessado.

Configuração

Com AWS SDKs e outras ferramentas para AWS desenvolvedores, como o AWS Command Line Interface (AWS CLI), você pode interagir com as APIs AWS de serviço. Antes de tentar isso, no entanto, você deve configurar o SDK ou a ferramenta com as informações necessárias para realizar a operação solicitada.

Essas informações incluem os seguintes itens:

- Informações de credenciais que identificam quem está chamando a API. As credenciais são usadas para criptografar a solicitação para os AWS servidores. Usando essas informações, AWS confirma sua identidade e pode recuperar as políticas de permissões associadas a ela. Em seguida, ele pode determinar quais ações você tem permissão para realizar.
- Outros detalhes de configuração que você usa para informar ao SDK AWS CLI ou ao SDK como processar a solicitação, para onde enviar a solicitação (para qual endpoint de AWS serviço) e como interpretar ou exibir a resposta.

Cada SDK ou ferramenta oferece suporte a várias fontes que você pode usar para fornecer as informações de credenciais e de configuração necessárias. Algumas fontes são exclusivas do SDK ou da ferramenta, e você deve consultar a documentação dessa ferramenta ou do SDK para obter detalhes sobre como usar esse método.

No entanto, a maioria dos AWS SDKs e ferramentas oferece suporte a configurações comuns de duas fontes principais (além do código em si):

- Arquivos de AWS configuração e credenciais compartilhados — Os arquivos compartilhados `config` e `credentials` os arquivos são a forma mais comum de especificar a autenticação e a configuração em um AWS SDK ou ferramenta. Use esses arquivos para armazenar as configurações que suas ferramentas e aplicativos podem usar. As configurações nos arquivos `config` e `credentials` compartilhados estão associadas a um perfil específico. Com vários perfis, você pode criar configurações diferentes para aplicar em diferentes cenários. Ao usar uma AWS ferramenta para invocar um comando ou usar um SDK para invocar uma AWS API, você pode especificar qual perfil e, portanto, quais definições de configuração usar para essa ação. Um dos perfis é designado como o perfil `default` e é usado automaticamente quando você não especifica explicitamente um perfil a ser usado. As configurações que você pode armazenar nesses arquivos estão documentadas neste guia de referência.

- Variáveis de ambiente — Algumas das configurações podem ser armazenadas alternativamente nas variáveis de ambiente do seu sistema operacional. Embora você possa ter somente um conjunto de variáveis de ambiente em vigor por vez, elas são facilmente modificadas dinamicamente à medida que seu programa é executado e seus requisitos mudam.

Tópicos adicionais nesta seção

- Arquivos config e credentials compartilhados
- Localização do compartilhado de arquivos config e credentials compartilhados
- Suporte a variáveis de ambiente
- Suporte às propriedades do sistema JVM

Arquivos **`config`** e **`credentials`** compartilhados

Os `credentials` arquivos compartilhados AWS `config` e contêm um conjunto de perfis. Um perfil é um conjunto de definições de configuração, em pares chave-valor, que é usado pelas ferramentas AWS Command Line Interface (AWS CLI) AWS SDKs, the e outras. Os valores de configuração são anexados a um perfil para configurar algum aspecto da SDK ferramenta /quando esse perfil é usado. Esses arquivos são “compartilhados”, pois os valores entram em vigor em qualquer aplicativo, processo ou SDKs no ambiente local de um usuário.

Tanto os arquivos compartilhados `config` quanto `credentials` os arquivos são arquivos de texto simples que contêm somente ASCII caracteres (codificados em UTF -8). Eles assumem a forma do que geralmente é chamado de INIarquivos.

Perfis

As configurações nos arquivos `config` e `credentials` compartilhados estão associadas a um perfil específico. Vários perfis podem ser definidos no arquivo para criar configurações de configuração diferentes para serem aplicadas em diferentes ambientes de desenvolvimento.

O `[default]` perfil contém os valores que são usados por uma operação de ferramenta SDK ou se um perfil nomeado específico não for especificado. Você também pode criar perfis separados aos quais você pode referenciar explicitamente pelo nome. Cada perfil pode usar configurações e valores diferentes conforme necessário para seu aplicativo e cenário.

> **Note**
>
> `[default]` é simplesmente um perfil sem nome. Esse perfil é nomeado `default` porque é o perfil padrão usado pelo SDK se o usuário não especificar um perfil. Ele não fornece valores padrão herdados para outros perfis. Se você definir algo no `[default]` perfil e não o definir em um perfil nomeado, o valor não será definido quando você usar o perfil nomeado.

Definir um perfil nomeado

O `[default]` perfil e vários perfis nomeados podem existir no mesmo arquivo. Use a configuração a seguir para selecionar quais configurações do perfil serão usadas por você SDK ou pela ferramenta ao executar seu código. Os perfis também podem ser selecionados dentro do código ou por comando ao trabalhar com o. AWS CLI

Configure essa funcionalidade definindo uma das seguintes opções:

`AWS_PROFILE`- variável de ambiente

Quando essa variável de ambiente é definida como um perfil nomeado ou “padrão”, todos os SDK códigos e AWS CLI comandos usam as configurações desse perfil.

Exemplo de configuração de variáveis de ambiente para Linux/macOS por meio da linha de comando:

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

Exemplo do Windows de configuração de variáveis de ambiente por meio da linha de comando:

```
setx AWS_PROFILE "my_default_profile_name"
```

`aws.profile`- propriedade JVM do sistema

SDKPara Kotlin no JVM e SDK para Java 2.x, você pode definir a propriedade do `aws.profile` sistema. Quando SDK cria um cliente de serviço, ele usa as configurações no perfil nomeado, a menos que a configuração seja substituída no código. O SDK for Java 1.x não suporta essa propriedade do sistema.

> **Note**
>
> Se seu aplicativo estiver em um servidor executando vários aplicativos, recomendamos que você sempre use perfis nomeados em vez do perfil padrão. O perfil padrão é automaticamente selecionado por qualquer AWS aplicativo no ambiente e compartilhado entre eles. Portanto, se outra pessoa atualizar o perfil padrão de seu aplicativo, isso poderá impactar involuntariamente os outros. Para se proteger contra isso, defina um perfil nomeado no `config` arquivo compartilhado e, em seguida, use esse perfil nomeado em seu aplicativo definindo o perfil nomeado em seu código. Você pode usar a variável de ambiente ou a propriedade do JVM sistema para definir o perfil nomeado se souber que seu escopo afeta apenas seu aplicativo.

Formato do arquivo de configuração

O arquivo `config` é organizado em seções. Uma seção é um conjunto nomeado de configurações e continua até que outra linha de definição de seção seja encontrada.

O arquivo `config` é um arquivo de texto simples que usam o seguinte formato:

- Todas as entradas em uma seção assumem a forma geral de `setting-name=value`.
- As linhas podem ser comentadas iniciando-as com um caractere de hashtag (#).

Tipos de seção

Uma definição de seção é uma linha que aplica um nome a uma coleção de configurações. As linhas de definição de seção começam e terminam com colchetes (`[ ]`). Dentro dos colchetes, há um identificador de tipo de seção e um nome personalizado para a seção. Você pode usar letras, números, hífens (`-`) e sublinhados (`_`), mas sem espaços.

Tipo de seção: **`default`**

Exemplo de linha de definição de seção: `[default]`

`[default]`é o único perfil que não exige o identificador da `profile` seção.

O exemplo a seguir mostra um arquivo `config` básico com um perfil `[default]`. Ele define a configuração `region`. Todas as configurações que seguem essa linha, até que outra definição de seção seja encontrada, fazem parte desse perfil.

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

Tipo de seção: **`profile`**

Exemplo de linha de definição de seção: `[profile dev]`

A linha de definição da `profile` seção é um agrupamento de configuração nomeado que você pode aplicar a diferentes cenários de desenvolvimento. Para entender melhor os perfis nomeados, consulte a seção anterior sobre Perfis.

O exemplo a seguir mostra um `config` arquivo com uma linha de definição de `profile` seção e um perfil nomeado chamado`foo`. Todas as configurações que seguem essa linha, até que outra definição de seção seja encontrada, fazem parte desse perfil nomeado.

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

Algumas configurações têm seu próprio grupo aninhado de subconfigurações, como a configuração e as subconfigurações de `s3` no exemplo a seguir. Associe as subconfigurações ao grupo recuando-as com um ou mais espaços.

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

Tipo de seção: **`sso-session`**

Exemplo de linha de definição de seção: `[sso-session my-sso]`

A linha de definição da `sso-session` seção nomeia um grupo de configurações que você usa para configurar um perfil para resolver AWS as credenciais usando AWS IAM Identity Center. Para obter mais informações sobre como configurar a autenticação de login único, consulte IAMAutenticação do Identity Center para sua ferramenta SDK ou. Um perfil é vinculado a uma seção `sso-session` por um par de valores-chave em que `sso-session` é a chave e o nome da sua seção `sso-session` é o valor, como `sso-session = <name-of-sso-session-section>`.

O exemplo a seguir configura um perfil que obterá AWS credenciais de curto prazo para a IAM função "SampleRole" na conta “111122223333" usando um token do “my-sso”. A seção `sso-session` “my-sso” é referenciada na seção `profile` pelo nome usando a chave `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
```

Tipo de seção: **`services`**

Exemplo de linha de definição de seção: `[services dev]`

> **Note**
>
> A `services` seção oferece suporte a personalizações de endpoints específicos do serviço e está disponível somente nas ferramentas que incluem esse SDKs recurso. Para ver se esse recurso está disponível para vocêSDK, consulte os Compatibilidade com AWS SDKs endpoints específicos do serviço.

A linha de definição da `services` seção nomeia um grupo de configurações que configura endpoints personalizados para AWS service (Serviço da AWS) solicitações. Um perfil é vinculado a uma seção `services` por um par de valores-chave em que `services` é a chave e o nome da sua seção `services` é o valor, como `services = <name-of-services-section>`.

A `services` seção é ainda separada em subseções por `<SERVICE> =` linhas, onde `<SERVICE>` está a chave AWS service (Serviço da AWS) identificadora. O AWS service (Serviço da AWS) identificador é baseado no API modelo, substituindo todos os espaços `serviceId` por sublinhados e colocando todas as letras em minúsculas. Para obter uma lista de todas as chaves de identificação de serviço a serem usadas na seção `services`, consulte Identificadores para endpoints específicos de serviço. A chave de identificação de serviço é seguida por configurações aninhadas, cada uma em sua própria linha e recuada por dois espaços.

O exemplo a seguir usa uma definição `services` para configurar o endpoint a ser usado para solicitações feitas somente para o serviço do Amazon DynamoDB . A seção `services` `"local-`

dynamodb" é referenciada na seção `profile` pelo nome usando a chave `services`. A chave AWS service (Serviço da AWS) identificadora é`dynamodb`. A subseção de Amazon DynamoDB serviço começa na linha`dynamodb = `. Todas as linhas imediatamente seguintes que estejam recuadas são incluídas nessa subseção e se aplicam a esse serviço.

```
[profile dev]
services = local-dynamodb

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

Para obter mais informações sobre a configuração de endpoint personalizado, consulte Endpoints específicos de serviço.

Formato do arquivo de credenciais

As regras para o arquivo `credentials` geralmente são idênticas às do arquivo `config`, exceto que as seções do perfil não começam com a palavra `profile`. Use somente o nome do perfil em si entre colchetes. O exemplo a seguir mostra um `credentials` arquivo com uma seção de perfil nomeada chamada`foo`.

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

Somente as seguintes configurações consideradas “secretas” ou confidenciais podem ser armazenadas no `credentials` arquivo: `aws_access_key_id`, `aws_secret_access_key`, `aws_session_token` e. Embora essas configurações possam ser colocadas alternativamente no `config` arquivo compartilhado, recomendamos que você mantenha esses valores confidenciais em um `credentials` arquivo separado. Dessa forma, você pode fornecer permissões separadas para cada arquivo, se necessário.

O exemplo a seguir mostra um arquivo `credentials` básico com um perfil `[default]`. Ele define as configurações `aws_access_key_id`, `aws_secret_access_key`, e `aws_session_token` globais.

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

```
aws_session_token=IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZVERYL0NGSTR1
```

Independentemente de você usar um perfil nomeado ou "`default`" em seu `credentials` arquivo, todas as configurações aqui serão combinadas com quaisquer configurações do seu `config` arquivo que usem o mesmo nome de perfil. Se houver credenciais nos dois arquivos para um perfil que compartilhe o mesmo nome, as chaves no arquivo de credenciais terão precedência.

Localização do compartilhado de arquivos **`config`** e **`credentials`** compartilhados

Os arquivos compartilhados AWS `config` e são `credentials` arquivos de texto simples que contêm informações de configuração para os AWS SDKs e as ferramentas. Os arquivos residem localmente em seu ambiente e são usados automaticamente pelo código do SDK ou pelos AWS CLI comandos que você executa nesse ambiente. Por exemplo, em seu próprio computador ou ao desenvolver em uma instância do Amazon Elastic Compute Cloud.

Quando o SDK ou a ferramenta são executados, eles verificam esses arquivos e carregam todas as configurações disponíveis. Se os arquivos ainda não existirem, um arquivo básico será criado automaticamente pelo SDK ou pela ferramenta.

Por padrão, os arquivos estão em uma pasta chamada `.aws` que é colocada na sua pasta home ou na pasta do usuário.

Sistema operacional	Local padrão e nome dos arquivos

Linux e macOS	`~/.aws/config`
	
	`~/.aws/credentials`

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

Resolução do diretório inicial

~só é usado para resolução de diretórios pessoais quando:

- Inicia o caminho

- É seguido imediatamente por `/` ou por um separador específico da plataforma. No Windows, `~/` e `~\` ambos são resolvidos para o diretório inicial.

Ao determinar o diretório inicial, as seguintes variáveis são verificadas:

- (Todas as plataformas) A variável de ambiente `HOME`
- (Plataformas Windows) A variável de ambiente `USERPROFILE`
- (Plataformas Windows) A concatenação de variáveis de `HOMEDRIVE` `HOMEPATH` ambiente () `$HOMEDRIVE$HOMEPATH`
- (Opcional por SDK ou ferramenta) Um SDK ou função de resolução de caminho inicial específica do SDK ou da ferramenta

Quando possível, se o diretório inicial de um usuário for especificado no início do caminho (por exemplo, `~username/`), ele será resolvido no diretório inicial do nome de usuário solicitado (por exemplo, `/home/username/.aws/config`).

Alterar a localização padrão desses arquivos

Você pode usar qualquer uma das opções a seguir para substituir de onde esses arquivos são carregados pelo SDK ou pela ferramenta.

Use variáveis de ambiente

As seguintes variáveis de ambiente podem ser definidas para alterar a localização ou o nome desses arquivos do valor padrão para um valor personalizado:

- Arquivo de variável de ambiente `config`: **`AWS_CONFIG_FILE`**
- Arquivo de variável de ambiente `credentials`: **`AWS_SHARED_CREDENTIALS_FILE`**

Linux/macOS

Você pode especificar um local alternativo executando os seguintes comandos de exportação no Linux ou no 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

Você pode especificar um local alternativo executando os seguintes comandos setx no 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
```

Para obter mais informações sobre como configurar seu sistema usando variáveis de ambiente, consulteSuporte a variáveis de ambiente.

Use as propriedades do sistema JVM

Para o SDK para Kotlin executado na JVM e para o SDK for Java 2.x, você pode definir as seguintes propriedades do sistema JVM para alterar a localização ou o nome desses arquivos do valor padrão para um valor personalizado:

- `config`propriedade do sistema JVM do arquivo: **`aws.configFile`**
- Arquivo de variável de ambiente `credentials`: **`aws.sharedCredentialsFile`**

Para obter instruções sobre como definir as propriedades do sistema JVM, consulte. the section called “Como definir as propriedades do sistema JVM” O SDK for Java 1.x não oferece suporte a essas propriedades do sistema.

Suporte a variáveis de ambiente

Variáveis de ambiente fornecem outra maneira de especificar opções de configuração e credenciais e podem ser úteis para criação de scripts ou configuração temporária de um perfil nomeado como o padrão. Para obter a lista das variáveis de ambiente suportadas pela maioriaSDKs, consulteLista de variáveis de ambiente.

Precedência de opções

- Se você especificar uma configuração usando sua variável de ambiente, ela substituirá qualquer valor carregado de um perfil nos arquivos compartilhados AWS `config` `ecredentials`.
- Se você especificar uma configuração usando um parâmetro na linha de AWS CLI comando, ela substituirá qualquer valor da variável de ambiente correspondente ou de um perfil no arquivo de configuração.

Como definir variáveis de ambiente

Os exemplos a seguir mostram como configurar variáveis de ambiente para o usuário padrão.

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
```

Configurar a variável de ambiente altera o valor usado até o final da sua sessão de shell ou até que você defina a variável como um valor diferente. Você pode tornar as variáveis persistentes em sessões futuras definindo-as no script de inicialização do shell.

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
```

O uso de `set` para definir uma variável de ambiente altera o valor usado até o final da atual sessão de prompt de comando ou até que você defina a variável como um valor diferente. O uso de `setx` para definir uma variável de ambiente altera o valor usado na sessão atual de prompt de comando e todas as sessões de prompt de comando que você criar após a execução do comando. Não afeta outros shells de comando que já estejam em execução no momento em que você executar o comando.

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"
```

Se você definir uma variável de ambiente no PowerShell prompt, conforme mostrado nos exemplos anteriores, ela salvará o valor somente durante a sessão atual. Para tornar a

configuração da variável de ambiente persistente em todas as sessões PowerShell e nas sessões do Prompt de Comando, armazene-a usando o aplicativo Sistema no Painel de Controle. Como alternativa, você pode definir a variável para todas as PowerShell sessões futuras adicionando-a ao seu PowerShell perfil. Consulte a PowerShell documentação para obter mais informações sobre como armazenar variáveis de ambiente ou persisti-las nas sessões.

Configuração de variável de ambiente sem servidor

Se você usa uma arquitetura sem servidor para desenvolvimento, você tem outras opções para definir variáveis de ambiente. Dependendo do seu contêiner, você pode usar estratégias diferentes de execução de código nesses contêineres para ver e acessar as variáveis de ambiente, semelhantes a ambientes fora da nuvem.

Por exemplo, com AWS Lambda, você pode definir diretamente as variáveis de ambiente. Para obter detalhes, consulte Usando variáveis de AWS Lambda ambiente no Guia do AWS Lambda desenvolvedor.

No Serverless Framework, muitas vezes você pode definir variáveis de SDK ambiente no `serverless.yml` arquivo sob a chave do provedor sob a configuração do ambiente. Para obter informações sobre o arquivo `serverless.yml`, consulte Configurações gerais da função na documentação do Serverless Framework.

Independentemente do mecanismo usado para definir variáveis de ambiente de contêiner, há algumas que são reservadas pelo contêiner, como aquelas documentadas para Lambda em Defined runtime environment variables. Sempre consulte a documentação oficial do contêiner que você está usando para determinar como as variáveis de ambiente são tratadas e se há alguma restrição.

Suporte às propriedades do sistema JVM

As propriedades do sistema JVM fornecem outra maneira de especificar opções de configuração e credenciais para SDKs executados na JVM, como o e o. AWS SDK for Java AWS SDK para KotlinPara obter uma lista das propriedades do sistema JVM suportadas pelos SDKs, consulte Referência de configurações.

Precedência de opções

- Se você especificar uma configuração usando sua propriedade de sistema JVM, ela substituirá qualquer valor encontrado nas variáveis de ambiente ou carregado de um perfil na AWS e nos arquivos compartilhados. `config` `credentials`

- Se você especificar uma configuração usando sua variável de ambiente, ela substituirá qualquer valor carregado de um perfil na AWS `config` e `credentials` nos arquivos compartilhados.

Como definir as propriedades do sistema JVM

Você pode definir as propriedades do sistema JVM de várias maneiras.

Na linha de comando

Defina as propriedades do sistema JVM na linha de comando ao invocar o `java` comando usando o switch. `-D` O comando a seguir configura Região da AWS globalmente para todos os clientes de serviço, a menos que você substitua explicitamente o valor no código.

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

Se você precisar definir várias propriedades do sistema JVM, especifique o `-D` switch várias vezes.

Com uma variável de ambiente

Se você não conseguir acessar a linha de comando para invocar a JVM para executar seu aplicativo, poderá usar a variável de `JAVA_TOOL_OPTIONS` ambiente para configurar as opções da linha de comando. Essa abordagem é útil em situações como executar uma AWS Lambda função no Java Runtime ou executar código em uma JVM incorporada.

O exemplo a seguir configura Região da AWS globalmente para todos os clientes de serviço, a menos que você substitua explicitamente o valor no código.

Linux, macOS, or Unix

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

Configurar a variável de ambiente altera o valor usado até o final da sua sessão de shell ou até que você defina a variável como um valor diferente. Você pode tornar as variáveis persistentes em sessões futuras definindo-as no script de inicialização do shell.

Windows Command Prompt

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

O uso de `set` para definir uma variável de ambiente altera o valor usado até o final da atual sessão de prompt de comando ou até que você defina a variável como um valor diferente. O uso de `setx` para definir uma variável de ambiente altera o valor usado na sessão atual de prompt de comando e todas as sessões de prompt de comando que você criar após a execução do comando. Não afeta outros shells de comando que já estejam em execução no momento em que você executar o comando.

Em tempo de execução

Você também pode definir as propriedades do sistema JVM em tempo de execução no código usando o `System.setProperty` método, conforme mostrado no exemplo a seguir.

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

Important

Defina todas as propriedades do sistema JVM antes de inicializar os clientes de serviço do SDK, caso contrário, os clientes de serviço poderão usar outros valores.

Autenticação e acesso

Você deve estabelecer como seu código é autenticado AWS quando você desenvolve com Serviços da AWS. Você pode configurar o acesso programático aos AWS recursos de maneiras diferentes, dependendo do ambiente e do AWS acesso disponível para você.

Opções de autenticação para código executado localmente (não na AWS)

- IAMAutenticação do Identity Center para sua ferramenta SDK ou— Como prática recomendada de segurança, recomendamos o uso AWS Organizations com o IAM Identity Center para gerenciar o acesso em todos os seus Contas da AWS. Você pode criar usuários AWS IAM Identity Center, usar o Microsoft Active Directory, usar um provedor de identidade (IdP) SAML 2.0 ou federar seu IdP individualmente em. Contas da AWS Para verificar se sua região oferece suporte ao IAM Identity Center, consulte AWS IAM Identity Center endpoints e cotas no. Referência geral da Amazon Web Services
- IAM Roles Anywhere— Você pode usar o IAM Roles Anywhere para obter credenciais de segurança temporárias IAM para cargas de trabalho, como servidores, contêineres e aplicativos executados fora do. AWS Para usar o IAM Roles Anywhere, suas cargas de trabalho devem usar certificados X.509.
- Assuma uma função com AWS credenciais— Você pode assumir a IAM função de acessar temporariamente AWS recursos aos quais talvez não tivesse acesso de outra forma.
- AWS chaves de acesso— Outras opções que podem ser menos convenientes ou aumentar o risco de segurança de seus AWS recursos.

Opções de autenticação para código executado em um AWS ambiente

Se seu código for executado AWS, as credenciais poderão ser disponibilizadas automaticamente para seu aplicativo. Por exemplo, se seu aplicativo estiver hospedado no Amazon Elastic Compute Cloud e houver uma IAM função associada a esse recurso, as credenciais serão disponibilizadas automaticamente para seu aplicativo. Da mesma forma, se você usa EKS contêineres da Amazon ECS ou da Amazon, as credenciais definidas para a IAM função podem ser obtidas automaticamente pelo código executado dentro do contêiner por meio SDK da cadeia de fornecedores de credenciais.

- Usando IAM funções para EC2 instâncias da Amazon— Use IAM funções para executar seu aplicativo com segurança em uma instância da AmazonEC2.
- Você pode interagir programaticamente AWS com o IAM Identity Center das seguintes maneiras:

- Use AWS CloudShellpara executar AWS CLI comandos no console.
- Para experimentar o espaço de colaboração baseado em nuvem para equipes de desenvolvimento de software, considere usar a Amazon. CodeCatalyst

Autenticação por meio de um provedor de identidades baseado na Web: aplicativos web baseados em clientes ou móvel

Se você estiver criando aplicativos móveis ou aplicativos web baseados em clientes que exigem acesso AWS, crie seu aplicativo para que ele solicite credenciais de AWS segurança temporárias de forma dinâmica usando a federação de identidades da web.

Com a federação de identidades da web, você não precisa criar código de login personalizado nem gerenciar suas próprias identidades de usuários. Em vez disso, os usuários do aplicativo podem fazer login usando um provedor de identidade externo (IdP) conhecido, como Login com Amazon, Facebook, Google ou qualquer outro IdP compatível com OpenID Connect ()OIDC. Eles podem receber um token de autenticação e, em seguida, trocar esse token por credenciais de segurança temporárias AWS nesse mapa para uma IAM função com permissões para usar os recursos em seu Conta da AWS.

Para saber como configurar isso para sua ferramenta SDK ou ferramenta, consulteAssuma uma função com identidade web ou OpenID Connect.

Para aplicações móveis, recomendamos o uso do Amazon Cognito. O Amazon Cognito atua como um agente de identidades e realiza a maioria do trabalho de federação para você. Para obter mais informações, consulte Como usar o Amazon Cognito para aplicativos móveis no Guia do IAMusuário.

Mais informações sobre gerenciamento de acesso

O Guia IAM do usuário tem as seguintes informações sobre o controle seguro do acesso aos AWS recursos:

- IAMIdentidades (usuários, grupos de usuários e funções) — Entenda os fundamentos das identidades em. AWS
- Melhores práticas de segurança em IAM — Recomendações de segurança a serem seguidas ao desenvolver AWS aplicativos de acordo com o modelo de responsabilidade compartilhada.

A Referência geral da Amazon Web Services tem noções básicas sobre o seguinte:

- Entendendo e obtendo suas AWS credenciais — Acesse as principais opções e práticas de gerenciamento para acesso programático e de console.

ID do builder AWS

Você ID do builder AWS complementa qualquer um Contas da AWS que você já possua ou queira criar. Enquanto um Conta da AWS atua como um contêiner para AWS os recursos que você cria e fornece um limite de segurança para esses recursos, você ID do builder AWS representa você como um indivíduo. Você pode fazer login com você ID do builder AWS para acessar ferramentas e serviços para desenvolvedores, como Amazon CodeWhisperer e Amazon CodeCatalyst.

- Faça login no Guia do Início de Sessão da AWS usuário — Saiba como criar e usar um ID do builder AWS e saiba o que o Builder ID fornece. ID do builder AWS
- Autenticação com CodeWhisperer e AWS Toolkit - Builder ID no Guia CodeWhisperer do Usuário — Saiba como CodeWhisperer usa um ID do builder AWS.
- CodeCatalyst conceitos - ID do builder AWS no Guia CodeCatalyst do usuário da Amazon — Saiba como CodeCatalyst usa um ID do builder AWS.

IAMAutenticação do Identity Center para sua ferramenta SDK ou

AWS IAM Identity Center é o método recomendado de fornecer AWS credenciais ao desenvolver em um serviço não AWS computacional. Por exemplo, isso seria algo como seu ambiente de desenvolvimento local. Se você estiver desenvolvendo em um AWS recurso, como o Amazon Elastic Compute Cloud (AmazonEC2) ou AWS Cloud9, recomendamos obter credenciais desse serviço.

Neste tutorial, você estabelece o acesso ao IAM Identity Center e o configura para sua ferramenta SDK ou usando o portal de AWS acesso e AWS CLI o.

- O portal de AWS acesso é o local da web em que você entra manualmente no IAM Identity Center. O formato do URL é `d-xxxxxxxxxx.awsapps.com/start` ou`your_subdomain`.`awsapps.com/start`. Quando conectado ao portal de AWS acesso, você pode visualizar Contas da AWS as funções que foram configuradas para esse usuário. Esse procedimento usa o portal de AWS acesso para obter os valores de configuração necessários para o processo de autenticação SDK /tool.
- O AWS CLI é usado para configurar sua ferramenta SDK ou para usar a autenticação do IAM Identity Center para API chamadas feitas pelo seu código. Esse processo único atualiza seu

AWS `config` arquivo compartilhado, que é usado por você SDK ou pela ferramenta quando você executa seu código.

Configure o acesso programático usando o IAM Identity Center

Etapa 1: Estabelecer o acesso e selecionar o conjunto de permissões apropriado

Se você ainda não ativou o IAM Identity Center, consulte Habilitando o IAM Identity Center no Guia AWS IAM Identity Center do usuário.

Escolha um dos métodos a seguir para acessar suas AWS credenciais.

Eu não tenho acesso estabelecido por meio do IAM Identity Center

1. Adicione um usuário e adicione permissões administrativas seguindo o procedimento Configurar o acesso do usuário com o diretório padrão do IAM Identity Center no Guia AWS IAM Identity Center do Usuário.
2. O conjunto de `AdministratorAccess` permissões não deve ser usado para desenvolvimento regular. Em vez disso, recomendamos usar o conjunto de `PowerUserAccess` permissões predefinido, a menos que seu empregador tenha criado um conjunto de permissões personalizado para essa finalidade.

   Siga novamente o mesmo procedimento Configurar acesso do usuário com o diretório padrão do IAM Identity Center, mas desta vez:

   - Em vez de criar o *`Admin team`* grupo, crie um *`Dev team`* grupo e substitua-o posteriormente nas instruções.
   - Você pode usar o usuário existente, mas o usuário deve ser adicionado ao novo *`Dev team`* grupo.
   - Em vez de criar o *`AdministratorAccess`* conjunto de *`PowerUserAccess`* permissões, crie um conjunto de permissões e substitua-o posteriormente nas instruções.

   Quando terminar, você deve ter o seguinte:

   - Um `Dev team` grupo.
   - Um conjunto de `PowerUserAccess` permissões anexado ao `Dev team` grupo.
   - Seu usuário foi adicionado ao `Dev team` grupo.

3. Saia do portal e entre novamente para ver suas opções Contas da AWS e para `Administrator` ou`PowerUserAccess`. Selecione `PowerUserAccess` ao trabalhar com sua ferramenta/SDK.

Eu já tenho acesso AWS por meio de um provedor de identidade federado gerenciado pelo meu empregador (como Microsoft Entra ou Okta)

Faça login AWS por meio do portal do seu provedor de identidade. Se o seu administrador de nuvem concedeu permissões a você `PowerUserAccess` (desenvolvedor), você vê o Contas da AWS que você tem acesso e seu conjunto de permissões. Ao lado do nome do seu conjunto de permissões, você vê opções para acessar as contas manual ou programaticamente usando esse conjunto de permissões.

Implementações personalizadas podem resultar em experiências diferentes, como nomes de conjuntos de permissões diferentes. Se não tiver certeza sobre qual conjunto de permissões usar, entre em contato com a equipe de TI para obter ajuda.

Eu já tenho acesso a AWS através do portal de AWS acesso gerenciado pelo meu empregador

Faça login AWS por meio do portal de AWS acesso. Se o seu administrador de nuvem concedeu permissões `PowerUserAccess` (de desenvolvedor) a você, serão exibidas as Contas da AWS às quais você tem acesso e seu conjunto de permissões. Ao lado do nome do seu conjunto de permissões, você vê opções para acessar as contas manual ou programaticamente usando esse conjunto de permissões.

Eu já tenho acesso AWS por meio de um provedor de identidade personalizado federado gerenciado pelo meu empregador

Entre em contato com a equipe de TI para obter ajuda.

Etapa 2: Configuração SDKs e ferramentas para usar o IAM Identity Center

1. Em sua máquina de desenvolvimento, instale a mais recente AWS CLI.

   a. Consulte Instalar ou atualizar a versão mais recente da AWS CLI no Guia do usuário da AWS Command Line Interface .

   b. (Opcional) Para verificar se o AWS CLI está funcionando, abra um prompt de comando e execute o `aws --version` comando.

2. Faça login no portal de AWS acesso. Seu empregador pode fornecer isso URL ou você pode recebê-lo em um e-mail seguindo a Etapa 1: Estabelecer acesso. Caso contrário, encontre seu portal de AWS acesso URL no Painel do https://console.aws.amazon.com/singlesignon/.

   a. No portal de AWS acesso, na guia Contas, selecione a conta individual a ser gerenciada. As funções do seu usuário são exibidas. Escolha Teclas de acesso para obter credenciais para a linha de comando ou acesso programático para o conjunto de permissões apropriado. Use o conjunto de permissões `PowerUserAccess` predefinido ou qualquer conjunto de permissões que você ou seu empregador tenha criado para aplicar as permissões de privilégios mínimos para desenvolvimento.

   b. Na caixa de diálogo Obter credenciais, selecione MacOS e Linux ou Windows, dependendo do sistema operacional.

   c. Escolha o método de credenciais do IAM Identity Center para obter os `SSO Region` valores `Issuer URL` e necessários para a próxima etapa. Nota: `SSO Start URL` pode ser usado de forma intercambiável com. `Issuer URL`

3. No prompt de AWS CLI comando, execute o `aws configure sso` comando. Quando solicitado, insira os valores de configuração que você coletou na etapa anterior. Para obter detalhes sobre esse AWS CLI comando, consulte Configurar seu perfil com o `aws configure sso` assistente.

   a. Para o prompt`SSO Start URL`, insira o valor obtido`Issuer URL`.

   b. Para o nome do CLI perfil, recomendamos que você insira *`default`* quando estiver começando. Para obter informações sobre como definir perfis não padrão (nomeados) e suas variáveis de ambiente associadas, consulte Perfis.

4. (Opcional) No prompt de AWS CLI comando, confirme a identidade da sessão ativa executando o `aws sts get-caller-identity` comando. A resposta deve mostrar o conjunto de permissões do IAM Identity Center que você configurou.

5. Se você estiver usando um AWS SDK, crie um aplicativo para você SDK em seu ambiente de desenvolvimento.

   a. Para algunsSDKs, pacotes adicionais, como `SSO` e, `SSOOIDC` devem ser adicionados ao seu aplicativo antes que você possa usar a autenticação do IAM Identity Center. Para obter detalhes, consulte seu específicoSDK.

   b. Se você configurou anteriormente o acesso ao AWS, revise o AWS `credentials` arquivo compartilhado para verificar se há algumAWS chaves de acesso. Você deve remover todas

as credenciais estáticas antes que a ferramenta SDK or use as credenciais do IAM Identity Center devido à Entenda a cadeia de fornecedores de credenciais precedência.

Para saber mais sobre como as ferramentas SDKs e usam e atualizam as credenciais usando essa configuração, consulte. Entenda a autenticação do IAM Identity Center

Dependendo da duração da sessão configurada, seu acesso acabará expirando e a ferramenta SDK or encontrará um erro de autenticação. Para atualizar a sessão do portal de acesso novamente quando necessário, use o AWS CLI para executar o `aws sso login` comando.

Você pode estender a duração da sessão do portal de acesso ao IAM Identity Center e a duração da sessão do conjunto de permissões. Isso aumenta a quantidade de tempo que você pode executar o código antes de precisar entrar manualmente novamente com a AWS CLI. Para obter mais informações, consulte os seguintes tópicos no Guia do usuário do AWS IAM Identity Center :

- IAMDuração da sessão do Identity Center — Configure a duração das sessões do portal de AWS acesso de seus usuários
- Permissão definir duração da sessão: definir duração da sessão

Para obter detalhes sobre todas as configurações SDKs e ferramentas do provedor do IAM Identity Center, consulte IAMProvedor de credenciais do Identity Center este guia.

Entenda a autenticação do IAM Identity Center

Termos relevantes do Centro de Identidade do IAM

Os termos a seguir ajudam você a entender o processo e a configuração por trás da AWS IAM Identity Center. A documentação das APIs do SDK AWS usa nomes diferentes do IAM Identity Center para alguns desses conceitos de autenticação. É útil conhecer os dois nomes.

A tabela a seguir mostra como os nomes alternativos se relacionam.

Nome do IAM Identity Center	Nome da API do SDK	Descrição
Identity Center	`sso`	Embora o Single Sign-On da AWS tenha sido renomeado , os namespaces da API `sso`

Nome do IAM Identity Center	Nome da API do SDK	Descrição
		manterão seu nome original para fins de compatibilidade com versões anteriores. Para obter mais informaçõ es, consulte Renomear o IAM Identity Center no Guia do usuário AWS IAM Identity Center.
Console do IAM Identity Center Console administrativo		O console que você usa para configurar o single sign-on.
URL do portal de acesso da AWS		Um URL exclusivo para sua conta do IAM Identity Center, como `https://xxx.awsapps.com/start` . Você faz login neste portal usando suas credenciais de login do IAM Identity Center.
Sessão do portal de acesso ao IAM Identity Center	Sessão de autenticação	Fornece um token de acesso do portador ao chamador.
Sessão de definição de permissões		A sessão do IAM que o SDK usa internamente para fazer as chamadas de AWS service (Serviço da AWS). Em discussões informais, você pode ver isso incorreta mente chamado de “sessão de funções”.

Nome do IAM Identity Center	Nome da API do SDK	Descrição
Credenciais do conjunto de permissões	Credenciais AWS credenciais sigv4	As credenciais que o SDK realmente usa para a maioria das chamadas de AWS service (Serviço da AWS) (especificamente, todas as chamadas AWS service (Serviço da AWS) sigv4). Em discussões informais, você pode ver isso incorretamente chamado de “credenciais de função”.
Provedor de credenciais do IAM Identity Center	Provedor de credenciais de SSO	Como você obtém as credenciais, como a classe ou o módulo que fornece a funcionalidade.

Entenda a resolução de credenciais do SDK para Serviços da AWS

A API do IAM Identity Center troca as credenciais do token do portador por credenciais sigv4. A maioria dos Serviços da AWS são APIs sigv4, com algumas exceções, como Amazon CodeWhisperer e Amazon CodeCatalyst. A seguir, descrevemos o processo de resolução de credenciais para dar suporte à maioria das chamadas AWS service (Serviço da AWS) para o código do seu aplicativo por meio de AWS IAM Identity Center.

Iniciar uma sessão do portal de acesso AWS

- Inicie o processo entrando na sessão com suas credenciais.
  - Use o comando `aws sso login` no AWS Command Line Interface (AWS CLI). Isso inicia uma nova sessão do IAM Identity Center se você ainda não tiver uma sessão ativa.
- Ao iniciar uma nova sessão, você recebe um token de atualização e um token de acesso do IAM Identity Center. O AWS CLI também atualiza um arquivo JSON de cache SSO com um novo token de acesso e token de atualização e o disponibiliza para uso por SDKs.
- Se você já tiver uma sessão ativa, o comando AWS CLI reutilizará a sessão existente e expirará sempre que a sessão existente expirar. Para saber como definir a duração de uma sessão do IAM

Identity Center, consulte Configurar a duração das sessões do portal de acesso AWS de seus usuários no Guia do usuário AWS IAM Identity Center.

- A duração máxima da sessão foi estendida para 90 dias para reduzir a necessidade de logins frequentes.

Como o SDK obtém credenciais para chamadas AWS service (Serviço da AWS)

Os SDKs fornecem acesso para Serviços da AWS quando você instancia um objeto cliente por serviço. Quando o perfil selecionado do arquivo de `config` compartilhado da AWS é configurado para resolução de credenciais do IAM Identity Center, o IAM Identity Center é usado para resolver as credenciais do seu aplicativo.

- O processo de resolução de credenciais é concluído durante o runtime quando um cliente é criado.

Para recuperar as credenciais das APIs sigv4 usando o login único do IAM Identity Center, o SDK usa o token de acesso do IAM Identity Center para obter uma sessão do IAM. Essa sessão do IAM é chamada de sessão de conjunto de permissões e fornece acesso AWS ao SDK assumindo um perfil do IAM.

- A duração da sessão do conjunto de permissões é definida independentemente da duração da sessão do IAM Identity Center.
  - Para saber como definir a duração da sessão do conjunto de permissões, consulte Definir a duração da sessão no Guia do usuário AWS IAM Identity Center.
- Lembre-se de que as credenciais do conjunto de permissões também são chamadas de credenciais e credenciais AWS e credenciais sigv4 na maioria das documentações da API do SDK AWS.

As credenciais do conjunto de permissões são retornadas de uma chamada para getRoleCredentials da API IAM Identity Center para o SDK. O objeto cliente do SDK usa esse perfil do IAM assumido para fazer chamadas para o AWS service (Serviço da AWS), como pedir ao Amazon S3 que liste os buckets em sua conta. O objeto cliente pode continuar operando usando essas credenciais do conjunto de permissões até que a sessão do conjunto de permissões expire.

Expiração e atualização da sessão

Ao usar o SSOconfiguração do provedor de token, o token de acesso por hora obtido do IAM Identity Center é atualizado automaticamente usando o token de atualização.

- Se o token de acesso expirar quando o SDK tentar usá-lo, o SDK usará o token de atualização para tentar obter um novo token de acesso. O IAM Identity Center compara o token de atualização com a duração da sessão do portal de acesso do IAM Identity Center. Se o token de atualização não expirar, o IAM Identity Center responderá com outro token de acesso.
- Esse token de acesso pode ser usado para atualizar a sessão do conjunto de permissões de clientes existentes ou para resolver credenciais para novos clientes.

No entanto, se a sessão do portal de acesso do IAM Identity Center expirar, nenhum novo token de acesso será concedido. Portanto, a duração do conjunto de permissões não pode ser renovada. Ele expirará (e o acesso será perdido) sempre que a duração da sessão definida em cache expirar para os clientes existentes.

Qualquer código que crie um novo cliente falhará na autenticação assim que a sessão do IAM Identity Center expirar. Isso ocorre porque as credenciais do conjunto de permissões não são armazenadas em cache. Seu código não conseguirá criar um novo cliente e concluir o processo de resolução de credenciais até que você tenha um token de acesso válido.

Para recapitular, quando o SDK precisa de novas credenciais de conjunto de permissões, ele primeiro verifica se há credenciais válidas existentes e as usa. Isso se aplica se as credenciais são para um novo cliente ou para um cliente existente com credenciais expiradas. Se as credenciais não forem encontradas ou não forem válidas, o SDK chama a API do IAM Identity Center para obter novas credenciais. Para chamar a API, ela precisa do token de acesso. Se o token de acesso expirar, o SDK usará o token de atualização para tentar obter um novo token de acesso a partir do seriço IAM Identity Center. Esse token é concedido se sua sessão do portal de acesso ao IAM Identity Center não tiver expirado.

IAM Roles Anywhere

Você pode usar o IAM Roles Anywhere para obter credenciais de segurança temporárias no IAM para workloads, como servidores, contêineres e aplicativos executados fora do AWS. Para usar o IAM Roles Anywhere, seu workload deve usar certificados X.509. Seu administrador de nuvem deve fornecer o certificado e a chave privada necessários para configurar o IAM Roles Anywhere como seu provedor de credenciais.

Etapa 1: configurar IAM Roles Anywhere

O IAM Roles Anywhere fornece uma maneira de obter credenciais temporárias para um workload ou processo executado fora do AWS. Uma âncora de confiança é estabelecida com a autoridade de

certificação para obter credenciais temporárias para o perfil do IAM associado. A função define as permissões que seu workload terá quando seu código for autenticado com o IAM Roles Anywhere.

Para ver as etapas para configurar a âncora de confiança, o perfil do IAM e o perfil do IAM Roles Anywhere, consulte Como criar uma âncora de confiança e AWS Identity and Access Managementum perfil em Roles Anywhere no Guia do usuário do IAM Roles Anywhere.

> **Note**
>
> Um perfil no Guia do usuário do IAM Roles Anywhere se refere a um conceito exclusivo no serviço IAM Roles Anywhere. Não está relacionado aos perfis no arquivo do `config` da AWS compartilhado.

Etapa 2: usar IAM Roles Anywhere

Para obter credenciais de segurança temporárias do IAM Roles Anywhere, use a ferramenta de assistente de credenciais fornecida pelo IAM Roles Anywhere. A ferramenta de credenciais implementa o processo de assinatura do IAM Roles Anywhere.

Para obter instruções sobre como baixar a ferramenta de assistente de credenciais, consulte Obter credenciais de segurança temporárias do Roles Anywhere AWS Identity and Access Management no Guia do usuário do IAM Roles Anywhere.

Para usar credenciais de segurança temporárias do IAM Roles Anywhere com SDKs AWS e o AWS CLI, você pode definir a configuração `credential_process` no arquivo `config` da AWS compartilhado. Os SDKs AWS CLI oferecem suporte a um provedor de credenciais de processo que usa `credential_process` para autenticar. O seguinte mostra a estrutura geral a definir `credential_process`.

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

O comando `credential-process` da ferramenta auxiliar retorna credenciais temporárias em um formato JSON padrão compatível com a configuração `credential_process`. Observe que o nome do comando contém um hífen, mas o nome da configuração contém um sublinhado. O comando requer os seguintes parâmetros:

- `private-key` – O caminho para a chave privada que assinou a solicitação.

- `certificate` – O caminho para o certificado.
- `role-arn` – O ARN da função para a qual obter credenciais temporárias.
- `profile-arn` – O ARN do perfil que fornece um mapeamento para a função especificada.
- `trust-anchor-arn` – O ARN da âncora de confiança usada para autenticar.

Seu administrador de nuvem deve fornecer o certificado e uma chave privada. Todos os três valores de ARN podem ser copiados do AWS Management Console. O exemplo a seguir mostra um arquivo `config` compartilhado que configura a recuperação de credenciais temporárias da ferramenta auxiliar.

```
[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
```

Para parâmetros opcionais e detalhes adicionais da ferramenta auxiliar, consulte Assistente de credenciais do IAM Roles Anywhere no GitHub.

Para obter detalhes sobre a própria configuração do SDK e o provedor de credenciais do processo, consulte Provedor de credenciais de processo neste guia.

Assuma uma função com AWS credenciais

Assumir um perfil envolve o uso de um conjunto de credenciais temporárias de segurança para acessar recursos da AWS aos quais você talvez não tenha acesso de outra forma. Essas credenciais de segurança temporárias consistem em um ID de chave de acesso, uma chave de acesso secreta e um token de segurança. Para saber mais sobre API solicitações AWS Security Token Service (AWS STS), consulte Ações na AWS Security Token Service APIreferência.

Para configurar sua ferramenta SDK ou ferramenta para assumir uma função, você deve primeiro criar ou identificar uma função específica a ser assumida. IAMas funções são identificadas exclusivamente por uma função Amazon Resource Name () ARN. Os perfis estabelecem as relações de confiança com uma outra entidade. A entidade confiável que usa a função pode ser uma AWS service (Serviço da AWS) ou outra Conta da AWS. Para saber mais sobre IAM funções, consulte Usando IAM funções no Guia do IAM usuário.

Depois que a IAM função for identificada, se você tiver a confiança dessa função, poderá configurar sua ferramenta SDK ou sua ferramenta para usar as permissões concedidas pela função.

> **Note**
>
> É uma prática AWS recomendada usar endpoints regionais sempre que possível e configurar seusRegião da AWS.

Assuma uma IAM função

Ao assumir uma função, AWS STS retorna um conjunto de credenciais de segurança temporárias. Essas credenciais são provenientes de outro perfil ou da instância ou contêiner em que seu código está sendo executado. Geralmente, esse tipo de assumir uma função é usado quando você tem AWS credenciais para uma conta, mas seu aplicativo precisa acessar recursos em outra conta.

Etapa 1: configurar uma IAM função

Para configurar sua ferramenta SDK ou ferramenta para assumir uma função, você deve primeiro criar ou identificar uma função específica a ser assumida. IAMas funções são identificadas de forma exclusiva usando uma função. ARN Os perfis estabelecem relações de confiança com outra entidade, normalmente dentro da sua conta ou para acesso entre contas. Para configurar isso, consulte Criação de IAM funções no Guia IAM do usuário.

Etapa 2: configurar a ferramenta SDK or

Configure a ferramenta SDK or para obter credenciais de `credential_source` ou`source_profile`.

Use `credential_source` para obter credenciais de um ECS contêiner da Amazon, de uma EC2 instância da Amazon ou de variáveis de ambiente.

Use `source_profile` para obter credenciais de outro perfil. O `source_profile` também suporta o encadeamento de perfis, que são hierarquias de perfis em que um perfil assumido é então usado para assumir outro perfil.

Quando você especifica isso em um perfil, a ferramenta SDK or faz automaticamente a AWS STS `AssumeRole`APIchamada correspondente para você. Para recuperar e usar credenciais temporárias

assumindo uma função, especifique os seguintes valores de configuração no arquivo compartilhado. AWS `config` Para obter mais detalhes sobre cada uma dessas configurações, consulte a seção Assuma as configurações do provedor de credenciais do perfil.

- `role_arn`- Da IAM função que você criou na Etapa 1
- Configure um `source_profile` ou `credential_source`
- (Opcional) `duration_seconds`
- (Opcional) `external_id`
- (Opcional) `mfa_serial`
- (Opcional) `role_session_name`

Os exemplos a seguir mostram a configuração de ambas as opções de perfis assumidos em um arquivo compartilhado `config`:

```
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
```

Para obter mais detalhes sobre todas as configurações do provedor de credenciais para assumir o perfil, consulte este guia Assuma o perfil de provedor de credenciais.

Assuma uma função com identidade web ou OpenID Connect

Assumir um perfil envolve o uso de um conjunto de credenciais temporárias de segurança para acessar recursos da AWS aos quais você talvez não tenha acesso de outra forma. Essas credenciais de segurança temporárias consistem em um ID de chave de acesso, uma chave de acesso secreta e um token de segurança. Para saber mais sobre API solicitações AWS Security Token Service (AWS STS), consulte Ações na AWS Security Token Service APIreferência.

Para configurar sua ferramenta SDK ou ferramenta para assumir uma função, você deve primeiro criar ou identificar uma função específica a ser assumida. IAMas funções são identificadas exclusivamente por uma função Amazon Resource Name () ARN. Os perfis estabelecem as relações de confiança com uma outra entidade. A entidade confiável que usa a função pode ser um provedor

de identidade da Web, o OpenID Connect (OIDC) ou SAML uma federação. Para saber mais sobre IAM funções, consulte Métodos para assumir uma função no Guia do IAM usuário.

Depois que a IAM função for configurada no seuSDK, se essa função estiver configurada para confiar em seu provedor de identidade, você poderá configurá-lo ainda mais SDK para assumir essa função a fim de obter AWS credenciais temporárias.

> **Note**
>
> É uma prática AWS recomendada usar endpoints regionais sempre que possível e configurar seusRegião da AWS.

Federar com identidade da Web ou OpenID Connect

Você pode usar os JSON Web Tokens (JWTs) de provedores de identidade pública, como Login With Amazon, Facebook, Google, para obter AWS credenciais temporárias usando`AssumeRoleWithWebIdentity`. Dependendo de como eles são usados, eles JWTs podem ser chamados de tokens de ID ou tokens de acesso. Você também pode usar JWTs emitidos por provedores de identidade (IdPs) que sejam compatíveis com OIDC o protocolo de descoberta da, como EntraId ou PingFederate.

Se você estiver usando o Amazon Elastic Kubernetes Service, esse recurso fornece a capacidade de especificar funções IAM diferentes para cada uma das suas contas de serviço em um cluster da Amazon. EKS Esse recurso do Kubernetes é distribuído JWTs para seus pods, que são então usados por esse provedor de credenciais para obter credenciais temporárias. AWS Para obter mais informações sobre essa EKS configuração da Amazon, consulte as IAMfunções das contas de serviço no Guia EKS do usuário da Amazon. No entanto, para uma opção mais simples, recomendamos que você use o Amazon EKS Pod Identities em vez disso, se for SDKcompatível.

Etapa 1: configurar um provedor de identidade e uma IAM função

Para configurar a federação com um IdP externo, use um provedor de IAM identidade para informar AWS sobre o IdP externo e sua configuração. Isso estabelece confiança entre seu IdP Conta da AWS e o IdP externo. Antes de configurar o SDK para usar o JSON Web Token (JWT) para autenticação, você deve primeiro configurar o provedor de identidade (IdP) e IAM a função usada para acessá-lo. Para configurá-los, consulte Criação de uma função para identidade na web ou OpenID Connect Federation (console) no Guia do IAMusuário.

Etapa 2: configurar a ferramenta SDK or

Configure a ferramenta SDK or para usar um JSON Web Token (JWT) AWS STS para autenticação.

Quando você especifica isso em um perfil, a ferramenta SDK or faz automaticamente a AWS STS `AssumeRoleWithWebIdentity`APIchamada correspondente para você. Para recuperar e usar credenciais temporárias usando a federação de identidade da web, especifique os seguintes valores de configuração no arquivo compartilhado AWS `config`. Para obter mais detalhes sobre cada uma dessas configurações, consulte a seção Assuma as configurações do provedor de credenciais do perfil.

- `role_arn`- Da IAM função que você criou na Etapa 1
- `web_identity_token_file`: do IdP externo
- (Opcional) `duration_seconds`
- (Opcional) `role_session_name`

Veja a seguir um exemplo de uma configuração de arquivo `config` compartilhado para assumir um perfil com a identidade da web:

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

> **Note**
>
> Para aplicações móveis, recomendamos o uso do Amazon Cognito. O Amazon Cognito atua como um agente de identidades e realiza a maioria do trabalho de federação para você. No entanto, o provedor de identidade do Amazon Cognito não está incluído nas bibliotecas principais das ferramentas SDKs e, como outros provedores de identidade. Para acessar o Amazon CognitoAPI, inclua o cliente do serviço Amazon Cognito na compilação ou nas bibliotecas da sua ferramenta ou ferramenta. SDK Para uso com AWS SDKs, consulte Exemplos de código no Guia do Desenvolvedor do Amazon Cognito.

Para obter mais detalhes sobre todas as configurações do provedor de credenciais para assumir o perfil, consulte este guia Assuma o perfil de provedor de credenciais.

AWS chaves de acesso

Use credenciais de curto prazo

Recomendamos configurar o seu SDK ou ferramenta para usar IAMAutenticação do Identity Center para sua ferramenta SDK ou para usar as opções de duração de sessão estendida.

No entanto, para configurar diretamente as credenciais temporárias do SDK ou da ferramenta, consulte Autenticar usando credenciais de curto prazo.

Use credenciais de longo prazo

> **Warning**
>
> Para evitar riscos de segurança, não use usuários do IAM para autenticação ao desenvolver software com propósito específico ou trabalhar com dados reais. Em vez disso, use federação com um provedor de identidade, como AWS IAM Identity Center.

Gerencie o acesso em Contas da AWS

Como prática recomendada de segurança, recomendamos o uso AWS Organizations com o IAM Identity Center para gerenciar o acesso em todos os seus Contas da AWS. Para obter mais informações, consulte Práticas recomendadas de segurança no IAM no Guia do usuário do IAM.

Você pode criar usuários no IAM Identity Center, usar o Microsoft Active Directory, usar um provedor de identidade (IdP) SAML 2.0 ou federar seu IdP individualmente para. Contas da AWSUsando uma dessas abordagens, você pode fornecer uma experiência de login único para seus usuários. Você também pode aplicar a autenticação multifator (MFA) e usar credenciais temporárias para acesso. Conta da AWS Isso difere de um usuário do IAM, que é uma credencial de longo prazo que pode ser compartilhada e que pode aumentar o risco de segurança de seus recursos AWS .

Crie usuários do IAM somente para ambientes de sandbox

Se você é novato AWS, pode criar um usuário de teste do IAM e usá-lo para executar tutoriais e explorar o que AWS tem a oferecer. Não há problema em usar esse tipo de credencial quando você está aprendendo, mas recomendamos que você evite usá-la fora de um ambiente sandbox.

Para os seguintes casos de uso, pode fazer sentido começar com os usuários do IAM em AWS:

- Comece a usar seu AWS SDK ou ferramenta e explore Serviços da AWS em um ambiente sandbox.
- Executar scripts agendados, trabalhos e outros processos automatizados que não oferecem suporte a um processo de login assistido por humanos como parte de seu aprendizado.

Se você estiver usando usuários do IAM fora desses casos de uso, faça a transição para o IAM Identity Center ou federe seu provedor de identidade o mais rápido Contas da AWS possível. Para obter mais informações, consulte Federação de identidades em AWS.

Garanta chaves de acesso para usuários do IAM

Você deve alternar chaves de acesso de usuário do IAM regularmente. Siga as orientações em Alternando chaves de acesso no Guia do usuário do IAM. Se você acredita que compartilhou acidentalmente suas chaves de acesso de usuário do IAM, alterne suas chaves de acesso.

As chaves de acesso do usuário do IAM devem ser armazenadas no AWS `credentials` arquivo compartilhado na máquina local. Não armazene as chaves de acesso do usuário do IAM em seu código. Não inclua arquivos de configuração que contenham suas chaves de acesso de usuário do IAM em nenhum software de gerenciamento de código-fonte. Ferramentas externas, como o projeto de código aberto git-secrets, podem ajudá-lo a evitar o envio inadvertido de informações confidenciais em um repositório Git. Para obter mais informações, consulte Identidades IAM (usuários, grupos e funções) no Guia Usuário do IAM.

Para configurar um usuário do IAM para começar, consulte Autenticar com credenciais de longo prazo.

Autenticar usando credenciais de curto prazo

Recomendamos configurar sua ferramenta SDK ou ferramenta para uso IAMAutenticação do Identity Center para sua ferramenta SDK ou com opções de duração de sessão estendida. No entanto, você pode copiar e usar credenciais temporárias que estão disponíveis no portal de AWS acesso. As novas credenciais precisarão ser copiadas quando essas expirarem. É possível usar as credenciais temporárias em um perfil ou usá-las como valores para propriedades do sistema e variáveis de ambiente.

Prática recomendada: em vez de gerenciar manualmente as chaves de acesso e um token no arquivo de credenciais, recomendamos que seu aplicativo use credenciais temporárias fornecidas por:

- Um serviço de AWS computação, como executar seu aplicativo no Amazon Elastic Compute Cloud ou em. AWS Lambda
- Outra opção na cadeia de fornecedores de credenciais, comoIAMAutenticação do Identity Center para sua ferramenta SDK ou.
- Ou use o Provedor de credenciais de processo para recuperar credenciais temporárias.

Configurar um arquivo de credenciais usando credenciais de curto prazo recuperadas do portal de acesso AWS

1. Criar um arquivo de credenciais compartilhadas.
2. No arquivo de credenciais, cole o texto do espaço reservado a seguir até colar as credenciais temporárias de trabalho.

```
[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. Salve o arquivo. Agora, o arquivo `~/.aws/credentials` deve existir em seu sistema de desenvolvimento local. Esse arquivo contém o perfil [padrão] que a ferramenta SDK or usa se um perfil nomeado específico não for especificado.
4. Faça login no portal de AWS acesso.
5. Siga estas instruções para atualização manual de credenciais para copiar as credenciais da IAM função do AWS portal de acesso.

   a. Para a etapa 4 nas instruções vinculadas, escolha o nome da IAM função que concede acesso às suas necessidades de desenvolvimento. Essa função geralmente tem um nome como PowerUserAccessou Desenvolvedor.

   b. Para a etapa 7 nas instruções vinculadas, selecione a opção Adicionar manualmente um perfil ao seu arquivo de credenciais da AWS e copie o conteúdo.

6. Copie e as credenciais copiadas em seu arquivo `credentials` local. O nome do perfil gerado não é necessário se você estiver usando o perfil `default`. Seu arquivo deve se parecer com o seguinte.

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

```
aws_session_token=IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZVERYLONG
```

7. Salve o arquivo `credentials`.

Ao SDK criar um cliente de serviço, ele acessará essas credenciais temporárias e as usará para cada solicitação. As configurações da IAM função escolhida na etapa 5a determinam por quanto tempo as credenciais temporárias são válidas. A duração máxima é de doze horas.

Depois que as credenciais temporárias expirarem, repita as etapas de 4 a 7.

Autenticar com credenciais de longo prazo

> **Warning**
>
> Para evitar riscos de segurança, não use IAM usuários para autenticação ao desenvolver software específico ou trabalhar com dados reais. Em vez disso, use federação com um provedor de identidade, como AWS IAM Identity Center.

Se você usa um IAM usuário para executar seu código, a ferramenta SDK or em seu ambiente de desenvolvimento se autentica usando credenciais de IAM usuário de longo prazo no arquivo compartilhado AWS `credentials`. Analise as melhores práticas de segurança no IAM tópico e faça a transição para o IAM Identity Center ou outras credenciais temporárias o mais rápido possível.

Avisos e orientações importantes para credenciais

Avisos para credenciais

- NOTUse as credenciais raiz da sua conta para acessar os AWS recursos. Estas credenciais fornecem acesso ilimitado à conta e são difíceis de revogar.
- NOTColoque chaves de acesso literais ou informações de credenciais nos arquivos do seu aplicativo. Se colocar, criará um risco de exposição acidental das credenciais se, por exemplo, fizer upload do projeto em um repositório público.
- NOTInclua arquivos que contenham credenciais em sua área de projeto.
- Esteja ciente de que todas as credenciais armazenadas no AWS `credentials` arquivo compartilhado são armazenadas em texto simples.

Orientação adicional para gerenciar credenciais com segurança

Para uma discussão geral sobre como gerenciar AWS credenciais com segurança, consulte Melhores práticas para gerenciar chaves de AWS acesso no. Referência geral da AWS Além dessa discussão, considere o seguinte:

- Use IAMfunções para tarefas do Amazon Elastic Container Service (AmazonECS).
- Use IAMfunções para aplicativos que estão sendo executados em EC2 instâncias da Amazon.

Pré-requisitos: Crie uma conta AWS

Para usar um IAM usuário para acessar AWS serviços, você precisa de uma AWS conta e AWS credenciais.

1. Crie uma conta.

   Para criar uma AWS conta, consulte Primeiros passos: você é um AWS usuário iniciante? no Guia AWS Account Management de referência.
2. Crie um usuário administrativo.

   Evite usar a conta de usuário raiz (a conta inicial criada) para acessar serviços e o console de gerenciamento. Em vez disso, crie uma conta de usuário administrativo, conforme explicado em Criar um usuário administrativo no Guia IAM do usuário.

   Depois de criar a conta de usuário administrativo e registrar os detalhes de login, saia da conta de usuário raiz e faça login novamente usando a conta administrativa.

Nenhuma dessas contas é apropriada para desenvolvimento AWS ou execução de aplicativos AWS. Como prática recomendada, você precisa criar usuários, conjuntos de permissões ou perfis de serviço que sejam apropriados para essas tarefas. Para obter mais informações, consulte Aplique permissões de privilégio mínimo, no Guia do usuário do IAM.

Etapa 1: Crie seu IAM usuário

- Crie seu IAM usuário seguindo o procedimento Criação de IAM usuários (console) no Guia do IAM usuário. Ao criar seu IAM usuário:
  - Recomendamos que você selecione Fornecer acesso ao usuário ao AWS Management Console. Isso permite que você visualize informações Serviços da AWS relacionadas ao

código que você está executando em um ambiente visual, como a verificação de registros de AWS CloudTrail diagnóstico ou o upload de arquivos para o Amazon Simple Storage Service, o que é útil ao depurar seu código.

- Em Definir permissões - Opções de permissão, selecione Anexar políticas diretamente para saber como você deseja atribuir permissões a esse usuário.
  - A maioria dos SDK tutoriais de “Introdução” usa o serviço Amazon S3 como exemplo. Para fornecer à aplicação acesso total ao Amazon S3, selecione a política `AmazonS3FullAccess` para anexar a esse usuário.
- Você pode ignorar as etapas opcionais desse procedimento em relação à definição de limites de permissão ou tags.

Etapa 2: obter as chaves de acesso

1. No painel de navegação do IAM console, selecione Usuários e, em seguida, selecione o **`User name`** usuário que você criou anteriormente.
2. Na página do usuário, selecione a página Credenciais de segurança. Depois, em Chaves de acesso, selecione Criar chave de acesso.
3. Para Criar chave de acesso Etapa 1, escolha Interface de linha de comando (CLI) ou Código local. Ambas as opções geram o mesmo tipo de chave para usar com SDKs o. AWS CLI
4. Em Criar chave de acesso: etapa 2, insira uma tag opcional e selecione Próximo.
5. Em Criar chave de acesso, Etapa 3, selecione Baixar arquivo.csv para salvar um **`.csv`** arquivo com a chave de acesso e a chave de acesso secreta do IAM usuário. Você precisará dessas informações posteriormente.

   > ⚠ Warning
   >
   > Use medidas de segurança apropriadas para manter essas credenciais seguras.

6. Selecione Concluído.

Etapa 3: atualizar o arquivo **`credentials`** compartilhado

1. Crie ou abra o arquivo AWS `credentials` compartilhado. Esse arquivo é `~/.aws/credentials` em sistemas Linux e macOS e `%USERPROFILE%\.aws\credentials` no Windows. Para obter mais informações, consulte Arquivos de credenciais de local.

2. Adicione o texto a seguir ao arquivo `credentials` compartilhado. Substitua o valor de ID de exemplo e o valor de chave de exemplo pelos valores no arquivo `.csv` que você baixou anteriormente.

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

3. Salve o arquivo.

O arquivo `credentials` compartilhado é a forma mais comum de armazenar credenciais. Eles também podem ser definidos como variáveis de ambiente, consulte AWS chaves de acesso para ver os nomes das variáveis de ambiente. Essa é uma forma de você começar, mas recomendamos que você faça a transição para o IAM Identity Center ou outras credenciais temporárias o mais rápido possível. Depois de deixar de usar credenciais de longo prazo, lembre-se de excluir essas credenciais do arquivo `credentials` compartilhado.

Usando IAM funções para EC2 instâncias da Amazon

Este exemplo aborda a configuração de uma AWS Identity and Access Management função com acesso ao Amazon S3 para uso em seu aplicativo implantado em uma instância da Amazon. EC2

Para executar seu AWS SDK aplicativo em uma instância do Amazon Elastic Compute Cloud, crie uma IAM função e, em seguida, conceda à sua EC2 instância da Amazon acesso a essa função. Para obter mais informações, consulte IAMFunções para a Amazon EC2 no Guia EC2 do usuário da Amazon.

Criar um perfil do IAM

O AWS SDK aplicativo que você desenvolve provavelmente acessa pelo menos um AWS service (Serviço da AWS) para realizar ações. Crie uma IAM função que conceda as permissões necessárias para que seu aplicativo seja executado.

Esse procedimento cria uma função que concede acesso somente de leitura ao Amazon S3 como exemplo. Muitos dos AWS SDK guias têm tutoriais de “introdução” que são lidos no Amazon S3.

1. Faça login no AWS Management Console e abra o IAM console em https://console.aws.amazon.com/iam/.

2. No painel de navegação, selecione Perfis e, em seguida, Criar perfil.
3. Em Selecionar entidade confiável, em Tipo de entidade confiável, escolha AWS service (Serviço da AWS).
4. Em Caso de uso, escolha Amazon eEC2, em seguida, selecione Avançar.
5. Em Adicionar permissões, marque a caixa de seleção do Acesso somente leitura do Amazon S3 na lista de políticas e, em seguida, selecione Próximo.
6. Insira um nome para o perfil e, em seguida, escolha Criar perfil. Lembre-se desse nome porque você precisará dele ao criar sua EC2 instância da Amazon.

Inicie uma EC2 instância da Amazon e especifique sua IAM função

Você pode criar e iniciar uma EC2 instância da Amazon usando sua IAM função fazendo o seguinte:

- Siga Execute rapidamente uma instância no Guia do EC2 usuário da Amazon. No entanto, antes da etapa final de envio, faça o seguinte:
  - Em Detalhes avançados, em Perfil da IAM instância, escolha a função que você criou na etapa anterior.

Com essa EC2 configuração IAM e a Amazon, você pode implantar seu aplicativo na EC2 instância da Amazon e seu aplicativo terá acesso de leitura ao serviço Amazon S3.

Conecte-se à EC2 instância

Conecte-se à EC2 instância da Amazon para poder transferir seu aplicativo para ela e, em seguida, executar o aplicativo. Você precisará do arquivo que contém a parte privada do par de chaves usado em Par de chaves (login) ao criar sua instância, ou seja, o PEM arquivo.

Você pode fazer isso seguindo as orientações para seu tipo de instância: Conecte-se à sua instância Linux ou Conecte-se à sua instância do Windows. Ao conectar-se, faça isso de maneira que possa transferir arquivos da sua máquina de desenvolvimento para sua instância.

> **Note**
>
> No terminal Linux ou macOS, você pode usar o comando secure copy para copiar seu aplicativo. Para usar `scp` com um key pair, você pode usar o seguinte comando:`scp -i path/to/key file/to/copy ec2-user@ec2-xx-xx-xxx-xxx.compute.amazonaws.com:~`.

Para obter mais informações sobre o Windows, consulte Transferir arquivos para instâncias do Windows.

Se você estiver usando um AWS kit de ferramentas, geralmente também poderá se conectar à instância usando o kit de ferramentas. Para obter mais informações, consulte o Guia do usuário específico para o kit de ferramentas que você usa.

Execute seu aplicativo na EC2 instância

1. Copie os arquivos do aplicativo da sua unidade local para sua EC2 instância da Amazon.
2. Iniciar a aplicação e verificar se é executada com os mesmos resultados da sua máquina de desenvolvimento.
3. (Opcional) Verifique se o aplicativo usa as credenciais fornecidas pela IAM função.
    a. Faça login no AWS Management Console e abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
    b. Selecione a instância.
    c. Escolha Ações, Segurança e, em seguida, escolha Modificar IAM função.
    d. Para IAMfunção, separe a IAM função escolhendo Sem IAM função.
    e. Escolha Atualizar IAM função.
    f. Execute o aplicativo novamente e confirme se ele retorna um erro de autorização.

Referência de configurações

SDKsforneça um idioma específico paraAPIs. Serviços da AWS Eles cuidam de parte do trabalho pesado necessário para fazer API chamadas com sucesso, incluindo autenticação, comportamento de repetição e muito mais. Para fazer isso, eles SDKs têm estratégias flexíveis para obter credenciais para usar em suas solicitações, manter as configurações a serem usadas com cada serviço e obter valores a serem usados nas configurações globais.

Você pode encontrar informações detalhadas sobre as definições de configuração nas seções a seguir:

- AWS SDKse ferramentas: provedores de credenciais padronizados— Provedores de credenciais comuns padronizados em vários. SDKs
- AWS SDKse ferramentas, recursos padronizados— Recursos comuns padronizados em váriosSDKs.

Criar clientes de serviço

Para acessar programaticamente Serviços da AWS, SDKs use uma classe/objeto cliente para cada um. AWS service (Serviço da AWS) Por exemplo, se seu aplicativo precisa acessar a AmazonEC2, seu aplicativo cria um objeto EC2 cliente da Amazon para interagir com esse serviço. Em seguida, você usa o cliente de serviço para fazer solicitações para esse AWS service (Serviço da AWS). Na maioria das vezesSDKs, um objeto de cliente de serviço é imutável, então você deve criar um novo cliente para cada serviço para o qual você faz solicitações e para fazer solicitações ao mesmo serviço usando uma configuração diferente.

Precedência de configurações

As configurações globais definem recursos, provedores de credenciais e outras funcionalidades que são suportadas pela maioria SDKs e têm um amplo impacto em todas Serviços da AWS as áreas. Todos SDKs têm uma série de lugares (ou fontes) que eles verificam para encontrar um valor para as configurações globais. A seguir está a configuração da precedência de pesquisa:

1. Qualquer configuração explícita definida no código ou no próprio cliente de serviço tem precedência sobre qualquer outra coisa.

- Algumas configurações podem ser definidas por operação e podem ser alteradas conforme necessário para cada operação que você invocar. Para o AWS CLI ou AWS Tools for PowerShell, eles assumem a forma de parâmetros por operação que você insere na linha de comando. Para umSDK, as atribuições explícitas podem assumir a forma de um parâmetro que você define ao instanciar um AWS service (Serviço da AWS) cliente ou objeto de configuração ou, às vezes, ao chamar um indivíduo. API

2. Somente Java/Kotlin: a propriedade do JVM sistema para a configuração é verificada. Se estiver definido, esse valor é usado para configurar o cliente.
3. A variável de ambiente está marcada. Se estiver definido, esse valor é usado para configurar o cliente.
4. O SDK verifica a configuração no `credentials` arquivo compartilhado. Se estiver definido, o cliente o usará.
5. O `config` arquivo compartilhado para a configuração. Se a configuração estiver presente, eles a SDK usarão.
   - A variável de `AWS_PROFILE` ambiente ou a propriedade `aws.profile` JVM do sistema podem ser usadas para especificar qual perfil será SDK carregado.
6. Qualquer valor padrão fornecido pelo próprio SDK código-fonte é usado por último.

> **Note**
>
> Algumas ferramentas SDKs e ferramentas podem ser verificadas em uma ordem diferente. Além disso, algumas SDKs ferramentas oferecem suporte a outros métodos de armazenamento e recuperação de parâmetros. Por exemplo, ele AWS SDK for .NET suporta uma fonte adicional chamada SDKStore. Para obter mais informações sobre provedores que são exclusivos de uma ferramenta SDK or, consulte o guia específico da ferramenta SDK or que você está usando.

A ordem determina quais métodos têm precedência e substituem outros. Por exemplo, se você configurar um perfil no `config` arquivo compartilhado, ele só será encontrado e usado depois que a ferramenta SDK ou verificar primeiro os outros lugares. Isso significa que, se você colocar uma configuração no arquivo `credentials`, ela será usada em vez de uma encontrada no arquivo `config`. Se você configurar uma variável de ambiente com uma configuração e um valor, ela substituirá essa configuração nos arquivos `credentials` e `config`. E, finalmente, uma

configuração na operação individual (parâmetro ou API parâmetro AWS CLI da linha de comando) ou no código substituiria todos os outros valores desse comando.

Páginas de configurações

As páginas na seção de referência de configurações deste guia detalham as configurações disponíveis que podem ser definidas por meio de vários mecanismos. As tabelas a seguir listam as configurações do arquivo de configuração e credencial, as variáveis de ambiente e (para Java e KotlinSDKs) as JVM configurações que podem ser usadas fora do seu código para configurar o recurso. Cada tópico vinculado em cada lista leva você à página de configurações correspondente.

- Lista de configurações de arquivo `Config`
- Lista de configurações de arquivo `Credentials`
- Lista de variáveis de ambiente
- JVMlista de propriedades do sistema

Cada provedor ou recurso de credenciais tem uma página na qual as configurações usadas para definir essa funcionalidade são listadas. Para cada configuração, geralmente você pode definir o valor adicionando a configuração a um arquivo de configuração ou definindo uma variável de ambiente ou (somente para Java e Kotlin) definindo uma propriedade JVM do sistema. Cada configuração lista todos os métodos compatíveis para definir o valor em um bloco acima dos detalhes da descrição. Embora a precedência varie, a funcionalidade resultante é a mesma, independentemente de como você a define.

A descrição incluirá o valor padrão, se houver, que entrará em vigor se você não fizer nada. Ele também define o que é um valor válido para essa configuração.

Por exemplo, vamos ver uma configuração na página de Compactação de solicitações recursos.

As informações da configuração de `disable_request_compression` exemplo comunicam o seguinte:

- Há três maneiras equivalentes de controlar a compactação de solicitações fora da sua base de código. Você também pode:
  - Defina-o em seu arquivo de configuração usando `disable_request_compression`
  - Defina-o como uma variável de ambiente usando `AWS_DISABLE_REQUEST_COMPRESSION`

- Ou, se você estiver usando Java ou KotlinSDK, defina-o como uma propriedade JVM do sistema usando `aws.disableRequestCompression`

> **Note**
>
> Também pode haver uma maneira de configurar a mesma funcionalidade diretamente em seu código, mas esta referência não cobre isso, pois é exclusiva de cada umaSDK. Se você quiser definir sua configuração no próprio código, consulte seu SDK guia ou API referência específica.

- Se você não fizer nada, o valor padrão será`false`.
- Os únicos valores válidos para essa configuração booleana são `true` e. `false`

Na parte inferior da página de cada recurso, há uma AWS SDKs tabela de compatibilidade com.

Esta tabela mostra se você SDK suporta as configurações listadas na página. A `Supported` coluna indica o nível de suporte com os seguintes valores:

- `Yes`— As configurações são totalmente suportadas pelo texto SDK escrito.
- `Partial`— Algumas das configurações são suportadas ou o comportamento se desvia da descrição. Pois`Partial`, uma nota adicional indica o desvio.
- No— Nenhuma das configurações é suportada. Isso não afirma se a mesma funcionalidade pode ser obtida no código; apenas indica que as configurações externas listadas não são suportadas.

Lista de configurações de arquivo **`Config`**

As configurações listadas na tabela a seguir podem ser atribuídas no AWS `config` arquivo compartilhado. Eles são globais e afetam a todos os Serviços da AWS. SDKse as ferramentas também podem oferecer suporte a configurações e variáveis de ambiente exclusivas. Para ver as configurações e as variáveis de ambiente suportadas somente por um indivíduo SDK ou ferramenta, consulte esse guia específico SDK ou de ferramentas.

Nome da configuração	Detalhes
`account_id_endpoint_mode`	Endpoints baseados em contas
`api_versions`	Definições gerais de configuração
`aws_access_key_id`	AWS chaves de acesso
`aws_account_id`	Endpoints baseados em contas
`aws_secret_access_key`	AWS chaves de acesso
`aws_session_token`	AWS chaves de acesso
`ca_bundle`	Definições gerais de configuração
`credential_process`	Provedor de credenciais de processo
`credential_source`	Assuma a função de provedor de credenciais
`defaults_mode`	Padrões de configuração inteligente
`disable_request_compression`	Compactação de solicitações
`duration_seconds`	Assuma a função de provedor de credenciais

Nome da configuração	Detalhes	
`ec2_metadata_service_endpoint`	IMDSprovedor de credenciais	
`ec2_metadata_service_endpoint_mode`	IMDSprovedor de credenciais	
`ec2_metadata_v1_disabled`	IMDSprovedor de credenciais	
`endpoint_discovery_enabled`	Descoberta de endpoint	
`endpoint_url`	Endpoints específicos de serviço	
`external_id`	Assuma a função de provedor de credenciais	
`ignore_configured_endpoint_urls`	Endpoints específicos de serviço	
`max_attempts`	Comportamento de repetição	
`metadata_service_num_attempts`	Metadados da EC2 instância Amazon	
`metadata_service_timeout`	Metadados da EC2 instância Amazon	
`mfa_serial`	Assuma a função de provedor de credenciais	

Nome da configuração	Detalhes
`output`	Definições gerais de configuração
`parameter_validation`	Definições gerais de configuração
`region`	Região da AWS
`request_min_compression_size_bytes`	Compactação de solicitações
`retry_mode`	Comportamento de repetição
`role_arn`	Assuma a função de provedor de credenciais
`role_session_name`	Assuma a função de provedor de credenciais
`s3_disable_multiregion_access_points`	Pontos de acesso de várias regiões do Amazon S3
`s3_use_arn_region`	Pontos de acesso Amazon S3
`sdk_ua_app_id`	ID do aplicativo
`source_profile`	Assuma a função de provedor de credenciais
`sso_account_id`	IAMProvedor de credenciais do Identity Center
`sso_region`	IAMProvedor de credenciais do Identity Center
`sso_registration_scopes`	IAMProvedor de credenciais do Identity Center

Nome da configuração	Detalhes	
`sso_role_name`	IAMProvedor de credenciais do Identity Center	
`sso_start_url`	IAMProvedor de credenciais do Identity Center	
`sts_regional_endpoints`	AWS STS Endpoints regionais	
`use_dualstack_endpoint`	Pilha dupla e endpoints FIPS	
`use_fips_endpoint`	Pilha dupla e endpoints FIPS	
`web_identity_token_file`	Assuma a função de provedor de credenciais	

Lista de configurações de arquivo **`Credentials`**

As configurações listadas na tabela a seguir podem ser atribuídas no AWS `credentials` arquivo compartilhado. Eles são globais e afetam a todos os Serviços da AWS. SDKse as ferramentas também podem oferecer suporte a configurações e variáveis de ambiente exclusivas. Para ver as configurações e as variáveis de ambiente suportadas somente por um indivíduo SDK ou ferramenta, consulte esse guia específico SDK ou de ferramentas.

Nome da configuração	Detalhes	
`aws_access_key_id`	AWS chaves de acesso	
`aws_secret_access_key`	AWS chaves de acesso	
`aws_session_token`	AWS chaves de acesso	

Lista de variáveis de ambiente

As variáveis de ambiente suportadas pela maioria SDKs estão listadas na tabela a seguir. Eles são globais e afetam a todos os Serviços da AWS. SDKse as ferramentas também podem oferecer suporte a configurações e variáveis de ambiente exclusivas. Para ver as configurações e as variáveis de ambiente suportadas somente por um indivíduo SDK ou ferramenta, consulte esse guia específico SDK ou de ferramentas.

Nome da configuração	Detalhes
`AWS_ACCESS_KEY_ID`	AWS chaves de acesso
`AWS_ACCOUNT_ID`	Endpoints baseados em contas
`AWS_ACCOUNT_ID_ENDPOINT_MODE`	Endpoints baseados em contas
`AWS_CA_BUNDLE`	Definições gerais de configuração
`AWS_CONFIG_FILE`	Localização dos arquivos `config` e `credentials` compartilhados
`AWS_CONTAINER_AUTHORIZATION_TOKEN`	Provedor de credenciais de contêiner
`AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE`	Provedor de credenciais de contêiner
`AWS_CONTAINER_CRED`	Provedor de credenciais de contêiner

Nome da configuração	Detalhes	
ENTIALS_FULL_URI		
AWS_CONTAINER_CREDENTIALS_RELATIVE_URI	Provedor de credenciais de contêiner	
AWS_DEFAULTS_MODE	Padrões de configuração inteligente	
AWS_DISABLE_REQUEST_COMPRESSION	Compactação de solicitações	
AWS_EC2_METADATA_DISABLED	IMDSprovedor de credenciais	
AWS_EC2_METADATA_SERVICE_ENDPOINT	IMDSprovedor de credenciais	
AWS_EC2_METADATA_SERVICE_ENDPOINT_MODE	IMDSprovedor de credenciais	
AWS_EC2_METADATA_V1_DISABLED	IMDSprovedor de credenciais	
AWS_ENABLE_ENDPOINT_DISCOVERY	Descoberta de endpoint	

Nome da configuração	Detalhes
`AWS_ENDPOINT_URL`	Endpoints específicos de serviço
`AWS_ENDPOINT_URL_<SERVICE>`	Endpoints específicos de serviço
`AWS_IGNORE_CONFIGURED_ENDPOINT_URLS`	Endpoints específicos de serviço
`AWS_MAX_ATTEMPTS`	Comportamento de repetição
`AWS_METADATA_SERVICE_NUM_ATTEMPTS`	Metadados da EC2 instância Amazon
`AWS_METADATA_SERVICE_TIMEOUT`	Metadados da EC2 instância Amazon
`AWS_PROFILE`	Arquivos `config` e `credentials` compartilhados
`AWS_REGION`	Região da AWS
`AWS_REQUEST_MIN_COMPRESSION_SIZE_BYTES`	Compactação de solicitações
`AWS_RETRY_MODE`	Comportamento de repetição

Nome da configuração	Detalhes
AWS_ROLE_ARN	Assuma a função de provedor de credenciais
AWS_ROLE_SESSION_NAME	Assuma a função de provedor de credenciais
AWS_S3_DISABLE_MULTIREGION_ACCESS_POINTS	Pontos de acesso de várias regiões do Amazon S3
AWS_S3_USE_ARN_REGION	Pontos de acesso Amazon S3
AWS_SDK_UA_APP_ID	ID do aplicativo
AWS_SECRET_ACCESS_KEY	AWS chaves de acesso
AWS_SESSION_TOKEN	AWS chaves de acesso
AWS_SHARED_CREDENTIALS_FILE	Localização dos arquivos `config` e `credentials` compartilhados
AWS_STS_REGIONAL_ENDPOINTS	AWS STS Endpoints regionais
AWS_USE_DUALSTACK_ENDPOINT	Pilha dupla e endpoints FIPS
AWS_USE_FIPS_ENDPOINT	Pilha dupla e endpoints FIPS

Nome da configuração	Detalhes	
`AWS_WEB_IDENTITY_TOKEN_FILE`	Assuma a função de provedor de credenciais	

JVMlista de propriedades do sistema

Você pode usar as seguintes propriedades JVM do sistema para o AWS SDK for Java e o AWS SDK para Kotlin (visando oJVM). Consulte the section called “Como definir as propriedades do sistema JVM” para obter instruções sobre como definir as propriedades JVM do sistema.

Nome da configuração	Detalhes	
`aws.accessKeyId`	AWS chaves de acesso	
`aws.accountId`	Endpoints baseados em contas	
`aws.accountIdEndpointMode`	Endpoints baseados em contas	
`aws.configFile`	Localização dos arquivos `config` e `credentials` compartilhados	
`aws.defaultsMode`	Padrões de configuração inteligente	
`aws.disableEc2MetadataV1`	IMDSprovedor de credenciais	
`aws.disableRequestCompression`	Compactação de solicitações	

Nome da configuração	Detalhes	
`aws.ec2MetadataServiceEndpoint`	IMDSprovedor de credenciais	
`aws.ec2MetadataServiceEndpointMode`	IMDSprovedor de credenciais	
`aws.endpointDiscoveryEnabled`	Descoberta de endpoint	
`aws.endpointUrl`	Endpoints específicos de serviço	
`aws.endpointUrl<ServiceName>`	Endpoints específicos de serviço	
`aws.ignoreConfiguredEndpointUrls`	Endpoints específicos de serviço	
`aws.maxAttempts`	Comportamento de repetição	
`aws.profile`	Arquivos `config` e `credentials` compartilhados	
`aws.region`	Região da AWS	
`aws.requestMinCompressionSizeBytes`	Compactação de solicitações	
`aws.retryMode`	Comportamento de repetição	

Nome da configuração	Detalhes	
`aws.roleArn`	Assuma a função de provedor de credenciais	
`aws.roleSessionName`	Assuma a função de provedor de credenciais	
`aws.s3DisableMultiRegionAccessPoints`	Pontos de acesso de várias regiões do Amazon S3	
`aws.s3UseArnRegion`	Pontos de acesso Amazon S3	
`aws.secretAccessKey`	AWS chaves de acesso	
`aws.sessionToken`	AWS chaves de acesso	
`aws.sharedCredentialsFile`	Localização dos arquivos `config` e `credentials` compartilhados	
`aws.useDualstackEndpoint`	Pilha dupla e endpoints FIPS	
`aws.useFipsEndpoint`	Pilha dupla e endpoints FIPS	
`aws.userAgentAppId`	ID do aplicativo	
`aws.webIdentityTokenFile`	Assuma a função de provedor de credenciais	

AWS SDKse ferramentas: provedores de credenciais padronizados

Muitos provedores de credenciais foram padronizados para padrões consistentes e para funcionar da mesma forma em muitos. SDKs Essa consistência aumenta a produtividade e a clareza ao codificar em váriosSDKs. Todas as configurações podem ser substituídas no código. Para obter detalhes, consulte seu específico SDKAPI.

> **Important**
>
> Nem todos SDKs oferecem suporte a todos os fornecedores, ou mesmo a todos os aspectos de um provedor.

Tópicos

Entenda a cadeia de fornecedores de credenciais

Todos SDKs têm uma série de locais (ou fontes) que eles verificam para encontrar credenciais válidas para usar para fazer uma solicitação a um AWS service (Serviço da AWS). Depois que as credenciais válidas são encontradas, a pesquisa é interrompida. Essa busca sistemática é chamada de cadeia de fornecedores de credenciais.

Ao usar um dos provedores de credenciais padronizados, eles AWS SDKs sempre tentam renovar as credenciais automaticamente quando elas expiram. A cadeia de provedores de credenciais integrada fornece ao seu aplicativo a capacidade de atualizar suas credenciais, independentemente do provedor que você está usando na cadeia. Nenhum código adicional é necessário SDK para fazer isso.

Embora a cadeia distinta usada por cada uma SDK varie, elas geralmente incluem fontes como as seguintes:

Provedor de credencial	Descrição
AWS chaves de acesso	AWS chaves de acesso para um IAM usuário (como `AWS_ACCESS_KEY_ID` e`AWS_SECRET_ACCESS_KEY`).
Federar com identidade da Web ou OpenID Connect: assumir a função de provedor de credenciais	Faça login usando um provedor de identidade externo (IdP) conhecido, como Login with Amazon, Facebook, Google ou qualquer outro IdP compatível com OpenID Connect ()OIDC. Assuma as permissões de uma IAM função usando um JSON Web Token (JWT) de AWS Security Token Service (AWS STS).
IAMProvedor de credenciais do Identity Center	Obtenha credenciais de AWS IAM Identity Center.
Assuma o perfil de provedor de credenciais	Tenha acesso a outros recursos assumindo as permissõe s de uma IAM função. (Recupere e use credenciais temporárias para uma função).
Provedor de credenciais de contêiner	Credenciais do Amazon Elastic Container Service (AmazonECS) e do Amazon Elastic Kubernetes Service (Amazon). EKS O provedor de credenciais de contêiner busca credenciais para o aplicativo em contêiner do cliente.
Provedor de credenciais de processo	Provedores de credenciais personalizados. Obtenha suas credenciais de uma fonte ou processo externo, incluindo IAM Roles Anywhere.
IMDSprovedor de credenciais	Credenciais do perfil da instância Amazon Elastic Compute Cloud (AmazonEC2). Associe uma IAM função a cada uma de suas EC2 instâncias. As credenciais temporárias para essa função estão disponíveis para o código em execução na instância. As credenciais são

Provedor de credencial	Descrição
	entregues por meio do serviço de EC2 metadados da Amazon.

Para cada etapa da cadeia, há várias maneiras de atribuir valores de configuração. Os valores de configuração especificados no código sempre têm precedência. No entanto, também existem Variáveis de ambiente e Arquivos `config` e `credentials` compartilhados. Para obter mais informações, consulte Precedência de configurações.

SDKcadeias de fornecedores de credenciais específicas e específicas de ferramentas

Para acessar diretamente os detalhes específicos SDK da sua cadeia de fornecedores de credenciais ou da ferramenta, escolha sua ferramenta SDK ou uma das seguintes opções:

- AWS CLI
- SDKpara C++
- SDKpara Go
- SDKpara Java
- SDKpara JavaScript
- SDKpara Kotlin
- SDKpara. NET
- SDK para PHP
- SDKpara Python (Boto3)
- SDKpara Ruby
- SDKpara Rust
- SDKpara Swift
- Ferramentas para PowerShell

AWS chaves de acesso

> **Warning**
>
> Para evitar riscos de segurança, não use IAM usuários para autenticação ao desenvolver software específico ou trabalhar com dados reais. Em vez disso, use federação com um provedor de identidade, como AWS IAM Identity Center.

AWS as chaves de acesso de um IAM usuário podem ser usadas como suas AWS credenciais. O usa AWS SDK automaticamente essas AWS credenciais para assinar API solicitações AWS, para que suas cargas de trabalho possam acessar seus AWS recursos e dados de forma segura e conveniente. É recomendável sempre usar o `aws_session_token` para que as credenciais sejam temporárias e não sejam mais válidas após expirarem. Não é recomendável usar credenciais de longo prazo.

> **Note**
>
> Se AWS não conseguir atualizar essas credenciais temporárias, AWS poderá estender a validade das credenciais para que suas cargas de trabalho não sejam afetadas.

O AWS `credentials` arquivo compartilhado é o local recomendado para armazenar informações de credenciais porque está fora dos diretórios de origem do aplicativo e separado das configurações SDK específicas do arquivo compartilhado. `config`

Para saber mais sobre AWS credenciais e o uso de chaves de acesso, consulte Credenciais de AWS segurança e Gerenciamento de chaves de acesso para IAM usuários no Guia do IAMusuário.

Configure essa funcionalidade usando o seguinte:

`aws_access_key_id`- configuração de AWS **`config`** arquivo compartilhado, **`aws_access_key_id`**- configuração de AWS **`credentials`** arquivo compartilhado (método recomendado), **`AWS_ACCESS_KEY_ID`**: variável de ambiente, **`aws.accessKeyId`**- propriedade JVM do sistema: somente Java/Kotlin

Especifica a chave de AWS acesso usada como parte das credenciais para autenticar o usuário.

`aws_secret_access_key`- configuração de AWS **`config`** arquivo compartilhado, **`aws_secret_access_key`**- configuração de AWS **`credentials`** arquivo compartilhado (método recomendado), **`AWS_SECRET_ACCESS_KEY`**: variável de ambiente, **`aws.secretAccessKey`**- propriedade JVM do sistema: somente Java/Kotlin

Especifica a chave AWS secreta usada como parte das credenciais para autenticar o usuário.

`aws_session_token`- configuração de AWS **`config`** arquivo compartilhado, **`aws_session_token`**- configuração de AWS **`credentials`** arquivo compartilhado (método recomendado), **`AWS_SESSION_TOKEN`**: variável de ambiente, **`aws.sessionToken`**- propriedade JVM do sistema: somente Java/Kotlin

Especifica um token de AWS sessão usado como parte das credenciais para autenticar o usuário. Você recebe esse valor como parte das credenciais temporárias retornadas por solicitações bem-sucedidas para assumir uma função. Um token de sessão só será necessário se você especificar manualmente credenciais de segurança temporárias. No entanto, recomendamos que você use sempre credenciais de segurança temporárias em vez de credenciais de longo prazo. Para recomendações de segurança, consulte Melhores práticas de segurança em IAM.

Para obter instruções sobre como obter esses valores, consulte Autenticar usando credenciais de curto prazo.

Exemplo de configuração desses valores necessários no arquivo `config` ou `credentials`:

```
[default]
aws_access_key_id = AKIAIOSFODNN7EXAMPLE
aws_secret_access_key = wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
aws_session_token = AQoEXAMPLEH4aoAH0gNCAPy...truncated...zrkuWJOgQs8IZZaIv2BXIa2R4Olgk
```

Exemplo de configuração de variáveis de ambiente para Linux/macOS por meio da linha de comando:

```
export AWS_ACCESS_KEY_ID=AKIAIOSFODNN7EXAMPLE
export AWS_SECRET_ACCESS_KEY=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
export
 AWS_SESSION_TOKEN=AQoEXAMPLEH4aoAH0gNCAPy...truncated...zrkuWJOgQs8IZZaIv2BXIa2R4Olgk
```

Exemplo do Windows de configuração de variáveis de ambiente por meio da linha de comando:

```
setx AWS_ACCESS_KEY_ID AKIAIOSFODNN7EXAMPLE
```

```
setx AWS_SECRET_ACCESS_KEY wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
setx
 AWS_SESSION_TOKEN AQoEXAMPLEH4aoAH0gNCAPy...truncated...zrkuWJOgQs8IZZaIv2BXIa2R4Olgk
```

Compatibilidade com AWS SDKs

Os itens a seguir SDKs oferecem suporte aos recursos e configurações descritos neste tópico. Quaisquer exceções parciais estão anotadas. Todas as configurações de propriedade do JVM sistema são suportadas pelo AWS SDK for Java e pelo AWS SDK para Kotlin único.

SDK	C I	Notas ou mais informações
AWS CLI v2	Sim	
SDKpara C++	Sim	arquivo compartilhado `config` não suportado.
SDKpara Go V2 (1.x)	Sim	
SDKpara Go 1.x (V1)	Sim	Para usar as configurações do arquivo `config` compartil hado, você deve ativar o carregamento do arquivo de configuração; consulte Sessões.
SDKpara Java 2.x	Sim	
SDKpara Java 1.x	Sim	
SDKpara JavaScript 3.x	Sim	
SDKpara JavaScript 2.x	Sim	
SDKpara Kotlin	Sim	
SDKpara. NET3.x	Sim	As variáveis de ambiente não são compatíveis.
SDKpara PHP 3.x	Sim	
SDKpara Python (Boto3)	Sim	
SDKpara Ruby 3.x	Sim	

SDK	C I	Notas ou mais informações
SDKpara Rust	Sim	
SDKpara Swift	Sim	
Ferramentas para PowerShel I	Sim	As variáveis de ambiente não são compatíveis.

Assuma o perfil de provedor de credenciais

Assumir um perfil envolve o uso de um conjunto de credenciais temporárias de segurança para acessar recursos da AWS aos quais você talvez não tenha acesso de outra forma. Essas credenciais de segurança temporárias consistem em um ID de chave de acesso, uma chave de acesso secreta e um token de segurança.

Para configurar sua ferramenta SDK ou ferramenta para assumir uma função, você deve primeiro criar ou identificar uma função específica a ser assumida. IAMas funções são identificadas exclusivamente por uma função Amazon Resource Name () ARN. Os perfis estabelecem as relações de confiança com uma outra entidade. A entidade confiável que usa a função pode ser uma AWS service (Serviço da AWS), outra Conta da AWS, um provedor de identidade da Web ou OIDC uma SAML federação.

Depois que a IAM função for identificada, se você tiver a confiança dessa função, poderá configurar sua ferramenta SDK ou sua ferramenta para usar as permissões concedidas pela função. Para fazer isso, execute as configurações a seguir.

Para obter orientação sobre como começar a usar essas configurações, consulte este guia Assuma uma função com AWS credenciais.

Assuma as configurações do provedor de credenciais do perfil

Configure essa funcionalidade usando o seguinte:

`credential_source`- configuração de AWS **`config`** arquivo compartilhado

Usado em EC2 instâncias da Amazon ou contêineres do Amazon Elastic Container Service para especificar onde a ferramenta SDK ou pode encontrar credenciais que tenham permissão para assumir a função especificada com o `role_arn` parâmetro.

Valor padrão: nenhum

Valores válidos:

- Ambiente — Especifica que a ferramenta SDK or deve recuperar as credenciais de origem das variáveis de ambiente e. `AWS_ACCESS_KEY_IDAWS_SECRET_ACCESS_KEY`
- Ec2 InstanceMetadata — Especifica que a ferramenta SDK or deve usar a IAMfunção anexada ao perfil da EC2 instância para obter as credenciais de origem.
- EcsContainer— Especifica que a ferramenta SDK or deve usar a IAMfunção anexada ao ECS contêiner para obter as credenciais de origem.

Não é possível especificar `credential_source` e `source_profile` no mesmo perfil.

Exemplo de configuração em um `config` arquivo para indicar que as credenciais devem ser provenientes da Amazon: EC2

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

`duration_seconds`- configuração de AWS **`config`** arquivo compartilhado

Especifica a duração máxima da sessão da função, em segundos.

Esta configuração se aplica somente quando o perfil especifica assumir uma função.

Valor padrão: 3.600 segundos (uma hora)

Valores válidos: o valor pode variar de 900 segundos (15 minutos) até o valor configurado de duração máxima da sessão para o perfil (que pode ser até 43200, ou 12 horas). Para obter mais informações, consulte Exibir a configuração de duração máxima da sessão para uma função no Guia IAM do usuário.

Exemplo de configuração em um arquivo `config`:

```
duration_seconds = 43200
```

`external_id`- configuração de AWS **`config`** arquivo compartilhado

Especifica um identificador exclusivo que é usado por terceiros para assumir uma função em suas contas de clientes.

Esta configuração se aplica somente quando o perfil especifica assumir uma função e a política de confiança do perfil exige um valor para `ExternalId`. O valor é mapeado para o parâmetro `ExternalId` que é passado para a operação `AssumeRole` quando o perfil especifica uma função.

Valor padrão: Nenhum.

Valores válidos: consulte Como usar uma ID externa ao conceder acesso aos seus AWS recursos a terceiros no Guia do IAM usuário.

Exemplo de configuração em um arquivo `config`:

```
external_id = unique_value_assigned_by_3rd_party
```

`mfa_serial`- configuração de AWS **`config`** arquivo compartilhado

Especifica a identificação ou o número de série de um dispositivo de autenticação multifator (MFA) que o usuário deve usar ao assumir uma função.

Obrigatório ao assumir uma função em que a política de confiança dessa função inclui uma condição que exige MFA autenticação. Para obter mais informações sobreMFA, consulte Autenticação AWS multifator IAM no Guia do IAM usuário.

Valor padrão: Nenhum.

Valores válidos: o valor pode ser um número de série para um dispositivo de hardware (comoGAHT12345678) ou um Amazon Resource Name (ARN) para um MFA dispositivo virtual. O formato do ARN é: `arn:aws:iam::account-id:mfa/mfa-device-name`

Exemplo de configuração em um arquivo `config`:

Este exemplo pressupõe um MFA dispositivo virtual, chamado`MyMFADevice`, que foi criado para a conta e habilitado para um usuário.

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

`role_arn`- configuração de AWS **`config`** arquivo compartilhado, **`AWS_ROLE_ARN`**: variável de ambiente, **`aws.roleArn`**- propriedade JVM do sistema: somente Java/Kotlin

Especifica o Amazon Resource Name (ARN) de uma IAM função que você deseja usar para realizar operações solicitadas usando esse perfil.

Valor padrão: nenhum.

Valores válidos: o valor deve ser o ARN de uma IAM função, formatado da seguinte forma: `arn:aws:iam::account-id:role/role-name`

Além disso, você também deve especificar uma das seguintes configurações:

- `source_profile`: identificar outro perfil a ser usado para encontrar credenciais que tenham permissão para assumir a função nesse perfil.
- `credential_source`— Usar credenciais identificadas pelas variáveis de ambiente atuais ou credenciais anexadas a um perfil de instância da Amazon ou a uma EC2 instância de ECS contêiner da Amazon.
- `web_identity_token_file`— Usar provedores de identidade públicos ou qualquer provedor de identidade compatível com OpenID Connect (OIDC) para usuários que foram autenticados em um aplicativo móvel ou web.

`role_session_name`- configuração de AWS **`config`** arquivo compartilhado, **`AWS_ROLE_SESSION_NAME`**: variável de ambiente, **`aws.roleSessionName`**- propriedade JVM do sistema: somente Java/Kotlin

Especifica o nome a ser associado à sessão da função. Este nome aparece nos logs do AWS CloudTrail para entradas associadas a esta sessão, que pode ser útil em uma auditoria. Para obter detalhes, consulte o CloudTrail userIdentity elemento no Guia AWS CloudTrail do usuário.

Valor padrão: um parâmetro opcional. Se você não fornecer este valor, um nome de sessão será gerado automaticamente se o perfil assumir uma função.

Valores válidos: fornecidos ao `RoleSessionName` parâmetro quando o AWS CLI ou AWS API chama a `AssumeRole` operação (ou operações como a `AssumeRoleWithWebIdentity` operação) em seu nome. O valor se torna parte da função assumida do usuário Amazon Resource Name (ARN) que você pode consultar e aparece como parte das entradas de CloudTrail registro das operações invocadas por esse perfil.

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

Exemplo de configuração em um arquivo `config`:

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

`source_profile`- configuração de AWS **`config`** arquivo compartilhado

Especifica outro perfil cujas credenciais são usadas para assumir o perfil especificado pela configuração `role_arn` no perfil original. Para entender como os perfis são usados no compartilhamento AWS `config` e nos `credentials` arquivos, consulteArquivos `config` e `credentials` compartilhados.

Se você especificar um perfil que também seja um perfil de assumir função, cada perfil será assumido em ordem sequencial para resolver totalmente as credenciais. Essa cadeia é interrompida quando ele SDK encontra um perfil com credenciais. O encadeamento de funções limita sua sessão AWS CLI ou sua AWS API função a no máximo uma hora e não pode ser aumentado. Para obter mais informações, consulte Termos e conceitos de funções no Guia IAM do usuário.

Valor padrão: nenhum.

Valores válidos: um string de texto que consiste no nome de um perfil definido nos arquivos `config` e `credentials`. Você também deve especificar um valor para `role_arn` no perfil atual.

Não é possível especificar `credential_source` e `source_profile` no mesmo perfil.

Exemplo de definição em um arquivo de configuração:

```
[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
```

No exemplo anterior, o `A` perfil instrui a ferramenta SDK ou a pesquisar automaticamente as credenciais do `B` perfil vinculado. Nesse caso, o `B` perfil usa a ferramenta auxiliar de credenciais fornecida por IAM Roles Anywhere para obter credenciais para o. AWS SDK Essas credenciais temporárias são então usadas pelo código para acessar recursos da AWS . A função especificada deve ter políticas de IAM permissões anexadas que permitam a execução do código solicitado, como o comando ou o API método. AWS service (Serviço da AWS) Cada ação realizada pelo perfil `A` tem o nome da sessão da função incluído nos CloudTrail registros.

Para um segundo exemplo de encadeamento de funções, a configuração a seguir pode ser usada se você tiver um aplicativo em uma instância do Amazon Elastic Compute Cloud e quiser que esse aplicativo assuma outra função.

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

[profile B]
credential_source=Ec2InstanceMetadata
```

O perfil `A` usará as credenciais da EC2 instância da Amazon para assumir a função especificada e renovará as credenciais automaticamente.

`web_identity_token_file`- configuração de AWS **`config`** arquivo compartilhado, **`AWS_WEB_IDENTITY_TOKEN_FILE`**: variável de ambiente, **`aws.webIdentityTokenFile`**- propriedade JVM do sistema: somente Java/Kotlin

Especifica o caminho para um arquivo que contém um token de acesso de um provedor OAuth 2.0 compatível ou provedor de identidade OpenID Connect ID.

Esta configuração permite a autenticação usando provedores de federação de identidade da web, como Google, Facebook e Amazon, entre muitos outros. A ferramenta SDK ou developer carrega o conteúdo desse arquivo e o passa como `WebIdentityToken` argumento quando chama a `AssumeRoleWithWebIdentity` operação em seu nome.

Valor padrão: nenhum.

Valores válidos: este valor deve ser um nome de caminho e de arquivo. O arquivo deve conter um token de acesso OAuth 2.0 ou um token OpenID Connect fornecido a você por um provedor

de identidade. Os caminhos relativos são tratados como relativos ao diretório de trabalho do processo.

Compatibilidade com AWS SDKs

Os itens a seguir SDKs oferecem suporte aos recursos e configurações descritos neste tópico. Quaisquer exceções parciais estão anotadas. Todas as configurações de propriedade do JVM sistema são suportadas pelo AWS SDK for Java e pelo AWS SDK para Kotlin único.

SDK	C l	Notas ou mais informações
AWS CLI v2	Sim	
SDKpara C++	Parci	`credential_source` não suportado. `duration_seconds` não suportado. `mfa_serial` não suportado.
SDKpara Go V2 (1.x)	Sim	
SDKpara Go 1.x (V1)	Sim	Para usar as configurações do arquivo `config` compartilhado, você deve ativar o carregamento do arquivo de configuração; consulte Sessões.
SDKpara Java 2.x	Parci	`mfa_serial` não suportado. `duration_seconds` não suportado.
SDKpara Java 1.x	Parci	`credential_source` não suportado. `mfa_serial` não suportado. JVMpropriedades do sistema não suportadas.
SDKpara JavaScript 3.x	Sim	
SDKpara JavaScript 2.x	Parci	`credential_source` incompatível.
SDKpara Kotlin	Sim	
SDKpara. NET3.x	Sim	
SDKpara PHP 3.x	Sim	
SDKpara Python (Boto3)	Sim	

SDK	C I	Notas ou mais informações
SDKpara Ruby 3.x	Sim	
SDKpara Rust	Sim	
SDKpara Swift	Sim	
Ferramentas para PowerShel I	Sim	

Provedor de credenciais de contêiner

O provedor de credenciais de contêiner busca credenciais para o aplicativo em contêiner do cliente. Esse provedor de credenciais é útil para clientes do Amazon Elastic Container Service (AmazonECS) e do Amazon Elastic Kubernetes Service (Amazon). EKS SDKstentativa de carregar credenciais do HTTP endpoint especificado por meio de uma GET solicitação.

Se você usa a AmazonECS, recomendamos que você use uma IAM função de tarefa para melhorar o isolamento de credenciais, a autorização e a auditabilidade. Quando configurada, a Amazon ECS define a variável de `AWS_CONTAINER_CREDENTIALS_RELATIVE_URI` ambiente que as ferramentas SDKs e ferramentas usam para obter credenciais. Para configurar a Amazon ECS para essa funcionalidade, consulte IAMFunção da tarefa no Guia do desenvolvedor do Amazon Elastic Container Service.

Se você usa a AmazonEKS, recomendamos que você use o Amazon EKS Pod Identity para melhorar o isolamento de credenciais, privilégios mínimos, auditabilidade, operação independente, reutilização e escalabilidade. Tanto seu pod quanto uma IAM função estão associados a uma conta de serviço do Kubernetes para gerenciar as credenciais dos seus aplicativos. Para saber mais sobre o Amazon EKS Pod Identity, consulte Amazon EKS Pod Identities no Guia do EKS usuário da Amazon. Quando configurada, a Amazon EKS define as variáveis de `AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE` ambiente `AWS_CONTAINER_CREDENTIALS_FULL_URI` e as variáveis que as ferramentas usam para obter credenciais. SDKs Para obter informações de configuração, consulte Configurar o Amazon EKS Pod Identity Agent no Guia EKS do usuário da Amazon ou o Amazon EKS Pod Identity simplifica IAM as permissões para aplicativos em EKS clusters da Amazon no site do AWS blog.

Configure essa funcionalidade usando o seguinte:

`AWS_CONTAINER_CREDENTIALS_FULL_URI`: variável de ambiente

Especifica o HTTP URL endpoint completo a SDK ser usado ao fazer uma solicitação de credenciais. Isso inclui o esquema e o host.

Valor padrão: Nenhum.

Valores válidos: válidosURI.

Nota: essa configuração é uma alternativa para *`AWS_CONTAINER_CREDENTIALS_RELATIVE_URI`* e só será usada se *`AWS_CONTAINER_CREDENTIALS_RELATIVE_URI`* não estiver definido.

Exemplo de configuração de variáveis de ambiente para Linux/macOS por meio da linha de comando:

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

ou

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

`AWS_CONTAINER_CREDENTIALS_RELATIVE_URI`: variável de ambiente

Especifica o HTTP URL endpoint relativo a SDK ser usado ao fazer uma solicitação de credenciais. O valor é anexado ao ECS nome de host padrão da Amazon de. `169.254.170.2`

Valor padrão: Nenhum.

Valores válidos: relativos válidosURI.

Exemplo de configuração de variáveis de ambiente para Linux/macOS por meio da linha de comando:

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

`AWS_CONTAINER_AUTHORIZATION_TOKEN`: variável de ambiente

Especifica o token de autorização em texto sem formatação. Se essa variável for definida, o SDK definirá o cabeçalho de autorização na HTTP solicitação com o valor da variável de ambiente.

Valor padrão: Nenhum.

Valores válidos: string.

Nota: essa configuração é uma alternativa para *AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE* e só será usada se *AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE* não estiver definido.

Exemplo de configuração de variáveis de ambiente para Linux/macOS por meio da linha de comando:

```
export AWS_CONTAINER_CREDENTIALS_FULL_URI=http://localhost/get-credential
export AWS_CONTAINER_AUTHORIZATION_TOKEN=Basic abcd
```

`AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE`: variável de ambiente

Especifica um caminho de arquivo absoluto para um arquivo que contém o token de autorização em texto simples.

Valor padrão: Nenhum.

Valores válidos: string.

Exemplo de configuração de variáveis de ambiente para Linux/macOS por meio da linha de comando:

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

Compatibilidade com AWS SDKs

Os itens a seguir SDKs oferecem suporte aos recursos e configurações descritos neste tópico. Quaisquer exceções parciais estão anotadas. Todas as configurações de propriedade do JVM sistema são suportadas pelo AWS SDK for Java e pelo AWS SDK para Kotlin único.

SDK	C I	Notas ou mais informações
AWS CLI v2	Sim	

SDK	C l	Notas ou mais informações
SDKpara C++	Sim	
SDKpara Go V2 (1.x)	Sim	
SDKpara Go 1.x (V1)	Sim	
SDKpara Java 2.x	Sim	`AWS_CONTAINER_CREDENTIALS_FULL_URI` e também `AWS_CONTAINER_AUTHORIZATION_TOKEN` são usados para Lambda SnapStart for Java.
SDKpara Java 1.x	Sim	`AWS_CONTAINER_CREDENTIALS_FULL_URI` e também `AWS_CONTAINER_AUTHORIZATION_TOKEN` são usados para Lambda SnapStart for Java.
SDKpara JavaScript 3.x	Sim	
SDKpara JavaScript 2.x	Sim	
SDKpara Kotlin	Sim	
SDKpara. NET3.x	Sim	
SDKpara PHP 3.x	Sim	
SDKpara Python (Boto3)	Sim	
SDKpara Ruby 3.x	Sim	
SDKpara Rust	Sim	
SDKpara Swift	Sim	
Ferramentas para PowerShel l	Sim	

IAMProvedor de credenciais do Identity Center

Esse mecanismo de autenticação é usado AWS IAM Identity Center para obter acesso de login único (SSO) ao seu Serviços da AWS código.

> **Note**
>
> Na AWS SDK API documentação, o provedor de credenciais do IAM Identity Center é chamado de provedor de SSO credenciais.

Depois de ativar o IAM Identity Center, você define um perfil para suas configurações no AWS `config` arquivo compartilhado. Esse perfil é usado para se conectar ao portal de acesso do IAM Identity Center. Quando um usuário se autentica com sucesso no IAM Identity Center, o portal retorna credenciais de curto prazo para a IAM função associada a esse usuário. Para saber como o SDK obtém credenciais temporárias da configuração e as usa para AWS service (Serviço da AWS) solicitações, consulteEntenda a autenticação do IAM Identity Center.

Há duas maneiras de configurar o IAM Identity Center por meio do `config` arquivo:

- Configuração (recomendada) SSO do provedor de token — durações de sessão estendidas. Inclui suporte para durações de sessão personalizadas.
- Configuração legada não atualizável — usa uma sessão fixa de oito horas.

Em ambas as configurações, você precisa entrar novamente quando sua sessão expirar.

Os dois guias a seguir contêm informações adicionais sobre o IAM Identity Center:

- AWS IAM Identity Center Guia do usuário
- AWS IAM Identity Center APIReferência do portal

Para saber mais sobre como as ferramentas SDKs e usam e atualizam as credenciais usando essa configuração, consulte. Entenda a autenticação do IAM Identity Center

Pré-requisitos

Você deve primeiro ativar o IAM Identity Center. Para obter detalhes sobre como habilitar a autenticação do IAM Identity Center, consulte Ativação AWS IAM Identity Center no Guia AWS IAM Identity Center do Usuário.

> **Note**
>
> Como alternativa, para obter os pré-requisitos completos e a configuração necessária de `config` arquivos compartilhados, detalhada nesta página, consulte as instruções guiadas de configuração. IAMAutenticação do Identity Center para sua ferramenta SDK ou

SSOconfiguração do provedor de token

Quando você usa a configuração do provedor de SSO token, sua ferramenta AWS SDK ou atualiza automaticamente sua sessão até o período estendido da sessão. Para obter mais informações sobre a duração e a duração máxima da sessão, consulte Configurar a duração da sessão do portal de AWS acesso e dos aplicativos integrados do IAM Identity Center no Guia AWS IAM Identity Center do Usuário.

A `sso-session` seção do `config` arquivo é usada para agrupar variáveis de configuração para adquirir tokens de SSO acesso, que podem então ser usados para adquirir AWS credenciais. Para obter mais detalhes sobre essa seção em um `config` arquivo, consulteFormato do arquivo de configuração.

O exemplo de `config` arquivo compartilhado a seguir configura a ferramenta SDK or usando um `dev` perfil para solicitar credenciais do IAM Identity Center.

```
[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
```

Os exemplos anteriores mostram que você define uma `sso-session` seção e a associa a um perfil. Normalmente, `sso_account_id` e `sso_role_name` deve ser definido na `profile` seção para que eles SDK possam solicitar AWS credenciais. `sso_region`,`sso_start_url`, e `sso_registration_scopes` deve ser definido dentro da `sso-session` seção.

`sso_account_id`e `sso_role_name` não são necessários para todos os cenários de configuração de SSO token. Se seu aplicativo usa apenas Serviços da AWS essa autenticação de portador de suporte, AWS as credenciais tradicionais não são necessárias. A autenticação do portador é um esquema de HTTP autenticação que usa tokens de segurança chamados tokens do portador. Nesse cenário, `sso_account_id` e `sso_role_name` não são obrigatórios. Consulte o AWS service (Serviço da AWS) guia individual para determinar se o serviço oferece suporte à autorização do token do portador.

Os escopos de registro são configurados como parte de um `sso-session`. O escopo é um mecanismo em OAuth 2.0 para limitar o acesso de um aplicativo à conta de um usuário. O exemplo anterior é configurado `sso_registration_scopes` para fornecer o acesso necessário para listar contas e funções.

O exemplo a seguir mostra como você pode reutilizar a mesma `sso-session` configuração em vários perfis.

```
[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
```

O token de autenticação é armazenado em cache no disco sob o diretório `~/.aws/sso/cache` com um nome de arquivo baseado no nome da sessão.

Configuração herdada não atualizável

A atualização automática de tokens não é compatível usando a configuração herdada não atualizável. Em vez disso, recomendamos usar SSOconfiguração do provedor de token.

Para usar a configuração legada não atualizável, você deve especificar as seguintes configurações no seu perfil:

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

Você especifica o portal do usuário para um perfil com as configurações `sso_start_url` e `sso_region`. Você especifica as permissões com as configurações `sso_account_id` e `sso_role_name`.

O exemplo a seguir define os quatro valores necessários no arquivo `config`.

```
[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
```

O token de autenticação é armazenado em cache no disco sob o diretório `~/.aws/sso/cache` com um nome de arquivo baseado no `sso_start_url`.

IAMConfigurações do provedor de credenciais do Identity Center

Configure essa funcionalidade usando o seguinte:

`sso_start_url`- configuração de AWS **`config`** arquivo compartilhado

O URL que aponta para o emissor URL ou portal URL de acesso do IAM Identity Center da sua organização. Para obter mais informações, consulte Usando o portal de AWS acesso no Guia AWS IAM Identity Center do usuário.

Para encontrar esse valor, abra o console do IAM Identity Center, visualize o Painel e encontre o portal de AWS acesso URL.

- Como alternativa, a partir da versão 2.22.0 do AWS CLI, você pode usar o valor para AWS Emissor. URL

`sso_region`- configuração de AWS **`config`** arquivo compartilhado

O Região da AWS que contém o host do portal do IAM Identity Center; ou seja, a região que você selecionou antes de ativar o IAM Identity Center. Isso é independente da sua AWS região padrão e pode ser diferente.

Para obter uma lista completa dos Regiões da AWS e de seus códigos, consulte Endpoints regionais no Referência geral da Amazon Web Services. Para encontrar esse valor, abra o console do IAM Identity Center, visualize o Painel e encontre a Região.

`sso_account_id`- configuração de AWS **`config`** arquivo compartilhado

O ID numérico do Conta da AWS que foi adicionado por meio do AWS Organizations serviço para uso na autenticação.

Para ver a lista de contas disponíveis, acesse o console do IAM Identity Center e abra a Contas da AWSpágina. Você também pode ver a lista de contas disponíveis usando o ListAccountsAPImétodo na APIReferência do AWS IAM Identity Center Portal. Por exemplo, você pode chamar o AWS CLI método list-accounts.

`sso_role_name`- configuração de AWS **`config`** arquivo compartilhado

O nome de um conjunto de permissões provisionado como uma IAM função que define as permissões resultantes do usuário. A função deve existir no Conta da AWS especificado por`sso_account_id`. Use o nome da função, não a função Amazon Resource Name (ARN).

Os conjuntos de permissões têm IAM políticas e políticas de permissões personalizadas anexadas a eles e definem o nível de acesso que os usuários têm às suas atribuições Contas da AWS.

Para ver a lista de conjuntos de permissões disponíveis por Conta da AWS, acesse o console do IAM Identity Center e abra a Contas da AWSpágina. Escolha o nome correto do conjunto de permissões listado na Contas da AWS tabela. Você também pode ver a lista de conjuntos de permissões disponíveis usando o ListAccountRolesAPImétodo na APIReferência do AWS IAM Identity Center Portal. Por exemplo, você pode chamar o AWS CLI método list-account-roles.

`sso_registration_scopes`- configuração de AWS **`config`** arquivo compartilhado

Uma lista delimitada por vírgulas de escopos a serem autorizados para `sso-session`. Um aplicativo pode solicitar um ou mais escopos, e o token de acesso emitido para o aplicativo está

limitado aos escopos concedidos. Um escopo mínimo de `sso:account:access` deve ser concedido para recuperar um token de atualização do serviço IAM Identity Center. Para obter a lista de opções de escopo de acesso disponíveis, consulte Escopos de acesso no Guia do AWS IAM Identity Center usuário.

Esses escopos definem as permissões solicitadas para serem autorizadas para o OIDC cliente registrado e os tokens de acesso recuperados pelo cliente. Os escopos autorizam o acesso aos endpoints autorizados do token portador do IAM Identity Center.

Esta configuração não é aplicável à configuração legada não atualizável. Os tokens emitidos usando a configuração legada estão limitados ao escopo `sso:account:access` implícito.

Compatibilidade com AWS SDKs

Os itens a seguir SDKs oferecem suporte aos recursos e configurações descritos neste tópico. Quaisquer exceções parciais estão anotadas. Todas as configurações de propriedade do JVM sistema são suportadas pelo AWS SDK for Java e pelo AWS SDK para Kotlin único.

SDK	C I	Notas ou mais informações
AWS CLI v2	Sim	
SDKpara C++	Sim	
SDKpara Go V2 (1.x)	Sim	
SDKpara Go 1.x (V1)	Sim	Para usar as configurações do arquivo `config` compartil hado, você deve ativar o carregamento do arquivo de configuração; consulte Sessões.
SDKpara Java 2.x	Sim	Valores de configuração também compatíveis no arquivo `credentials` .
SDKpara Java 1.x	Não	
SDKpara JavaScript 3.x	Sim	
SDKpara JavaScript 2.x	Sim	

SDK	C I	Notas ou mais informações
SDKpara Kotlin	Sim	
SDKpara. NET3.x	Sim	
SDKpara PHP 3.x	Sim	
SDKpara Python (Boto3)	Sim	
SDKpara Ruby 3.x	Sim	
SDKpara Rust	Parci	Somente configuração herdada não atualizável.
SDKpara Swift	Sim	
Ferramentas para PowerShel l	Sim	

IMDSprovedor de credenciais

O Instance Metadata Service (IMDS) fornece dados sobre sua instância que você pode usar para configurar ou gerenciar a instância em execução. Para obter mais informações sobre os dados disponíveis, consulte Trabalhar com metadados de instância no Guia do EC2 usuário da Amazon. EC2A Amazon fornece um endpoint local disponível para instâncias que pode fornecer várias informações para a instância. Se a instância tiver uma função anexada, ela poderá fornecer um conjunto de credenciais válidas para essa função. Eles SDKs podem usar esse endpoint para resolver credenciais como parte de sua cadeia de fornecedores de credenciais padrão. O Instance Metadata Service Version 2 (IMDSv2), uma versão mais segura IMDS que usa um token de sessão, é usado por padrão. Se isso falhar devido a uma condição que não pode ser repetida (códigos de HTTP erro 403, 404, 405), IMDSv1 é usado como alternativa.

Configure essa funcionalidade usando o seguinte:

`AWS_EC2_METADATA_DISABLED`: variável de ambiente

Se você deve ou não tentar usar o Amazon EC2 Instance Metadata Service (IMDS) para obter credenciais.

Valor padrão: `false`.

Valores válidos:

- **`true`**— Não use IMDS para obter credenciais.
- **`false`**— Use IMDS para obter credenciais.

`ec2_metadata_v1_disabled`- configuração de AWS **`config`** arquivo compartilhado, **`AWS_EC2_METADATA_V1_DISABLED`**: variável de ambiente, **`aws.disableEc2MetadataV1`**- propriedade JVM do sistema: somente Java/Kotlin

Se deve ou não usar o Instance Metadata Service Version 1 (IMDSv1) como alternativa em caso IMDSv2 de falha.

> **Note**
>
> Os novos SDKs não oferecem suporte IMDSv1 e, portanto, não oferecem suporte a essa configuração. Para obter detalhes, consulte a tabela Compatibilidade com AWS SDKs.

Valor padrão: `false`.

Valores válidos:

- **`true`**— Não use IMDSv1 como substituto.
- **`false`**— Use IMDSv1 como substituto.

`ec2_metadata_service_endpoint`- configuração de AWS **`config`** arquivo compartilhado, **`AWS_EC2_METADATA_SERVICE_ENDPOINT`**: variável de ambiente, **`aws.ec2MetadataServiceEndpoint`**- propriedade JVM do sistema: somente Java/Kotlin

O ponto final de. IMDS Esse valor substitui o local padrão em que as ferramentas pesquisarão AWS SDKs os metadados da EC2 instância da Amazon.

Valor padrão: se `ec2_metadata_service_endpoint_mode` for igual a `IPv4`, o endpoint padrão será `http://169.254.169.254`. Se `ec2_metadata_service_endpoint_mode` for igual a `IPv6`, o endpoint padrão será `http://[fd00:ec2::254]`.

Valores válidos: válidosURI.

`ec2_metadata_service_endpoint_mode`- configuração de AWS **`config`** arquivo compartilhado, **`AWS_EC2_METADATA_SERVICE_ENDPOINT_MODE`**: variável de ambiente, **`aws.ec2MetadataServiceEndpointMode`**- propriedade JVM do sistema: somente Java/Kotlin

O modo de endpoint do. IMDS

Valor padrão:`IPv4`.

Valores válidos: `IPv4`, `IPv6`.

> **Note**
>
> O provedor de IMDS credenciais faz parte doEntenda a cadeia de fornecedores de credenciais. No entanto, o provedor de IMDS credenciais só é verificado após vários outros provedores que estão nesta série. Portanto, se você quiser que seu programa use as credenciais desse provedor, você deve remover outros provedores de credenciais válidos da sua configuração ou usar um perfil diferente. Como alternativa, em vez de confiar na cadeia de fornecedores de credenciais para descobrir automaticamente qual provedor retorna credenciais válidas, especifique o uso do provedor de IMDS credenciais no código. Você pode especificar fontes de credenciais diretamente ao criar clientes de serviço.

Segurança para IMDS credenciais

Por padrão, quando não AWS SDK estiver configurado com credenciais válidas, SDK ele tentará usar o Amazon EC2 Instance Metadata Service (IMDS) para recuperar as credenciais de uma função. AWS Esse comportamento pode ser desativado definindo a variável de ambiente `AWS_EC2_METADATA_DISABLED` como `true`. Isso evita atividades desnecessárias na rede e aumenta a segurança em redes não confiáveis nas quais o Amazon EC2 Instance Metadata Service pode ser representado.

> **Note**
>
> AWS SDKclientes configurados com credenciais válidas nunca usarão IMDS para recuperar credenciais, independentemente de qualquer uma dessas configurações.

Desativando o uso das credenciais da Amazon EC2 IMDS

A forma como você define essa variável de ambiente depende do sistema operacional em uso, bem como se você deseja ou não que a alteração seja persistente.

Linux e macOS

Os clientes que usam Linux ou macOS podem definir essa variável de ambiente com o comando a seguir:

```
$ export AWS_EC2_METADATA_DISABLED=true
```

Se você quiser que essa configuração seja persistente em várias sessões de shell e reinicializações do sistema, você pode adicionar o comando acima ao seu arquivo de perfil de shell, como `.bash_profile`, `.zsh_profile` ou `.profile`.

Windows

Os clientes que usam Windows podem definir essa variável de ambiente com o comando a seguir:

```
$ set AWS_EC2_METADATA_DISABLED=true
```

Se você quiser que essa configuração seja persistente em várias sessões de shell e reinicializações do sistema, use o seguinte comando em vez disso:

```
$ setx AWS_EC2_METADATA_DISABLED=true
```

> **Note**
>
> O comando setx não aplica o valor à sessão atual do shell, então você precisará recarregar ou reabrir o shell para que a alteração entre em vigor.

Compatibilidade com AWS SDKs

Os itens a seguir SDKs oferecem suporte aos recursos e configurações descritos neste tópico. Quaisquer exceções parciais estão anotadas. Todas as configurações de propriedade do JVM sistema são suportadas pelo AWS SDK for Java e pelo AWS SDK para Kotlin único.

SDK	C l	Notas ou mais informações
AWS CLI v2	Sim	
SDKpara C++	Sim	
SDKpara Go V2 (1.x)	Sim	
SDKpara Go 1.x (V1)	Sim	Para usar as configurações do arquivo `config` compartil hado, você deve ativar o carregamento do arquivo de configuração; consulte Sessões.
SDKpara Java 2.x	Sim	
SDKpara Java 1.x	Parci	JVMpropriedades do sistema: Use `com.amazonaws.sdk. disableEc2MetadataV1` em vez de`aws.disab leEc2MetadataV1` ; `aws.ec2MetadataSer viceEndpoint` e `aws.ec2MetadataServiceEndpo intMode` não suportado.
SDKpara JavaScript 3.x	Sim	
SDKpara JavaScript 2.x	Sim	
SDKpara Kotlin	Sim	Não usa IMDSv1 fallback.
SDKpara. NET3.x	Sim	
SDKpara PHP 3.x	Sim	
SDKpara Python (Boto3)	Sim	
SDKpara Ruby 3.x	Sim	
SDKpara Rust	Sim	Não usa IMDSv1 fallback.
SDKpara Swift	Sim	

SDK	C l	Notas ou mais informações
Ferramentas para PowerShel l	Sim	Você pode desativar o IMDSv1 fallback explicitamente no código usando. `[Amazon.Util.EC2InstanceMetadata]::EC2MetadataV1Disabled = $true`

Provedor de credenciais de processo

SDKsfornecem uma forma de estender a cadeia de fornecedores de credenciais para casos de uso personalizados. Esse provedor pode ser usado para fornecer implementações personalizadas, como recuperar credenciais de um repositório de credenciais local ou integrar-se ao seu provedor de identificação local.

Por exemplo, o IAM Roles Anywhere usa `credential_process` para obter credenciais temporárias em nome do seu aplicativo. Para configurar `credential_process` para esse uso, consulte IAM Roles Anywhere.

> **Note**
>
> O seguinte descreve um método de obtenção de credenciais de um processo externo e pode ser usado se você estiver executando software fora do AWS. Se você estiver construindo em um AWS recurso de computação, use outros provedores de credenciais. Ao usar essa opção, certifique-se de que o arquivo de configuração esteja o mais bloqueado possível usando as melhores práticas de segurança para seu sistema operacional. Confirme se sua ferramenta de credencial personalizada não grava nenhuma informação secreta`StdErr`, porque o e SDKs AWS CLI pode capturar e registrar essas informações, potencialmente expondo-as a usuários não autorizados.

Configure essa funcionalidade usando o seguinte:

`credential_process`- compartilhado AWS **`config`**configuração de arquivo

Especifica um comando externo que a ferramenta SDK or executa em seu nome para gerar ou recuperar credenciais de autenticação para uso. A configuração especifica o nome de um programa/comando que será invocadoSDK. Quando SDK invoca o processo, ele espera que o

processo grave os dados. JSON `stdout` O provedor personalizado deve retornar informações em um formato específico. Essas informações contêm as credenciais que a ferramenta SDK or pode usar para autenticar você.

> **Note**
>
> O provedor de credenciais do processo faz parte do Entenda a cadeia de fornecedores de credenciais. No entanto, o provedor de credenciais do processo só é verificado após vários outros provedores que estão nesta série. Portanto, se você quiser que seu programa use as credenciais deste provedor, você deve remover outros provedores de credenciais válidos da sua configuração ou usar um perfil diferente. Como alternativa, em vez de confiar na cadeia de fornecedores de credenciais para descobrir automaticamente qual provedor retorna credenciais válidas, especifique o uso do provedor de credenciais do processo no código. Você pode especificar fontes de credenciais diretamente ao criar clientes de serviço.

Especificando o caminho para o programa de credenciais

O valor da configuração é uma string que contém um caminho para um programa que a ferramenta de desenvolvimento SDK ou executa em seu nome:

- O caminho e o nome do arquivo podem consistir somente dos seguintes caracteres: A-Z, a-z, 0-9, hífen (-), sublinhado (_), barra (/), barra invertida (\) e espaço.
- Se o caminho ou o nome do arquivo contiver um espaço, coloque o caminho completo e o nome do arquivo entre aspas duplas (“ ”).
- Se um nome de parâmetro ou um valor de parâmetro tiver um espaço, coloque esse elemento entre aspas duplas (“ ”). Coloque somente o nome ou o valor entre aspas, não o par.
- Não inclua variáveis de ambiente nas strings. Por exemplo, não inclua `$HOME` ou `%USERPROFILE%`.
- Não especifique a pasta base como ~. * Você deve especificar o caminho completo ou o nome do arquivo base. Se houver um nome de arquivo base, o sistema tentará encontrar o programa nas pastas especificadas pela variável de ambiente `PATH`. O caminho varia de acordo com o sistema operacional:

  O exemplo a seguir mostra a configuração de credential_process no arquivo compartilhado `config` no Linux/macOS.

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

O exemplo a seguir mostra a configuração de credential_process no arquivo compartilhado `config` no Windows.

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

- Pode ser especificado em um perfil dedicado:

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

Saída válida do programa de credenciais

O SDK executa o comando conforme especificado no perfil e, em seguida, lê os dados do fluxo de saída padrão. O comando especificado, seja um script ou um programa binário, deve gerar uma JSON saída `STDOUT` que corresponda à sintaxe a seguir.

```
{
    "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**
>
> No momento da elaboração deste documento, a chave `Version` deve ser definida como 1. Isso pode aumentar ao longo do tempo conforme a estrutura evolui.

A `Expiration` chave é um carimbo de data/hora RFC3339 formatado. Se a `Expiration` chave não estiver presente na saída da ferramenta, SDK presume-se que as credenciais sejam credenciais de longo prazo que não são atualizadas. Caso contrário, as credenciais serão

consideradas temporárias e serão atualizadas automaticamente com a nova execução do comando `credential_process` antes de expirarem.

> **Note**
>
> O SDK não armazena em cache as credenciais do processo externo da mesma forma que assume as credenciais de função. Se o armazenamento em cache for obrigatório, implemente-o no processo externo.

O processo externo pode retornar um código de retorno diferente de zero para indicar que ocorreu um erro ao recuperar as credenciais.

Compatibilidade com AWS SDKs

Os itens a seguir SDKs oferecem suporte aos recursos e configurações descritos neste tópico. Quaisquer exceções parciais estão anotadas. Todas as configurações de propriedade do JVM sistema são suportadas pelo AWS SDK for Java e o AWS SDK para Kotlin somente.

SDK	C l	Notas ou mais informações
AWS CLI v2	Sim	
SDKpara C++	Sim	
SDKpara Go V2 (1.x)	Sim	
SDKpara Go 1.x (V1)	Sim	Para usar as configurações do arquivo `config` compartil hado, você deve ativar o carregamento do arquivo de configuração; consulte Sessões.
SDKpara Java 2.x	Sim	
SDKpara Java 1.x	Sim	
SDKpara JavaScript 3.x	Sim	
SDKpara JavaScript 2.x	Sim	

SDK	C I	Notas ou mais informações
SDKpara Kotlin	Sim	
SDKpara. NET3.x	Sim	
SDKpara PHP 3.x	Sim	
SDKpara Python (Boto3)	Sim	
SDKpara Ruby 3.x	Sim	
SDKpara Rust	Sim	
SDKpara Swift	Sim	
Ferramentas para PowerShel l	Sim	

AWS SDKse ferramentas, recursos padronizados

Muitos recursos foram padronizados para padrões consistentes e para funcionar da mesma forma em muitos. SDKs Essa consistência aumenta a produtividade e a clareza ao codificar em váriosSDKs. Todas as configurações podem ser substituídas no código. Consulte suas configurações específicas SDK API para obter detalhes.

> **Important**
>
> Nem todos SDKs oferecem suporte a todos os recursos, ou mesmo a todos os aspectos de um recurso.

Tópicos

- Endpoints baseados em contas
- ID da aplicação
- Metadados da EC2 instância Amazon
- Pontos de acesso Amazon S3

- Pontos de acesso de várias regiões do Amazon S3
- Região da AWS
- AWS STS Endpoints regionais
- Pilha dupla e endpoints FIPS
- Descoberta de endpoint
- Definições gerais da configuração
- IMDScliente
- Comportamento de repetição
- Compactação de solicitações
- Endpoints específicos de serviço
- Padrões de configuração inteligente

Endpoints baseados em contas

Os endpoints baseados em conta ajudam a garantir alto desempenho e escalabilidade usando sua Conta da AWS ID para otimizar o roteamento de AWS service (Serviço da AWS) solicitações de serviços que oferecem suporte a esse recurso. Quando você usa um AWS SDK provedor de credenciais e um serviço que oferece suporte a endpoints baseados em contas, eles constroem e usam SDK automaticamente um endpoint baseado em conta em vez de um endpoint regional. Os endpoints baseados em conta assumem a forma de`https://<account-id>.ddb.<region>.amazonaws.com`, onde *`<account-id>`* é substituído por seu Conta da AWS ID e *`<region>`* substituído por seu Região da AWS

Por padrão, o ID da conta é coletado quando a solicitação é processada e usada para construir um endpoint. A resolução de credenciais também ocorre quando a solicitação é processada e pode alterar o método de resolução do endpoint. Dependendo do provedor de credenciais que você está usando, o ID da conta pode ter uma origem diferente.

Configure essa funcionalidade usando o seguinte:

`aws_account_id`- configuração de AWS **`config`** arquivo compartilhado, **`AWS_ACCOUNT_ID`**: variável de ambiente, **`aws.accountId`**- propriedade JVM do sistema: somente Java/Kotlin

> O Conta da AWS ID. Usado para roteamento de endpoints baseado em contas. Um Conta da AWS ID tem um formato como 111122223333.

O roteamento de endpoints baseado em conta fornece melhor desempenho de solicitações para alguns serviços.

`account_id_endpoint_mode`- configuração de AWS **`config`** arquivo compartilhado, **`AWS_ACCOUNT_ID_ENDPOINT_MODE`**: variável de ambiente, **`aws.accountIdEndpointMode`**- propriedade JVM do sistema: somente Java/Kotlin

Essa configuração é usada para desativar o roteamento de endpoints baseado em conta, se necessário, e ignorar as regras baseadas em contas.

Valor padrão: `preferred`

Valores válidos:

- **`preferred`**— O endpoint deve incluir o ID da conta, se disponível.
- **`disabled`**— Um endpoint resolvido não inclui o ID da conta.
- **`required`**— O endpoint deve incluir o ID da conta. Se o ID da conta não estiver disponível, SDK ocorrerá um erro.

Compatibilidade com AWS SDKs

Os itens a seguir SDKs oferecem suporte aos recursos e configurações descritos neste tópico. Quaisquer exceções parciais estão anotadas. Todas as configurações de propriedade do JVM sistema são suportadas pelo AWS SDK for Java e pelo AWS SDK para Kotlin único.

SDK	Com l	Lançado na SDK versão	Notas ou mais informações
AWS CLI v2	Não		
SDKpara C++	Não		
SDKpara Go V2 (1.x)	Sim	v1.35.0	
SDKpara Go 1.x (V1)	Não		
SDKpara Java 2.x	Sim	v2.28.4	

SDK	Com l	Lançado na SDK versão	Notas ou mais informações
SDKpara Java 1.x	Sim	v1.12.771	
SDKpara JavaScript 3.x	Sim	v3.656.0	
SDKpara JavaScript 2.x	Não		
SDKpara Kotlin	Sim	v1.3.37	
SDKpara. NET3.x	Não		
SDKpara PHP 3.x	Sim	v3.318.0	
SDKpara Python (Boto3)	Não		
SDKpara Ruby 3.x	Sim	v1.123.0	
SDKpara Rust	Não		
SDKpara Swift	Não		
Ferramentas para PowerShell	Não		

ID da aplicação

Um único Conta da AWS pode ser usado por vários aplicativos de clientes para fazer chamadas para Serviços da AWS. O ID do aplicativo fornece uma maneira de os clientes identificarem qual aplicativo de origem fez um conjunto de chamadas usando um Conta da AWS. AWS SDKse os serviços não usam nem interpretam esse valor a não ser para trazê-lo de volta às comunicações com o cliente. Por exemplo, esse valor pode ser incluído em e-mails operacionais ou no AWS Health Dashboard para identificar com exclusividade quais dos seus aplicativos estão associados à notificação.

Configure essa funcionalidade usando o seguinte:

`sdk_ua_app_id`- compartilhado AWS **`config`**configuração de arquivo, **`AWS_SDK_UA_APP_ID`**: variável de ambiente, **`aws.userAgentAppId`**- propriedade JVM do sistema: somente Java/Kotlin

Essa configuração é uma string exclusiva que você atribui ao seu aplicativo para identificar quais dos seus aplicativos em um determinado aplicativo. Conta da AWS faz chamadas para AWS.

Valor padrão: `None`

Valores válidos: string com comprimento máximo de 50. Letras, números e os seguintes caracteres especiais são permitidos: `!`,`$`,`%`,`&`,`*`,`+`,`-`,`.`,`,`,`^`,`_`,`` ` ``,`|`,`~`.

Exemplo de configuração desse valor no arquivo `config`:

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

Exemplo de configuração de variáveis de ambiente para Linux/macOS por meio da linha de comando:

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

Exemplo do Windows de configuração de variáveis de ambiente por meio da linha de comando:

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

Se você incluir símbolos que tenham um significado especial para a concha que está sendo usada, escape do valor conforme apropriado.

Compatibilidade com AWS SDKs

Os itens a seguir SDKs oferecem suporte aos recursos e configurações descritos neste tópico. Quaisquer exceções parciais estão anotadas. Todas as configurações de propriedade do JVM sistema são suportadas pelo AWS SDK for Java e o AWS SDK para Kotlin somente.

SDK	C l	Notas ou mais informações
AWS CLI v2	Sim	
SDKpara C++	Sim	arquivo compartilhado `config` não suportado.
SDKpara Go V2 (1.x)	Sim	
SDKpara Go 1.x (V1)	Não	
SDKpara Java 2.x	Parci	Configuração de `config` arquivo compartilhado não suportada; variável de ambiente não suportada.
SDKpara Java 1.x	Não	
SDKpara JavaScript 3.x	Sim	
SDKpara JavaScript 2.x	Não	
SDKpara Kotlin	Sim	
SDKpara. NET3.x	Sim	As variáveis de ambiente não são compatíveis.
SDKpara PHP 3.x	Sim	
SDKpara Python (Boto3)	Sim	
SDKpara Ruby 3.x	Sim	
SDKpara Rust	Sim	
SDKpara Swift	Não	
Ferramentas para PowerShel l	Não	

Metadados da EC2 instância Amazon

EC2A Amazon fornece um serviço em instâncias chamado Instance Metadata Service (IMDS). Para saber mais sobre esse serviço, consulte Trabalhar com metadados de instância no Guia do EC2 usuário da Amazon. Ao tentar recuperar credenciais em uma EC2 instância da Amazon que tenha sido configurada com uma IAM função, a conexão com o serviço de metadados da instância é ajustável.

Configure essa funcionalidade usando o seguinte:

`metadata_service_num_attempts`- compartilhado AWS **`config`**configuração de arquivo, **`AWS_METADATA_SERVICE_NUM_ATTEMPTS`**: variável de ambiente

Esta configuração especifica o número de tentativas totais a serem feitas antes de desistir ao recuperar dados do serviço de metadados de instância.

Valor padrão: 1

Valores válidos: número maior ou igual a 1.

`metadata_service_timeout`- compartilhado AWS **`config`**configuração de arquivo, **`AWS_METADATA_SERVICE_TIMEOUT`**: variável de ambiente

Especifica o número de segundos antes de atingir o tempo limite ao recuperar dados do serviço de metadados da instância.

Valor padrão: 1

Valores válidos: número maior ou igual a 1.

Exemplo de configuração desses valores no arquivo `config`:

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

Exemplo de configuração de variáveis de ambiente para Linux/macOS por meio da linha de comando:

```
export AWS_METADATA_SERVICE_NUM_ATTEMPTS=10
```

```
export AWS_METADATA_SERVICE_TIMEOUT=10
```

Exemplo do Windows de configuração de variáveis de ambiente por meio da linha de comando:

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

Compatibilidade com AWS SDKs

Os itens a seguir SDKs oferecem suporte aos recursos e configurações descritos neste tópico. Quaisquer exceções parciais estão anotadas. Todas as configurações de propriedade do JVM sistema são suportadas pelo AWS SDK for Java e o AWS SDK para Kotlin somente.

SDK	C l	Notas ou mais informações
AWS CLI v2	Sim	
SDKpara C++	Não	
SDKpara Go V2 (1.x)	Não	
SDKpara Go 1.x (V1)	Não	
SDKpara Java 2.x	Não	
SDKpara Java 1.x	Parci	Somente `AWS_METADATA_SERVICE_TIMEOUT` é suportado.
SDKpara JavaScript 3.x	Não	
SDKpara JavaScript 2.x	Não	
SDKpara Kotlin	Não	
SDKpara. NET3.x	Não	
SDKpara PHP 3.x	Sim	
SDKpara Python (Boto3)	Sim	

SDK	C l	Notas ou mais informações
SDKpara Ruby 3.x	Não	
SDKpara Rust	Não	
SDKpara Swift	Não	
Ferramentas para PowerShel l	Não	

Pontos de acesso Amazon S3

O serviço Amazon S3 fornece pontos de acesso como uma forma alternativa de interagir com os buckets do Amazon S3. Os pontos de acesso têm políticas e configurações exclusivas aplicadas a eles, em vez de diretamente ao bucket. Com AWS SDKs, você pode usar o ponto de acesso Amazon Resource Names (ARNs) no campo do bucket para API operações em vez de especificar o nome do bucket explicitamente. Eles são usados para operações específicas, como usar um ponto de acesso ARN com `GetObject`para buscar um objeto de um bucket ou usar um ponto de acesso ARN com `PutObject`para adicionar um objeto a um bucket.

Para saber mais sobre os pontos de acesso do Amazon S3 eARNs, consulte Uso de pontos de acesso no Guia do usuário do Amazon S3.

Configure essa funcionalidade usando o seguinte:

`s3_use_arn_region`- compartilhado AWS **`config`**configuração de arquivo, **`AWS_S3_USE_ARN_REGION`**: variável de ambiente, **`aws.s3UseArnRegion`**- propriedade JVM do sistema: somente Java/Kotlin, Para configurar o valor diretamente no código, consulte seu específico SDK diretamente.

> Essa configuração controla se o SDK usa o ponto de acesso ARN Região da AWS para construir o endpoint regional para a solicitação. O SDK valida que o ARN Região da AWS é servido pelo mesmo AWS particionar conforme configurado pelo cliente Região da AWS para evitar chamadas entre partições que provavelmente falharão. Se definido por multiplicação, a configuração configurada pelo código terá precedência, seguida pela configuração da variável de ambiente.
>
> Valor padrão: `false`

Valores válidos:

- **`true`**— Os SDK usos ARN do Região da AWS ao construir o endpoint em vez do configurado pelo cliente Região da AWS. Exceção: se o cliente estiver configurado Região da AWS é um FIPS Região da AWS, então deve corresponder ARN ao Região da AWS. Caso contrário, ocorrerá um erro.
- **`false`**— Os SDK usos que o cliente configurou Região da AWS ao construir o endpoint.

Compatibilidade com AWS SDKs

Os itens a seguir SDKs oferecem suporte aos recursos e configurações descritos neste tópico. Quaisquer exceções parciais estão anotadas. Todas as configurações de propriedade do JVM sistema são suportadas pelo AWS SDK for Java e o AWS SDK para Kotlin somente.

SDK	C l	Notas ou mais informações
AWS CLI v2	Sim	
SDKpara C++	Sim	
SDKpara Go V2 (1.x)	Sim	
SDKpara Go 1.x (V1)	Sim	Para usar as configurações do arquivo `config` compartil hado, você deve ativar o carregamento do arquivo de configuração; consulte Sessões.
SDKpara Java 2.x	Sim	
SDKpara Java 1.x	Sim	JVMpropriedade do sistema não suportada.
SDKpara JavaScript 3.x	Sim	
SDKpara JavaScript 2.x	Sim	
SDKpara Kotlin	Sim	
SDKpara. NET3.x	Sim	Não segue a precedência padrão; o valor do arquivo `config` compartilhado tem precedência sobre a variável de ambiente.

SDK	C l	Notas ou mais informações
SDKpara PHP 3.x	Sim	
SDKpara Python (Boto3)	Sim	
SDKpara Ruby 3.x	Sim	
SDKpara Rust	Não	
SDKpara Swift	Não	
Ferramentas para PowerShell	Sim	Não segue a precedência padrão; o valor do arquivo `config` compartilhado tem precedência sobre a variável de ambiente.

Pontos de acesso de várias regiões do Amazon S3

Os pontos de acesso multirregionais do Amazon S3 fornecem um endpoint global que os aplicativos podem usar para atender solicitações de buckets do Amazon S3 localizados em vários Regiões da AWS. Você pode usar pontos de acesso multirregionais para criar aplicativos multirregionais com a mesma arquitetura usada em uma única região e, em seguida, executar esses aplicativos em qualquer lugar do mundo.

Para saber mais sobre pontos de acesso de várias regiões, consulte Pontos de acesso de várias regiões no Amazon S3, no Guia do usuário do Amazon S3.

Para saber mais sobre o ponto de acesso multirregional Amazon Resource Names (ARNs), consulte Fazer solicitações usando um ponto de acesso multirregional no Guia do usuário do Amazon S3.

Para saber mais sobre pontos de acesso de várias regiões, consulte Gerenciar pontos de acesso de várias regiões no Guia do usuário do Amazon S3.

O algoritmo SigV4a é a implementação de assinatura usada para assinar as solicitações globais da região. Esse algoritmo é obtido pelo por SDK meio de uma dependência doAWS Bibliotecas Common Runtime (CRT).

Configure essa funcionalidade usando o seguinte:

`s3_disable_multiregion_access_points`- compartilhado AWS **`config`**configuração de arquivo, **`AWS_S3_DISABLE_MULTIREGION_ACCESS_POINTS`**: variável de ambiente, **`aws.s3DisableMultiRegionAccessPoints`**- propriedade JVM do sistema: somente Java/Kotlin, Para configurar o valor diretamente no código, consulte seu específico SDK diretamente.

Essa configuração controla se o SDK potencial tenta solicitações entre regiões. Se definido por multiplicação, a configuração configurada pelo código terá precedência, seguida pela configuração da variável de ambiente.

Valor padrão: `false`

Valores válidos:

- **`true`**: interrompe o uso de solicitações entre regiões.
- **`false`**: permite solicitações entre regiões usando pontos de acesso multirregionais.

Compatibilidade com AWS SDKs

Os itens a seguir SDKs oferecem suporte aos recursos e configurações descritos neste tópico. Quaisquer exceções parciais estão anotadas. Todas as configurações de propriedade do JVM sistema são suportadas pelo AWS SDK for Java e o AWS SDK para Kotlin somente.

SDK	C l	Notas ou mais informações
AWS CLI v2	Sim	
SDKpara C++	Sim	
SDKpara Go V2 (1.x)	Sim	
SDKpara Go 1.x (V1)	Não	
SDKpara Java 2.x	Sim	
SDKpara Java 1.x	Não	
SDKpara JavaScript 3.x	Sim	
SDKpara JavaScript 2.x	Não	

SDK	C I	Notas ou mais informações
SDKpara Kotlin	Sim	
SDKpara. NET3.x	Sim	
SDKpara PHP 3.x	Sim	
SDKpara Python (Boto3)	Sim	
SDKpara Ruby 3.x	Sim	
SDKpara Rust	Sim	
SDKpara Swift	Não	
Ferramentas para PowerShell	Sim	

Região da AWS

Regiões da AWS são um conceito importante a ser entendido ao trabalhar com Serviços da AWS.

Com Regiões da AWS, você pode acessar aqueles Serviços da AWS que residem fisicamente em uma área geográfica específica. Isso pode ser útil para manter os seus dados e aplicativos em execução próximo ao lugar em que você e os seus usuários os acessarão. As regiões fornecem tolerância a falhas, estabilidade e resiliência e também podem reduzir a latência. Com Regiões, você pode criar recursos redundantes que permanecem disponíveis e não são afetados por uma interrupção regional.

A maioria das AWS service (Serviço da AWS) solicitações está associada a uma região geográfica específica. Os atributos que você cria em uma Região não existem em qualquer outra Região, a menos que você use explicitamente um atributo de replicação oferecido por AWS service (Serviço da AWS). Por exemplo, o Amazon S3 e o Amazon EC2 oferecem suporte à replicação entre regiões. Alguns serviços, comoIAM, não têm recursos regionais.

A Referência geral da AWS contém as seguintes informações:

- Para entender a relação entre Regiões e endpoints e ver uma lista dos endpoints regionais existentes, consulte Endpoints de serviço da AWS.
- Para exibir a lista atual de todas as Regiões e endpoints compatíveis para cada AWS service (Serviço da AWS), consulte Endpoints e cotas de serviço.

Criar clientes de serviço

Para acessar programaticamente Serviços da AWS, SDKs use uma classe/objeto cliente para cada um. AWS service (Serviço da AWS) Se seu aplicativo precisar acessar a AmazonEC2, por exemplo, seu aplicativo criará um objeto EC2 cliente da Amazon para interagir com esse serviço.

Se nenhuma região for especificada explicitamente para o cliente no próprio código, o cliente usará como padrão a região definida por meio da configuração a seguir. `region` No entanto, a Região ativa de um cliente pode ser definida explicitamente para qualquer objeto de cliente individual. Definir a Região desta maneira tem precedência sobre qualquer configuração global para aquele cliente de serviço particular. A região alternativa é especificada durante a instanciação desse cliente, específica para o seu SDK (consulte seu SDK Guia específico ou sua base SDK de código).

Configure essa funcionalidade usando o seguinte:

`region`- configuração de AWS **`config`** arquivo compartilhado, **`AWS_REGION`**: variável de ambiente, **`aws.region`**- propriedade JVM do sistema: somente Java/Kotlin

Especifica o padrão Região da AWS a ser usado para AWS solicitações. Essa região é usada para solicitações SDK de serviço que não são fornecidas com uma região específica para uso.

Valor padrão: Nenhum. Você deve especificar esse valor explicitamente.

Valores válidos:

- Qualquer um dos códigos de Região disponíveis para o serviço escolhido, conforme listado em Endpoints de serviço da AWS na Referência geral da AWS . Por exemplo, o valor `us-east-1` define o endpoint para a Região da AWS Leste dos EUA (Norte da Virgínia).
- `aws-global`especifica o endpoint global para serviços que oferecem suporte a um endpoint global separado, além dos endpoints regionais, como AWS Security Token Service ()AWS STS e Amazon Simple Storage Service (Amazon S3).

Exemplo de configuração desse valor no arquivo `config`:

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

Exemplo de configuração de variáveis de ambiente para Linux/macOS por meio da linha de comando:

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

Exemplo do Windows de configuração de variáveis de ambiente por meio da linha de comando:

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

A maioria SDKs tem um objeto de “configuração” que está disponível para definir a região padrão a partir do código do aplicativo. Para obter detalhes, consulte seu guia específico para AWS SDK desenvolvedores.

Compatibilidade com AWS SDKs

Os itens a seguir SDKs oferecem suporte aos recursos e configurações descritos neste tópico. Quaisquer exceções parciais estão anotadas. Todas as configurações de propriedade do JVM sistema são suportadas pelo AWS SDK for Java e pelo AWS SDK para Kotlin único.

SDK	C l	Notas ou mais informações
AWS CLI v2	Sim	AWS CLI v2 usa qualquer valor em `AWS_REGION` antes de qualquer valor em `AWS_DEFAULT_REGION` (ambas as variáveis são verificadas).
AWS CLI v1	Sim	AWS CLI v1 usa uma variável de ambiente nomeada `AWS_DEFAULT_REGION` para essa finalidade.
SDKpara C++	Sim	
SDKpara Go V2 (1.x)	Sim	
SDKpara Go 1.x (V1)	Sim	Para usar as configurações do arquivo `config` compartil hado, você deve ativar o carregamento do arquivo de configuração; consulte Sessões.

SDK	C I	Notas ou mais informações
SDKpara Java 2.x	Sim	
SDKpara Java 1.x	Sim	
SDKpara JavaScript 3.x	Sim	
SDKpara JavaScript 2.x	Sim	
SDKpara Kotlin	Sim	
SDKpara. NET3.x	Sim	
SDKpara PHP 3.x	Sim	
SDKpara Python (Boto3)	Sim	Isso SDK usa uma variável de ambiente nomeada `AWS_DEFAULT_REGION` para essa finalidade.
SDKpara Ruby 3.x	Sim	
SDKpara Rust	Sim	
SDKpara Swift	Sim	
Ferramentas para PowerShel l	Sim	

AWS STS Endpoints regionais

AWS Security Token Service (AWS STS) está disponível como um serviço global e regional. Alguns CLIs usam o endpoint de AWS SDKs serviço global (`https://sts.amazonaws.com`) por padrão, enquanto outros usam os endpoints de serviço regional ()`https://sts.`*`{region_identifier}`*`.`*`{partition_domain}`*. As solicitações globais são mapeadas para a região Leste dos EUA (Norte da Virgínia). Para obter mais informações sobre AWS STS endpoints, consulte Endpoints na AWS Security Token Service API referência. Ou aprenda a gerenciar AWS STS em um Região da AWS no Guia do AWS Identity and Access Management usuário.

É uma prática AWS recomendada usar endpoints regionais sempre que possível e configurar seusRegião da AWS. Clientes em partições que não sejam comerciais devem usar endpoints regionais. Nem todas SDKs as ferramentas oferecem suporte a essa configuração, mas todas têm um comportamento definido em relação aos endpoints globais e regionais. Consulte a seção a seguir para obter mais informações.

Para SDKs as ferramentas que oferecem suporte a essa configuração, os clientes podem configurar a funcionalidade usando o seguinte:

`sts_regional_endpoints`- configuração de AWS **`config`** arquivo compartilhado,
`AWS_STS_REGIONAL_ENDPOINTS`: variável de ambiente

Essa configuração especifica como a ferramenta SDK or determina o AWS service (Serviço da AWS) endpoint que ela usa para se comunicar com o AWS Security Token Service ()AWS STS.

Valor padrão: `legacy`

> **Note**
>
> Todas as novas versões SDK principais lançadas após julho de 2022 terão como padrão. `regional` As novas versões SDK principais podem remover essa configuração e usar o `regional` comportamento. Para reduzir o impacto futuro em relação a essa alteração, recomendamos que você comece a usar `regional` em seu aplicativo sempre que possível.

Valores válidos: (Valor recomendado: *`regional`*)

- **`legacy`**— Usa o AWS STS endpoint global,`sts.amazonaws.com`.
- **`regional`**— A ferramenta SDK or sempre usa o AWS STS endpoint da região atualmente configurada. Por exemplo, se o cliente estiver configurado para usar`us-west-2`, todas as chamadas AWS STS serão feitas para o endpoint regional`sts.us-west-2.amazonaws.com`, em vez do `sts.amazonaws.com` endpoint global. Para enviar uma solicitação para o endpoint global enquanto a configuração é habilitada, você pode definir a Região como `aws-global`.

Exemplo de configuração desses valores no arquivo `config`:

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

Exemplo de configuração de variáveis de ambiente para Linux/macOS por meio da linha de comando:

```
export AWS_STS_REGIONAL_ENDPOINTS=regional
```

Exemplo do Windows de configuração de variáveis de ambiente por meio da linha de comando:

```
setx AWS_STS_REGIONAL_ENDPOINTS regional
```

Compatibilidade com AWS SDKs

> **Note**
>
> É uma prática AWS recomendada usar endpoints regionais sempre que possível e configurar seusRegião da AWS.

A tabela a seguir resume, para sua ferramenta SDK ou ferramenta:

- Configuração de suporte: se a variável de `config` arquivo compartilhado e a variável de ambiente para endpoints STS regionais são suportadas.
- Valor da configuração padrão: o valor padrão da configuração, se ela for suportada.
- STSPonto final de destino do cliente de serviço padrão: qual endpoint padrão é usado pelo cliente, mesmo que a configuração para alterá-lo não esteja disponível.
- Comportamento de fallback do cliente de serviço: o que ele SDK faz quando deveria usar um endpoint regional, mas nenhuma região foi configurada. Esse é o comportamento, independentemente de ele estar usando um endpoint regional por causa de um padrão ou porque `regional` foi selecionado pela configuração.

A tabela também usa os seguintes valores:

- Ponto final global:`https://sts.amazonaws.com`.
- Endpoint regional: com base na configuração Região da AWS usada pelo seu aplicativo.
- **`us-east-1`**(Regional): usa o endpoint da `us-east-1` região, mas com tokens de sessão mais longos do que as solicitações globais típicas.

SDK	Valor de configuração padrão	STSEndpoint de destino do cliente de serviço padrão	Comportamento alternativo do cliente de serviço	Notas ou mais informações
AWS CLI v2	N/D	Endpoint regional	Endpoint global	
AWS CLI v1	`legacy`	Endpoint global	Endpoint global	
SDKpara C++	N/D	Endpoint regional	`us-east-1` (Regional)	
SDKpara Go V2 (1.x)	N/D	Endpoint regional	Falha na solicitação	
SDKpara Go 1.x (V1)	`legacy`	Endpoint global	Endpoint global	Para usar as configurações do arquivo `config` compartilhado, você deve ativar o carregamento do arquivo de configuração; consulte Sessões.
SDKpara Java 2.x	N/D	Endpoint regional	Falha na solicitação	Se nenhuma região estiver configurada, o `AssumeRole` e `AssumeRoleWithWebIdentity` usará o STS endpoint global
SDKpara Java 1.x	`legacy`	Endpoint global	Endpoint global	
SDKpara JavaScript 3.x	N/D	Endpoint regional	Falha na solicitação	

SDK	Valor de configuração padrão	STSEndpoint de destino do cliente de serviço padrão	Comportamento alternativo do cliente de serviço	Notas ou mais informações
SDKpara JavaScript 2.x	`legacy`	Endpoint global	Endpoint global	
SDKpara Kotlin	N/D	Endpoint regional	Endpoint global	
SDKpara. NET3.x	`legacy`	Endpoint global	Endpoint global	
SDKpara PHP 3.x	`legacy`	Endpoint global	Falha na solicitação	
SDKpara Python (Boto3)	`legacy`	Endpoint global	Endpoint global	
SDKpara Ruby 3.x	`regional`	Endpoint regional	Falha na solicitação	
SDKpara Rust	N/D	Endpoint regional	Falha na solicitação	
SDKpara Swift	N/D	Endpoint regional	Falha na solicitação	
Ferramentas para PowerShell	`legacy`	Endpoint global	Endpoint global	

Pilha dupla e endpoints FIPS

Configure essa funcionalidade usando o seguinte:

`use_dualstack_endpoint`- compartilhado AWS **`config`**configuração de arquivo, **`AWS_USE_DUALSTACK_ENDPOINT`**: variável de ambiente, **`aws.useDualstackEndpoint`**- propriedade JVM do sistema: somente Java/Kotlin

Ativa ou desativa se o SDK enviará solicitações para endpoints de pilha dupla. Para saber mais sobre endpoints de pilha dupla, que oferecem suporte tanto ao tráfego quanto ao IPv6 tráfego, consulte Como IPv4 usar endpoints de pilha dupla do Amazon S3 no Guia do usuário do Amazon Simple Storage Service. Endpoints de pilha dupla estão disponíveis para alguns serviços em algumas regiões.

Valor padrão: `false`

Valores válidos:

- **`true`**— A ferramenta SDK or tentará usar endpoints de pilha dupla para fazer solicitações de rede. Se não existir um endpoint de pilha dupla para o serviço e/ou Região da AWS, a solicitação falhará.
- **`false`**— A ferramenta SDK or não usará endpoints de pilha dupla para fazer solicitações de rede.

`use_fips_endpoint`- compartilhado AWS **`config`**configuração de arquivo, **`AWS_USE_FIPS_ENDPOINT`**: variável de ambiente, **`aws.useFipsEndpoint`**- propriedade JVM do sistema: somente Java/Kotlin

Ativa ou desativa se a ferramenta SDK ou enviará solicitações para FIPS endpoints compatíveis. Os Padrões Federais de Processamento de Informações (FIPS) são um conjunto de requisitos de segurança do governo dos EUA para dados e sua criptografia. Agências governamentais, parceiros e aqueles que desejam fazer negócios com o governo federal devem seguir as FIPS diretrizes. Ao contrário do padrão AWS endpoints, os FIPS endpoints usam uma biblioteca de TLS software compatível com 140-2. FIPS Se essa configuração estiver ativada e não existir um FIPS endpoint para o serviço em seu Região da AWS, o AWS a chamada pode falhar. Endpoints específicos de serviçoe a `--endpoint-url` opção pelo AWS Command Line Interface substitua essa configuração.

Para saber mais sobre outras formas de especificar FIPS endpoints por Região da AWS, consulte FIPSEndpoints by Service. Para obter mais informações sobre os endpoints do serviço Amazon

Elastic Compute Cloud, consulte endpoints de pilha dupla (IPv4e) na Amazon IPv6 Reference. EC2 API

Valor padrão: `false`

Valores válidos:

- **`true`**— A ferramenta SDK or enviará solicitações para FIPS endpoints compatíveis.
- **`false`**— A ferramenta SDK or não enviará solicitações para FIPS endpoints compatíveis.

Compatibilidade com AWS SDKs

Os itens a seguir SDKs oferecem suporte aos recursos e configurações descritos neste tópico. Quaisquer exceções parciais estão anotadas. Todas as configurações de propriedade do JVM sistema são suportadas pelo AWS SDK for Java e o AWS SDK para Kotlin somente.

SDK	C l	Notas ou mais informações
AWS CLI v2	Sim	
SDKpara C++	Sim	
SDKpara Go V2 (1.x)	Sim	
SDKpara Go 1.x (V1)	Sim	Para usar as configurações do arquivo `config` compartil hado, você deve ativar o carregamento do arquivo de configuração; consulte Sessões.
SDKpara Java 2.x	Sim	
SDKpara Java 1.x	Não	
SDKpara JavaScript 3.x	Sim	
SDKpara JavaScript 2.x	Sim	
SDKpara Kotlin	Sim	
SDKpara. NET3.x	Sim	

SDK	C l	Notas ou mais informações
SDKpara PHP 3.x	Sim	
SDKpara Python (Boto3)	Sim	
SDKpara Ruby 3.x	Sim	
SDKpara Rust	Sim	
SDKpara Swift	Sim	
Ferramentas para PowerShel l	Sim	

Descoberta de endpoint

SDKsuse a descoberta de endpoints para acessar os endpoints de serviço (URLspara acessar vários recursos), mantendo a flexibilidade para AWS para alterar URLs conforme necessário. Dessa forma, seu código pode detectar automaticamente novos endpoints. Não há endpoints fixos para alguns serviços. Em vez disso, você obtém os endpoints disponíveis durante o runtime fazendo uma solicitação para obter os endpoints primeiro. Depois de recuperar os endpoints disponíveis, o código usa o endpoint para acessar outras operações. Por exemplo, para o Amazon Timestream, ele faz `DescribeEndpoints` uma solicitação para recuperar SDK os endpoints disponíveis e, em seguida, usa esses endpoints para concluir operações específicas, como ou. `CreateDatabase` `CreateTable`

Configure essa funcionalidade usando o seguinte:

`endpoint_discovery_enabled`- compartilhado AWS **`config`**configuração de arquivo, **`AWS_ENABLE_ENDPOINT_DISCOVERY`**: variável de ambiente, **`aws.endpointDiscoveryEnabled`**- propriedade JVM do sistema: somente Java/Kotlin, Para configurar o valor diretamente no código, consulte seu específico SDK diretamente.

Ativa ou desativa a descoberta de endpoints para o DynamoDB.

A descoberta de endpoints é necessária no Timestream e opcional no Amazon DynamoDB. Essa configuração é padronizada `true` ou `false` depende de o serviço exigir a descoberta

do endpoint. As solicitações de Timestream são padronizadas para`true`, e as solicitações do Amazon DynamoDB, como padrão. `false`

Valores válidos:

- **`true`**— Eles SDK devem tentar descobrir automaticamente um endpoint para serviços em que a descoberta de endpoint é opcional.
- **`false`**— Eles não SDK devem tentar descobrir automaticamente um endpoint para serviços em que a descoberta de endpoint é opcional.

Compatibilidade com AWS SDKs

Os itens a seguir SDKs oferecem suporte aos recursos e configurações descritos neste tópico. Quaisquer exceções parciais estão anotadas. Todas as configurações de propriedade do JVM sistema são suportadas pelo AWS SDK for Java e o AWS SDK para Kotlin somente.

SDK	C l	Notas ou mais informações
AWS CLI v2	Sim	
SDKpara C++	Sim	
SDKpara Go V2 (1.x)	Sim	
SDKpara Go 1.x (V1)	Sim	Para usar as configurações do arquivo `config` compartil hado, você deve ativar o carregamento do arquivo de configuração; consulte Sessões.
SDKpara Java 2.x	Sim	O SDK for Java 2.x usa `AWS_ENDPOINT_DISCO VERY_ENABLED` para o nome da variável de ambiente.
SDKpara Java 1.x	Parci	JVMpropriedade do sistema não suportada.
SDKpara JavaScript 3.x	Sim	
SDKpara JavaScript 2.x	Sim	
SDKpara Kotlin	Sim	

SDK	C I	Notas ou mais informações
SDKpara. NET3.x	Sim	
SDKpara PHP 3.x	Sim	
SDKpara Python (Boto3)	Sim	
SDKpara Ruby 3.x	Sim	
SDKpara Rust	Parci	Compatível somente com Timestream.
SDKpara Swift	Não	
Ferramentas para PowerShell	Sim	

Definições gerais da configuração

SDKssuportam algumas configurações gerais que definem SDK comportamentos gerais.

Configure essa funcionalidade usando o seguinte:

`api_versions`- compartilhado AWS **`config`**configuração de arquivo

Alguns AWS os serviços mantêm várias API versões para oferecer suporte à compatibilidade com versões anteriores. Por padrão, SDK e AWS CLI as operações usam a API versão mais recente disponível. Para exigir uma API versão específica para usar em suas solicitações, inclua a `api_versions` configuração em seu perfil.

Valor padrão: Nenhum. (A API versão mais recente é usada peloSDK.)

Valores válidos: essa é uma configuração aninhada seguida por uma ou mais linhas recuadas, cada uma identificando uma AWS serviço e a API versão a ser usada. Veja a documentação do AWS serviço para entender quais API versões estão disponíveis.

O exemplo define uma API versão específica para dois AWS serviços no `config` arquivo. Essas API versões são usadas somente para comandos executados sob o perfil que contém essas

configurações. Os comandos para qualquer outro serviço usam a versão mais recente desse serviçoAPI.

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

`ca_bundle`- compartilhado AWS **`config`**configuração de arquivo, **`AWS_CA_BUNDLE`**: variável de ambiente

Especifica o caminho para um pacote de certificados personalizado (um arquivo com uma `.pem` extensão) a ser usado ao estabelecer conexões SSL TLS /.

Valor padrão: nenhum

Valores válidos: especifique o caminho completo ou o nome do arquivo base. Se houver um nome de arquivo base, o sistema tentará encontrar o programa nas pastas especificadas pela variável de ambiente PATH.

Exemplo de configuração desse valor no arquivo `config`:

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

Devido às diferenças na forma como os sistemas operacionais manipulam caminhos e escapam de caracteres de caminho, o seguinte é um exemplo de como definir esse valor no `config` arquivo no Windows:

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

Exemplo de configuração de variáveis de ambiente para Linux/macOS por meio da linha de comando:

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

Exemplo do Windows de configuração de variáveis de ambiente por meio da linha de comando:

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

`output`- compartilhado AWS **`config`**configuração de arquivo

Especifica como os resultados são formatados no AWS CLI e outros AWS SDKse ferramentas.

Valor padrão: `json`

Valores válidos:

- **`json`**— A saída é formatada como uma JSONstring.
- **`yaml`**— A saída é formatada como uma YAMLstring.
- **`yaml-stream`**— A saída é transmitida e formatada como uma YAMLstring. A transmissão possibilita um manuseio mais rápido de tipos de dados grandes.
- **`text`**: a saída é formatada como várias linhas de valores de string separados por tabulação. Isso pode ser útil para passar a saída para um processador de texto, como `grep`, `sed` ou `awk`.
- **`table`**: a saída é formatada como uma tabela usando os caracteres +|- para formar as bordas da célula. Geralmente, a informação é apresentada em um formato "amigável", que é muito mais fácil de ler do que outros, mas não tão útil programaticamente.

`parameter_validation`- compartilhado AWS **`config`**configuração de arquivo

Especifica se a ferramenta SDK ou tenta validar os parâmetros da linha de comando antes de enviá-los para o AWS ponto final do serviço.

Valor padrão: `true`

Valores válidos:

- **`true`** – O padrão. A ferramenta SDK or executa a validação dos parâmetros da linha de comando no lado do cliente. Isso ajuda a ferramenta SDK or a confirmar que os parâmetros são válidos e detecta alguns erros. A ferramenta SDK or pode rejeitar solicitações que não são válidas antes de enviar solicitações para o AWS ponto final do serviço.
- **`false`**— A ferramenta SDK or não valida os parâmetros da linha de comando antes de enviá-los para o AWS ponto final do serviço. A ferramenta AWS o endpoint de serviço é responsável por validar todas as solicitações e rejeitar solicitações que não são válidas.

Compatibilidade com AWS SDKs

Os itens a seguir SDKs oferecem suporte aos recursos e configurações descritos neste tópico. Quaisquer exceções parciais estão anotadas. Todas as configurações de propriedade do JVM sistema são suportadas pelo AWS SDK for Java e o AWS SDK para Kotlin somente.

SDK	C l	Notas ou mais informações
AWS CLI v2	Parci	`api_versions` incompatível.
SDKpara C++	Sim	
SDKpara Go V2 (1.x)	Parci	`api_versions` e `parameter_validation` não são compatíveis.
SDKpara Go 1.x (V1)	Parci	`api_versions` e `parameter_validation` não são compatíveis. Para usar as configurações `config` do arquivo compartilhado, você deve ativar o carregamento do arquivo de configuração; consulte Sessões.
SDKpara Java 2.x	Não	
SDKpara Java 1.x	Não	
SDKpara JavaScript 3.x	Sim	
SDKpara JavaScript 2.x	Sim	
SDKpara Kotlin	Não	
SDKpara. NET3.x	Não	
SDKpara PHP 3.x	Sim	
SDKpara Python (Boto3)	Sim	
SDKpara Ruby 3.x	Sim	
SDKpara Rust	Não	
SDKpara Swift	Não	
Ferramentas para PowerShel l	Não	

IMDScliente

SDKsimplemente um cliente do Instance Metadata Service versão 2 (IMDSv2) usando solicitações orientadas à sessão. Para obter mais informações sobreIMDSv2, consulte Use IMDSv2 no Guia do EC2 usuário da Amazon. O IMDS cliente é configurável por meio de um objeto de configuração do cliente disponível na base de SDK código.

Configure essa funcionalidade usando o seguinte:

`retries`: membro do objeto de configuração do cliente

O número de tentativas adicionais para qualquer solicitação com falha.

Valor padrão: 3

Valores válidos: número maior que zero.

`port`: membro do objeto de configuração do cliente

A porta para o endpoint.

Valor padrão: 80

Valores válidos: número.

`token_ttl`: membro do objeto de configuração do cliente

O TTL do token.

Valor padrão: 21.600 segundos (6 horas, o tempo máximo alocado).

Valores válidos: número.

`endpoint`: membro do objeto de configuração do cliente

O ponto final de. IMDS

Valor padrão: se `endpoint_mode` for igual a `IPv4`, o endpoint padrão será `http://169.254.169.254`. Se `endpoint_mode` for igual a `IPv6`, o endpoint padrão será `http://[fd00:ec2::254]`.

Valores válidos: válidosURI.

As opções a seguir são suportadas pela maioriaSDKs. Consulte sua base de SDK código específica para obter detalhes.

`endpoint_mode`: membro do objeto de configuração do cliente

O modo de endpoint do. IMDS

Valor padrão: `IPv4`

Valores válidos: `IPv4`, `IPv6`

`http_open_timeout`: membro do objeto de configuração do cliente (o nome pode variar)

O número de segundos a aguardar até que a conexão seja aberta.

Valor padrão: 1 segundo.

Valores válidos: número maior que zero.

`http_read_timeout`: membro do objeto de configuração do cliente (o nome pode variar)

O número de segundos para que um bloco de dados seja lido.

Valor padrão: 1 segundo.

Valores válidos: número maior que zero.

`http_debug_output`: membro do objeto de configuração do cliente (o nome pode variar)

Define um fluxo de saída para depuração.

Valor padrão: nenhum.

Valores válidos: um fluxo de E/S válido, comoSTDOUT.

`backoff`: membro do objeto de configuração do cliente (o nome pode variar)

O número de segundos para dormir entre as novas tentativas ou o cliente forneceu a função de desligamento para chamar. Isto substitui a estratégia padrão de recuo exponencial.

Valor padrão: varia de acordo comSDK.

Valores válidos: varia de acordo comSDK. Pode ser um valor numérico ou uma chamada para uma função personalizada.

Compatibilidade com AWS SDKs

Os itens a seguir SDKs oferecem suporte aos recursos e configurações descritos neste tópico. Quaisquer exceções parciais estão anotadas. Todas as configurações de propriedade do JVM sistema são suportadas pelo AWS SDK for Java e pelo AWS SDK para Kotlin único.

SDK	C l	Notas ou mais informações
AWS CLI v2	Sim	
SDKpara C++	Não	
SDKpara Go V2 (1.x)	Sim	
SDKpara Go 1.x (V1)	Sim	
SDKpara Java 2.x	Sim	
SDKpara Java 1.x	Sim	
SDKpara JavaScript 3.x	Sim	
SDKpara JavaScript 2.x	Sim	
SDKpara Kotlin	Não	
SDKpara. NET3.x	Sim	
SDKpara PHP 3.x	Sim	
SDKpara Python (Boto3)	Sim	
SDKpara Ruby 3.x	Sim	
SDKpara Rust	Sim	
SDKpara Swift	Sim	
Ferramentas para PowerShel l	Sim	

Comportamento de repetição

O comportamento de repetição inclui configurações sobre como a SDKs tentativa de se recuperar de falhas resultantes de solicitações feitas para Serviços da AWS.

Configure essa funcionalidade usando o seguinte:

`retry_mode`- compartilhado AWS **`config`**configuração de arquivo, **`AWS_RETRY_MODE`**: variável de ambiente, **`aws.retryMode`**- propriedade JVM do sistema: somente Java/Kotlin

Especifica como a ferramenta SDK ou o desenvolvedor tenta novas tentativas.

Valor padrão: esse valor é específico para o seuSDK. Verifique seu SDK guia específico ou sua base SDK de código para ver o padrão`retry_mode`.

Valores válidos:

- `standard`— (Recomendado) O conjunto recomendado de regras de repetição em AWS SDKs. Esse modo inclui um conjunto padrão de erros que são repetidos e ajusta automaticamente o número de novas tentativas para maximizar a disponibilidade e a estabilidade. Esse modo é seguro para uso em aplicativos multilocatários. O número máximo padrão de tentativas com esse modo é três, a menos que `max_attempts` esteja explicitamente configurado.
- `adaptive`— Um modo de repetição, apropriado somente para casos de uso especializados, que inclui a funcionalidade do modo padrão, bem como a limitação automática de taxa do lado do cliente. Esse modo de repetição não é recomendado para aplicativos multilocatários, a menos que você tenha o cuidado de isolar os inquilinos do aplicativo. Consulte Escolher entre os `standard` modos e `adaptive` tentar novamente Para mais informações. Esse modo é experimental e pode mudar o comportamento no futuro.
- `legacy`— (Não recomendado) Específico para você SDK (verifique seu SDK guia específico ou sua base SDK de código).

`max_attempts`- compartilhado AWS **`config`**configuração de arquivo, **`AWS_MAX_ATTEMPTS`**: variável de ambiente, **`aws.maxAttempts`**- propriedade JVM do sistema: somente Java/Kotlin

Especifica o número máximo de tentativas a serem feitas em uma solicitação.

Valor padrão: se esse valor não for especificado, seu padrão dependerá do valor da configuração `retry_mode`:

- Se `retry_mode` for `legacy` — Usa um valor padrão específico para seu SDK (verifique seu SDK guia específico ou sua base SDK de código para ver o `max_attempts` padrão).

- Se `retry_mode` for `standard`: faz três tentativas.
- Se `retry_mode` for `adaptive`: faz três tentativas.

Valores válidos: número maior que zero.

Escolher entre os **`standard`** modos e **`adaptive`** tentar novamente

Recomendamos que você use o modo de `standard` repetição, a menos que tenha certeza de que seu uso é mais adequado`adaptive`.

> **Note**
>
> O `adaptive` modo pressupõe que você esteja agrupando clientes com base no escopo no qual o serviço de back-end pode limitar as solicitações. Se você não fizer isso, as limitações em um recurso podem atrasar as solicitações de um recurso não relacionado se você estiver usando o mesmo cliente para os dois recursos.

Padrão	Adaptável
Casos de uso de aplicativos: todos.	Casos de uso de aplicativos: 1. Não é sensível à latência. 2. O cliente acessa apenas um único recurso, ou você está fornecendo lógica para agrupar seus clientes separadamente pelo recurso de serviço que está sendo acessado.
Suporta interrupção de circuito para evitar que eles tentem novamente durante SDK interrupç ões.	Suporta interrupção de circuito para evitar que eles tentem novamente durante SDK interrupç ões.
Usa um recuo exponencial instável em caso de falhas.	Usa durações dinâmicas de recuo para tentar minimizar o número de solicitações com falha, em troca do potencial de maior latência.
Nunca atrasa a primeira tentativa de solicitaç ão, somente as novas tentativas.	Pode acelerar ou atrasar a tentativa de solicitaç ão inicial.

Se você optar por usar o `adaptive` modo, seu aplicativo deverá criar clientes projetados com base em cada recurso que possa ser limitado. Um recurso, nesse caso, é mais refinado do que apenas pensar em cada um AWS service (Serviço da AWS). Serviços da AWS podem ter dimensões adicionais que eles usam para acelerar as solicitações. Vamos usar o serviço Amazon DynamoDB como exemplo. O DynamoDB usa Região da AWS além da tabela que está sendo acessada para acelerar as solicitações. Isso significa que uma tabela que seu código está acessando pode ser mais limitada do que outras. Se seu código usou o mesmo cliente para acessar todas as tabelas e as solicitações para uma dessas tabelas forem limitadas, o modo de repetição adaptável reduzirá a taxa de solicitação de todas as tabelas. Seu código deve ser projetado para ter um cliente por egion-and-table par R. Se você tiver uma latência inesperada ao usar o `adaptive` modo, consulte o específico AWS guia de documentação para o serviço que você está usando.

Detalhes da implementação do modo de repetição

A ferramenta AWS SDKsuse repositórios de tokens para decidir se uma solicitação deve ser repetida e (no caso do modo de `adaptive` repetição) com que rapidez as solicitações devem ser enviadas. Dois repositórios de tokens são usados peloSDK: um repositório de tokens de nova tentativa e um repositório de tokens de taxa de solicitação.

- O repositório de tokens de repetição é usado para determinar se as tentativas SDK devem ser temporariamente desativadas para proteger os serviços upstream e downstream durante interrupções. Os tokens são adquiridos do bucket antes da tentativa de novas tentativas, e os tokens são devolvidos ao bucket quando as solicitações são bem-sucedidas. Se o bucket estiver vazio quando uma nova tentativa for tentada, SDK ele não repetirá a solicitação.
- O repositório de tokens da taxa de solicitação é usado somente no modo de nova `adaptive` tentativa para determinar a taxa na qual enviar solicitações. Os tokens são adquiridos do bucket antes do envio de uma solicitação, e os tokens são devolvidos ao bucket a uma taxa determinada dinamicamente com base nas respostas de limitação retornadas pelo serviço.

A seguir está o pseudocódigo de alto nível para os modos de repetição de repetição `standard` e `adaptive`:

```
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)
  }
}
```

A seguir estão mais detalhes sobre os componentes usados no pseudocódigo:

`GetSendToken`:

Essa etapa é usada somente no modo de `adaptive` repetição. Essa etapa adquire um token do repositório de tokens da taxa de solicitação. Se um token não estiver disponível, ele aguardará até que um fique disponível. Você SDK pode ter opções de configuração disponíveis para falhar na solicitação em vez de esperar. Os tokens no bucket são recarregados a uma taxa determinada dinamicamente, com base no número de respostas de limitação recebidas pelo cliente.

`SendHTTPRequest`:

Essa etapa envia a solicitação para AWS. A maioria AWS SDKsuse uma HTTP biblioteca que usa grupos de conexões para reutilizar uma conexão existente ao fazer uma HTTP solicitação. Geralmente, as conexões são reutilizadas se uma solicitação falhar devido a erros de limitação, mas não se uma solicitação falhar devido a um erro transitório.

`RequestBookkeeping`:

Os tokens são adicionados ao token bucket se a solicitação for bem-sucedida. Somente para o modo de `adaptive` repetição, a taxa de preenchimento do bucket de tokens da taxa de solicitação é atualizada com base no tipo de resposta recebida.

`Retryable`:

Essa etapa determina se uma resposta pode ser repetida com base no seguinte:

- O código de HTTP status.
- O código de erro retornado do serviço.

- Erros de conexão, definidos como qualquer erro recebido pelo SDK no qual uma HTTP resposta do serviço não é recebida.

Erros transitórios (códigos de HTTP status 400, 408, 500, 502, 503 e 504) e erros de limitação (códigos de HTTP status 400, 403, 429, 502, 503 e 509) podem ser tentados novamente. SDKo comportamento de repetição é determinado em combinação com códigos de erro ou outros dados do serviço.

`MAX_ATTEMPTS`:

O número padrão de tentativas máximas é definido pela `retry_mode` configuração, a menos que seja substituído pela `max_attempts` configuração.

`HasRetryQuota`

Essa etapa adquire um token do repositório de tokens de repetição. Se o repositório de tokens de nova tentativa estiver vazio, a solicitação não será repetida.

`ExponentialBackoff`

Para um erro que pode ser repetido, o atraso da nova tentativa é calculado usando o recuo exponencial truncado. O SDKs uso de recuo exponencial binário truncado com instabilidade. O algoritmo a seguir mostra como a quantidade de tempo de sono, em segundos, é definida para uma resposta à solicitação `i`:

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

No algoritmo anterior, os seguintes valores se aplicam:

`b = random number within the range of: 0 <= b <= 1`

`r = 2`

`MAX_BACKOFF = 20 seconds`para a maioriaSDKs. Consulte seu SDK guia específico ou código-fonte para confirmação.

Compatibilidade com AWS SDKs

Os itens a seguir SDKs oferecem suporte aos recursos e configurações descritos neste tópico. Quaisquer exceções parciais estão anotadas. Todas as configurações de propriedade do JVM sistema são suportadas pelo AWS SDK for Java e o AWS SDK para Kotlin somente.

SDK	C l	Notas ou mais informações
AWS CLI v2	Sim	
SDKpara C++	Sim	
SDKpara Go V2 (1.x)	Sim	
SDKpara Go 1.x (V1)	Não	
SDKpara Java 2.x	Sim	
SDKpara Java 1.x	Sim	JVMpropriedades do sistema: use `com.amazonaws.sdk.maxAttempts` em vez de `aws.maxAttempts` ; use `com.amazonaws.sdk.retryMode` em vez de `aws.retryMode` .
SDKpara JavaScript 3.x	Sim	
SDKpara JavaScript 2.x	Não	Suporta um número máximo de novas tentativas, recuo exponencial com instabilidade e a opção de um método personalizado para recuar novamente.
SDKpara Kotlin	Sim	
SDKpara. NET3.x	Sim	
SDKpara PHP 3.x	Sim	
SDKpara Python (Boto3)	Sim	
SDKpara Ruby 3.x	Sim	
SDKpara Rust	Sim	
SDKpara Swift	Sim	
Ferramentas para PowerShell	Sim	

Compactação de solicitações

> **Note**
>
> Para obter ajuda na compreensão do layout das páginas de configurações ou na interpretação da Compatibilidade com AWS SDKstabela a seguir, vejaPáginas de configurações.

AWS SDKse as ferramentas podem compactar cargas automaticamente ao enviar solicitações para Serviços da AWS que suportam o recebimento de cargas comprimidas. Compactar a carga útil do cliente antes de enviá-la para um serviço pode reduzir o número geral de solicitações e a largura de banda necessárias para enviar dados ao serviço, bem como reduzir as solicitações malsucedidas devido às limitações do serviço no tamanho da carga útil. Para compactação, a ferramenta SDK or seleciona um algoritmo de codificação compatível com o serviço e o. SDK No entanto, a lista atual de codificações possíveis consiste apenas em gzip, mas pode se expandir no futuro.

A compactação de solicitações pode ser especialmente útil se seu aplicativo estiver usando a Amazon CloudWatch. CloudWatch é um serviço de monitoramento e observabilidade que coleta dados operacionais e de monitoramento na forma de registros, métricas e eventos. Um exemplo de operação de serviço que suporta compressão CloudWatch é o PutMetricDataAPImétodo.

Configure essa funcionalidade usando o seguinte:

`disable_request_compression`- compartilhado AWS **`config`**configuração de arquivo, **`AWS_DISABLE_REQUEST_COMPRESSION`**: variável de ambiente, **`aws.disableRequestCompression`**- propriedade JVM do sistema: somente Java/Kotlin

Ativa ou desativa se a ferramenta SDK or comprimirá uma carga antes de enviar uma solicitação.

Valor padrão: `false`

Valores válidos:

- **`true`** – Desative a compactação de solicitações.
- **`false`** – Use a compactação de solicitações quando possível.

`request_min_compression_size_bytes`- compartilhado AWS **`config`**configuração de arquivo, **`AWS_REQUEST_MIN_COMPRESSION_SIZE_BYTES`**: variável de ambiente, **`aws.requestMinCompressionSizeBytes`**- propriedade JVM do sistema: somente Java/Kotlin

Define o tamanho mínimo em bytes do corpo da solicitação que a ferramenta SDK or deve compactar. Cargas pequenas podem ficar maiores quando compactadas, portanto, há um limite mínimo em que faz sentido realizar a compactação. Esse valor é inclusivo, um tamanho de solicitação maior que ou igual ao valor é compactado.

Valor padrão: 10240 bytes

Valores válidos: valor inteiro entre 0 e 10485760 bytes, inclusive.

Compatibilidade com AWS SDKs

Os itens a seguir SDKs oferecem suporte aos recursos e configurações descritos neste tópico. Quaisquer exceções parciais estão anotadas. Todas as configurações de propriedade do JVM sistema são suportadas pelo AWS SDK for Java e o AWS SDK para Kotlin somente.

SDK	C l	Notas ou mais informações
AWS CLI v2	Sim	
SDKpara C++	Sim	
SDKpara Go V2 (1.x)	Sim	
SDKpara Go 1.x (V1)	Não	
SDKpara Java 2.x	Sim	
SDKpara Java 1.x	Não	
SDKpara JavaScript 3.x	Sim	
SDKpara JavaScript 2.x	Não	
SDKpara Kotlin	Sim	

SDK	C l	Notas ou mais informações
SDKpara. NET3.x	Sim	
SDKpara PHP 3.x	Sim	
SDKpara Python (Boto3)	Sim	
SDKpara Ruby 3.x	Sim	
SDKpara Rust	Sim	
SDKpara Swift	Não	
Ferramentas para PowerShel l	Sim	

Endpoints específicos de serviço

A configuração de endpoint específico do serviço oferece a opção de usar um endpoint de sua escolha para API solicitações e fazer com que essa escolha persista. Essas configurações fornecem flexibilidade para oferecer suporte a endpoints locais, VPC endpoints e locais de terceiros AWS ambientes de desenvolvimento. Diferentes endpoints podem ser usados para ambientes de teste e produção. Você pode especificar um endpoint URL para cada pessoa Serviços da AWS.

Configure essa funcionalidade usando o seguinte:

`endpoint_url`- compartilhado AWS **`config`**configuração de arquivo, **`AWS_ENDPOINT_URL`**: variável de ambiente, **`aws.endpointUrl`**- propriedade JVM do sistema: somente Java/Kotlin

> Quando especificada diretamente em um perfil ou como uma variável de ambiente, esta configuração especifica o endpoint usado para todas as solicitações de serviço. Este endpoint é substituído por qualquer endpoint específico do serviço configurado.
>
> Você também pode usar essa configuração em uma `services` seção de um compartilhado AWS `config`arquivo para definir um endpoint personalizado para um serviço específico. Para obter uma lista de todas as chaves de identificação de serviço a serem usadas nas subseções dentro da seção `services`, consulte Identificadores para endpoints específicos de serviço.

Valor padrão: none

Valores válidos: A URL incluindo o esquema e o host do endpoint. Opcionalmente, eles URL podem conter um componente de caminho que contenha um ou mais segmentos de caminho.

AWS_ENDPOINT_URL_<SERVICE>: variável de ambiente, **aws.endpointUrl<ServiceName>**- propriedade JVM do sistema: somente Java/Kotlin

AWS_ENDPOINT_URL_<SERVICE>, onde <SERVICE> está o AWS service (Serviço da AWS) identificador, define um endpoint personalizado para um serviço específico. Para obter uma lista de todas as variáveis de ambiente específicas do serviço, consulte Identificadores para endpoints específicos de serviço.

Este endpoint específico do serviço substitui qualquer endpoint global configurado em AWS_ENDPOINT_URL.

Valor padrão: none

Valores válidos: A URL incluindo o esquema e o host do endpoint. Opcionalmente, eles URL podem conter um componente de caminho que contenha um ou mais segmentos de caminho.

ignore_configured_endpoint_urls- compartilhado AWS **config**configuração de arquivo, **AWS_IGNORE_CONFIGURED_ENDPOINT_URLS**: variável de ambiente, **aws.ignoreConfiguredEndpointUrls**- propriedade JVM do sistema: somente Java/Kotlin

Esta configuração é usada para ignorar todas as configurações personalizadas de endpoints.

Observe que qualquer endpoint explícito definido no código ou no próprio cliente de serviço é usado independentemente desta configuração. Por exemplo, incluindo o parâmetro da linha de --endpoint-url comando com um AWS CLI comandar ou passar um endpoint URL para um construtor de cliente sempre terá efeito.

Valor padrão: false

Valores válidos:

- **true**— A ferramenta SDK or não lê nenhuma opção de configuração personalizada do config arquivo compartilhado ou das variáveis de ambiente para definir um endpointURL.
- **false**— A ferramenta SDK or usa todos os endpoints disponíveis fornecidos pelo usuário a partir do config arquivo compartilhado ou de variáveis de ambiente.

Configurar endpoints usando variáveis de ambiente

Para encaminhar solicitações de todos os serviços para um endpoint personalizadoURL, defina a variável de ambiente `AWS_ENDPOINT_URL` global.

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

Para encaminhar solicitações para um determinado AWS service (Serviço da AWS) para um endpoint personalizadoURL, use a variável de `AWS_ENDPOINT_URL_<SERVICE>` ambiente. Amazon DynamoDB tem um `serviceId` de `DynamoDB`. Para esse serviço, a variável de URL ambiente do endpoint é`AWS_ENDPOINT_URL_DYNAMODB`. Este endpoint tem precedência sobre o endpoint global definido em `AWS_ENDPOINT_URL` para este serviço.

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

Como outro exemplo, AWS Elastic Beanstalk tem um `serviceId` de `Elastic Beanstalk`. A ferramenta AWS service (Serviço da AWS) O identificador é baseado no API modelo, substituindo todos os espaços `serviceId` por sublinhados e colocando todas as letras em maiúsculas. Para configurar o endpoint para este serviço, a variável de ambiente correspondente é `AWS_ENDPOINT_URL_ELASTIC_BEANSTALK`. Para obter uma lista de todas as variáveis de ambiente específicas do serviço, consulte Identificadores para endpoints específicos de serviço.

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

Configurar endpoints usando o arquivo compartilhado **`config`**

No arquivo compartilhado `config`, `endpoint_url` é usado em locais diferentes para diferentes funcionalidades.

- `endpoint_url` especificado diretamente em um `profile` torna esse endpoint no endpoint global.
- `endpoint_url` aninhado sob uma chave identificadora de serviço em uma seção `services`, faz com que esse endpoint se aplique às solicitações feitas somente para esse serviço. Para obter detalhes sobre como definir uma seção `services` no arquivo compartilhado Formato do arquivo de configuração, consulte `config`.

O exemplo a seguir usa uma `services` definição para configurar um endpoint específico de serviço URL a ser usado para o Amazon S3 e um endpoint global personalizado para ser usado para todos os outros serviços:

```
[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
```

Um único perfil pode configurar endpoints para vários serviços. Este exemplo mostra como definir o endpoint específico do serviço para o Amazon URLs S3 e AWS Elastic Beanstalk no mesmo perfil. AWS Elastic Beanstalk tem um `serviceId` de `Elastic Beanstalk`. A ferramenta AWS service (Serviço da AWS) O identificador é baseado no API modelo, substituindo todos os espaços `serviceId` por sublinhados e colocando todas as letras em minúsculas. Assim, a chave identificadora de serviço se torna `elastic_beanstalk` e as configurações deste serviço começam na linha `elastic_beanstalk =` . Para obter uma lista de todas as chaves de identificação de serviço a serem usadas na seção `services`, consulte Identificadores para endpoints específicos de serviço.

```
[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
```

A seção de configuração de serviço pode ser usada a partir de vários perfis. Por exemplo, dois perfis podem usar a mesma definição `services` ao alterar outras propriedades do perfil:

```
[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
```

Configure endpoints em perfis usando credenciais baseadas em funções

Se seu perfil tiver credenciais baseadas em função configuradas por meio de um `source_profile` parâmetro para IAM assumir a funcionalidade de função, ele usará SDK somente configurações de serviço para o perfil especificado. Ele não usa perfis com funções vinculadas a ele. Por exemplo, usando o seguinte arquivo compartilhado `config`:

```
[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
```

Se você usar o perfil `B` e fizer uma chamada em seu código para a AmazonEC2, o endpoint será resolvido como. `https://profile-b-ec2-endpoint.aws` Se o seu código fizer uma solicitação para qualquer outro serviço, a resolução do endpoint não seguirá nenhuma lógica personalizada. O endpoint não é resolvido para o endpoint global definido no perfil `A`. Para que um endpoint global tenha efeito para o perfil `B`, você precisaria configurar `endpoint_url` diretamente no perfil `B`. Para obter mais informações sobre a configuração `source_profile`, consulte Assuma o perfil de provedor de credenciais.

Precedência de configurações

As configurações deste atributo podem ser usadas ao mesmo tempo, mas somente um valor terá prioridade por serviço. Para API chamadas feitas para um determinado AWS service (Serviço da AWS), a seguinte ordem é usada para selecionar um valor:

1. Qualquer configuração explícita definida no código ou no próprio cliente de serviço tem precedência sobre qualquer outra coisa.
   - Para o AWS CLI, esse é o valor fornecido pelo parâmetro da linha de `--endpoint-url` comando. Para umSDK, as atribuições explícitas podem assumir a forma de um parâmetro que você define ao instanciar um AWS service (Serviço da AWS) cliente ou objeto de configuração.
2. O valor fornecido por uma variável de ambiente específica do serviço, como `AWS_ENDPOINT_URL_DYNAMODB`.
3. O valor fornecido pela variável de ambiente global do endpoint `AWS_ENDPOINT_URL`.
4. O valor fornecido pela configuração `endpoint_url` aninhada em uma chave identificadora de serviço em uma seção `services` do arquivo compartilhado `config`.
5. O valor fornecido pela configuração `endpoint_url` especificado diretamente em um `profile` do arquivo compartilhado `config`.
6. Qualquer endpoint padrão URL para o respectivo AWS service (Serviço da AWS) é usado por último.

Compatibilidade com AWS SDKs

Os itens a seguir SDKs oferecem suporte aos recursos e configurações descritos neste tópico. Quaisquer exceções parciais estão anotadas. Todas as configurações de propriedade do JVM sistema são suportadas pelo AWS SDK for Java e o AWS SDK para Kotlin somente.

SDK	C l	Notas ou mais informações
AWS CLI v2	Sim	
SDKpara C++	Não	
SDKpara Go V2 (1.x)	Sim	
SDKpara Go 1.x (V1)	Não	
SDKpara Java 2.x	Sim	
SDKpara Java 1.x	Não	
SDKpara JavaScript 3.x	Sim	

SDK	C I	Notas ou mais informações
SDKpara JavaScript 2.x	Não	
SDKpara Kotlin	Sim	
SDKpara. NET3.x	Sim	
SDKpara PHP 3.x	Sim	
SDKpara Python (Boto3)	Sim	
SDKpara Ruby 3.x	Sim	
SDKpara Rust	Não	
SDKpara Swift	Não	
Ferramentas para PowerShel l	Sim	

Identificadores para endpoints específicos de serviço

Para obter informações sobre como e onde usar os identificadores na tabela a seguir, consulte Endpoints específicos de serviço.

serviceId	Cl id ac de se pa cc ha AV **cc** fil	**AWS_ENDPOINT_URL_<SERVICE>** variável de ambiente
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: _I	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

serviceId	Cl id ac de se pa cc ha A\ **cc** fil	**AWS_ENDPOINT_URL_<SERVICE>** variável de ambiente
ApiGatewayManagemeentApi	ap yr nt	AWS_ENDPOINT_URL_APIGATEWAYMANAGEMENTAPI
ApiGatewayV2	ap yv	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

serviceId	Cl id ac de se pa cc ha A\ **cc** fil	**AWS_ENDPOINT_URL_<SERVICE>** variável de ambiente
Application Insights	ap or ts	AWS_ENDPOINT_URL_APPLICATION_INSIGHTS
ApplicationCostPro filer	ap or f:	AWS_ENDPOINT_URL_APPLICATIONCOSTPROFILER
App Mesh	ap	AWS_ENDPOINT_URL_APP_MESH
AppRunner	ap	AWS_ENDPOINT_URL_APPRUNNER
AppStream	ap	AWS_ENDPOINT_URL_APPSTREAM
AppSync	ap	AWS_ENDPOINT_URL_APPSYNC
ARC Zonal Shift	a: _s	AWS_ENDPOINT_URL_ARC_ZONAL_SHIFT
Artifact	a:	AWS_ENDPOINT_URL_ARTIFACT
Athena	at	AWS_ENDPOINT_URL_ATHENA
AuditManager	au ge	AWS_ENDPOINT_URL_AUDITMANAGER

serviceId	Cl id ac de se pa cc ha A\ **cc** fil	**AWS_ENDPOINT_URL_<SERVICE>** variável de ambiente
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
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

serviceId	Cl id ac de se pa cc ha A\ **cc** fil	**AWS_ENDPOINT_URL_<SERVICE>** variável de ambiente
Bedrock Agent Runtime	be ge ir	AWS_ENDPOINT_URL_BEDROCK_AGENT_RUNTIME
Bedrock Runtime	be ur	AWS_ENDPOINT_URL_BEDROCK_RUNTIME
billingconductor	b: nc	AWS_ENDPOINT_URL_BILLINGCONDUCTOR
Braket	b:	AWS_ENDPOINT_URL_BRAKET
Budgets	bu	AWS_ENDPOINT_URL_BUDGETS
Cost Explorer	cc o:	AWS_ENDPOINT_URL_COST_EXPLORER
chatbot	ch	AWS_ENDPOINT_URL_CHATBOT
Chime	ch	AWS_ENDPOINT_URL_CHIME
Chime SDK Identity	ch _:	AWS_ENDPOINT_URL_CHIME_SDK_IDENTITY

serviceId	Cl id ac de se pa cc ha A\ **cc** fil	**AWS_ENDPOINT_URL_<SERVICE>** variável de ambiente
Chime SDK Media Pipelines	ch _r pe	AWS_ENDPOINT_URL_CHIME_SDK_MEDIA_PIPELINES
Chime SDK Meetings	ch _r	AWS_ENDPOINT_URL_CHIME_SDK_MEETINGS
Chime SDK Messaging	ch _r g	AWS_ENDPOINT_URL_CHIME_SDK_MESSAGING
Chime SDK Voice	ch _\	AWS_ENDPOINT_URL_CHIME_SDK_VOICE
CleanRooms	cl s	AWS_ENDPOINT_URL_CLEANROOMS
CleanRoomsML	cl sr	AWS_ENDPOINT_URL_CLEANROOMSML
Cloud9	cl	AWS_ENDPOINT_URL_CLOUD9
CloudControl	cl rc	AWS_ENDPOINT_URL_CLOUDCONTROL

serviceId	Cl id ac de se pa cc ha A\ **cc** fil	**AWS_ENDPOINT_URL_<SERVICE>** variável de ambiente
CloudDirectory	c: ct	AWS_ENDPOINT_URL_CLOUDDIRECTORY
CloudFormation	c: at	AWS_ENDPOINT_URL_CLOUDFORMATION
CloudFront	c: t	AWS_ENDPOINT_URL_CLOUDFRONT
CloudFront KeyValueStore	c: t_ es	AWS_ENDPOINT_URL_CLOUDFRONT_KEYVALUESTORE
CloudHSM	c:	AWS_ENDPOINT_URL_CLOUDHSM
CloudHSM V2	c: v2	AWS_ENDPOINT_URL_CLOUDHSM_V2
CloudSearch	c: ch	AWS_ENDPOINT_URL_CLOUDSEARCH
CloudSearch Domain	c: ch	AWS_ENDPOINT_URL_CLOUDSEARCH_DOMAIN

serviceId	Cl id ac de se pa cc ha A\ **cc** fil	**AWS_ENDPOINT_URL_<SERVICE>** variável de ambiente
CloudTrail	cl l	AWS_ENDPOINT_URL_CLOUDTRAIL
CloudTrail Data	cl l_	AWS_ENDPOINT_URL_CLOUDTRAIL_DATA
CloudWatch	cl 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	Cl id ac de se pa co ha AV **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** variável de ambiente
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 notificat ions	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

serviceId	Cl id ac de se pa cc ha A\ **cc** fil	**AWS_ENDPOINT_URL_<SERVICE>** variável de ambiente
Cognito Sync	cc yr	AWS_ENDPOINT_URL_COGNITO_SYNC
Comprehend	cc d	AWS_ENDPOINT_URL_COMPREHEND
ComprehendMedical	cc dr	AWS_ENDPOINT_URL_COMPREHENDMEDICAL
Compute Optimizer	cc pt	AWS_ENDPOINT_URL_COMPUTE_OPTIMIZER
Config Service	cc rv	AWS_ENDPOINT_URL_CONFIG_SERVICE
Connect	cc	AWS_ENDPOINT_URL_CONNECT
Connect Contact Lens	cc or ns	AWS_ENDPOINT_URL_CONNECT_CONTACT_LENS
ConnectCampaigns	cc mp	AWS_ENDPOINT_URL_CONNECTCAMPAIGNS
ConnectCases	cc se	AWS_ENDPOINT_URL_CONNECTCASES

serviceId	Cl id ac de se pa cc ha A\ **cc** fil	**AWS_ENDPOINT_URL_<SERVICE>** variável de ambiente
ConnectParticipant	cc rt	AWS_ENDPOINT_URL_CONNECTPARTICIPANT
ControlTower	cc we	AWS_ENDPOINT_URL_CONTROLTOWER
Cost Optimization Hub	cc m: hu	AWS_ENDPOINT_URL_COST_OPTIMIZATION_HUB
Cost and Usage Report Service	cc us o: ce	AWS_ENDPOINT_URL_COST_AND_USAGE_REPO RT_SERVICE
Customer Profiles	cu pr	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

serviceId	Cl id ac de se pa cc ha A\ **cc** fil	**AWS_ENDPOINT_URL_<SERVICE>** variável de ambiente
DataSync	da	AWS_ENDPOINT_URL_DATASYNC
DataZone	da	AWS_ENDPOINT_URL_DATAZONE
DAX	da	AWS_ENDPOINT_URL_DAX
Detective	de	AWS_ENDPOINT_URL_DETECTIVE
Device Farm	de rr	AWS_ENDPOINT_URL_DEVICE_FARM
DevOps Guru	de ru	AWS_ENDPOINT_URL_DEVOPS_GURU
Direct Connect	d: nr	AWS_ENDPOINT_URL_DIRECT_CONNECT
Application Discovery Service	ap or e: ce	AWS_ENDPOINT_URL_APPLICATION_DISCOVERY_SERVICE
DLM	dl	AWS_ENDPOINT_URL_DLM

serviceId	Cl id ac de se pa cc ha A\ **cc** fil	**AWS_ENDPOINT_URL_<SERVICE>** variável de ambiente
Database Migration Service	da m: _s	AWS_ENDPOINT_URL_DATABASE_MIGRATION_ SERVICE
DocDB	dc	AWS_ENDPOINT_URL_DOCDB
DocDB Elastic	dc st	AWS_ENDPOINT_URL_DOCDB_ELASTIC
drs	dr	AWS_ENDPOINT_URL_DRS
Directory Service	di _s	AWS_ENDPOINT_URL_DIRECTORY_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 nc ct	AWS_ENDPOINT_URL_EC2_INSTANCE_CONNECT

serviceId	Cl id ac de se pa cc ha A\ **cc** fil	**AWS_ENDPOINT_URL_<SERVICE>** variável de ambiente
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
Elastic Beanstalk	el ea	AWS_ENDPOINT_URL_ELASTIC_BEANSTALK
Elastic Transcoder	el ra	AWS_ENDPOINT_URL_ELASTIC_TRANSCODER

serviceId	Cl id ac de se pa cc ha A\ **cc** fil	**AWS_ENDPOINT_URL_<SERVICE>** variável de ambiente
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
EventBridge	ev ge	AWS_ENDPOINT_URL_EVENTBRIDGE

serviceId	Cl id ac de se pa cc ha A\ **cc** fil	**AWS_ENDPOINT_URL_<SERVICE>** variável de ambiente
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	Cl id ac de se pa cc ha A\ **cc** fil	**AWS_ENDPOINT_URL_<SERVICE>** variável de ambiente
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 t:	AWS_ENDPOINT_URL_GROUNDSTATION
GuardDuty	gu	AWS_ENDPOINT_URL_GUARDDUTY
Health	he	AWS_ENDPOINT_URL_HEALTH
HealthLake	he e	AWS_ENDPOINT_URL_HEALTHLAKE

serviceId	Cl id ac de se pa cc ha A\ **cc** fil	**AWS_ENDPOINT_URL_<SERVICE>** variável de ambiente
Honeycode	ho	AWS_ENDPOINT_URL_HONEYCODE
IAM	ia	AWS_ENDPOINT_URL_IAM
identitystore	ic to	AWS_ENDPOINT_URL_IDENTITYSTORE
imagebuilder	ir de	AWS_ENDPOINT_URL_IMAGEBUILDER
ImportExport	ir o:	AWS_ENDPOINT_URL_IMPORTEXPORT
Inspector	ir	AWS_ENDPOINT_URL_INSPECTOR
Inspector Scan	ir _s	AWS_ENDPOINT_URL_INSPECTOR_SCAN
Inspector2	ir 2	AWS_ENDPOINT_URL_INSPECTOR2
InternetMonitor	ir or	AWS_ENDPOINT_URL_INTERNETMONITOR
IoT	ic	AWS_ENDPOINT_URL_IOT

serviceId	Cl id ac de se pa cc ha A\ **cc** fil	**AWS_ENDPOINT_URL_<SERVICE>** variável de ambiente
IoT Data Plane	ic p:	AWS_ENDPOINT_URL_IOT_DATA_PLANE
IoT Jobs Data Plane	ic da e	AWS_ENDPOINT_URL_IOT_JOBS_DATA_PLANE
IoT 1Click Devices Service	ic k_ _s	AWS_ENDPOINT_URL_IOT_1CLICK_DEVICES_ SERVICE
IoT 1Click Projects	ic k_ s	AWS_ENDPOINT_URL_IOT_1CLICK_PROJECTS
IoTAnalytics	ic ic	AWS_ENDPOINT_URL_IOTANALYTICS
IotDeviceAdvisor	ic ac	AWS_ENDPOINT_URL_IOTDEVICEADVISOR
IoT Events	ic s	AWS_ENDPOINT_URL_IOT_EVENTS
IoT Events Data	ic s_	AWS_ENDPOINT_URL_IOT_EVENTS_DATA

serviceId	Cl id ac de se pa cc ha A\ **cc** fil	**AWS_ENDPOINT_URL_<SERVICE>** variável de ambiente
IoTFleetHub	ic ul	AWS_ENDPOINT_URL_IOTFLEETHUB
IoTFleetWise	ic is	AWS_ENDPOINT_URL_IOTFLEETWISE
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

serviceId	Cl id ac de se pa cc ha A\ **cc** fil	**AWS_ENDPOINT_URL_<SERVICE>** variável de ambiente
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
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

serviceId	Cl id ac de se pa cc ha A\ **cc** fil	**AWS_ENDPOINT_URL_<SERVICE>** variável de ambiente
Kinesis Video Signaling	k: ic a:	AWS_ENDPOINT_URL_KINESIS_VIDEO_SIGNALING
Kinesis Video WebRTC Storage	k: ic t e	AWS_ENDPOINT_URL_KINESIS_VIDEO_WEBRT C_STORAGE
Kinesis Analytics	k: na	AWS_ENDPOINT_URL_KINESIS_ANALYTICS
Kinesis Analytics V2	k: na v2	AWS_ENDPOINT_URL_KINESIS_ANALYTICS_V2
Kinesis Video	k: ic	AWS_ENDPOINT_URL_KINESIS_VIDEO
KMS	kr	AWS_ENDPOINT_URL_KMS
LakeFormation	la t:	AWS_ENDPOINT_URL_LAKEFORMATION
Lambda	la	AWS_ENDPOINT_URL_LAMBDA

serviceId	Cl id ac de se pa cc ha A\ **cc** fil	**AWS_ENDPOINT_URL_<SERVICE>** variável de ambiente
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
License Manager	li ar	AWS_ENDPOINT_URL_LICENSE_MANAGER
License Manager Linux Subscriptions	li ar nu r:	AWS_ENDPOINT_URL_LICENSE_MANAGER_LIN UX_SUBSCRIPTIONS

serviceId	Cl id ac de se pa cc ha A\ **cc** fil	**AWS_ENDPOINT_URL_<SERVICE>** variável de ambiente
License Manager User Subscriptions	l: ar e: ip	AWS_ENDPOINT_URL_LICENSE_MANAGER_USE R_SUBSCRIPTIONS
Lightsail	l:	AWS_ENDPOINT_URL_LIGHTSAIL
Location	l(	AWS_ENDPOINT_URL_LOCATION
CloudWatch Logs	c: h_	AWS_ENDPOINT_URL_CLOUDWATCH_LOGS
CloudWatch Logs	c: h_	AWS_ENDPOINT_URL_CLOUDWATCH_LOGS
LookoutEquipment	l(u:	AWS_ENDPOINT_URL_LOOKOUTEQUIPMENT
LookoutMetrics	l(t:	AWS_ENDPOINT_URL_LOOKOUTMETRICS
LookoutVision	l(s:	AWS_ENDPOINT_URL_LOOKOUTVISION
m2	m2	AWS_ENDPOINT_URL_M2

serviceId	Cl id ac de se pa cc ha A\ **cc** fil	**AWS_ENDPOINT_URL_<SERVICE>** variável de ambiente
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 Deploymen t	ma ce me	AWS_ENDPOINT_URL_MARKETPLACE_DEPLOYMENT

serviceId	Cl id ac de se pa cc ha A\ **cc** fil	**AWS_ENDPOINT_URL_<SERVICE>** variável de ambiente
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

serviceId	Cl id ac de se pa cc ha A\ **cc** fil	**AWS_ENDPOINT_URL_<SERVICE>** variável de ambiente
MediaPackageV2	me ag	AWS_ENDPOINT_URL_MEDIAPACKAGEV2
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

serviceId	Cl id ac de se pa cc ha A\ **cc** fil	**AWS_ENDPOINT_URL_<SERVICE>** variável de ambiente
Migration Hub Refactor Spaces	m: _l ct es	AWS_ENDPOINT_URL_MIGRATION_HUB_REFAC TOR_SPACES
MigrationHub Config	m: hu g	AWS_ENDPOINT_URL_MIGRATIONHUB_CONFIG
MigrationHubOrches trator	m: hu t:	AWS_ENDPOINT_URL_MIGRATIONHUBORCHESTRATOR
MigrationHubStrategy	m: 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

serviceId	Cl id ac de se pa cc ha A\ **cc** fil	**AWS_ENDPOINT_URL_<SERVICE>** variável de ambiente
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
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

serviceId	Cl id ac de se pa cc ha A\ **cc** fil	**AWS_ENDPOINT_URL_<SERVICE>** variável de ambiente
OpsWorks	op	AWS_ENDPOINT_URL_OPSWORKS
OpsWorksCM	op m	AWS_ENDPOINT_URL_OPSWORKSCM
Organizations	or ic	AWS_ENDPOINT_URL_ORGANIZATIONS
OSIS	os	AWS_ENDPOINT_URL_OSIS
Outposts	ou	AWS_ENDPOINT_URL_OUTPOSTS
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

serviceId	Cl id ac de se pa cc ha AV **cc** fil	**AWS_ENDPOINT_URL_<SERVICE>** variável de ambiente
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
Pinpoint	pi	AWS_ENDPOINT_URL_PINPOINT
Pinpoint Email	pi er	AWS_ENDPOINT_URL_PINPOINT_EMAIL
Pinpoint SMS Voice	pi sr	AWS_ENDPOINT_URL_PINPOINT_SMS_VOICE

serviceId	Cl id ac de se pa cc ha A\ **cc** fil	**AWS_ENDPOINT_URL_<SERVICE>** variável de ambiente
Pinpoint SMS Voice V2	p: sr _\	AWS_ENDPOINT_URL_PINPOINT_SMS_VOICE_V2
Pipes	p:	AWS_ENDPOINT_URL_PIPES
Polly	p(	AWS_ENDPOINT_URL_POLLY
Pricing	p:	AWS_ENDPOINT_URL_PRICING
PrivateNetworks	p: tv	AWS_ENDPOINT_URL_PRIVATENETWORKS
Proton	p:	AWS_ENDPOINT_URL_PROTON
QBusiness	ql	AWS_ENDPOINT_URL_QBUSINESS
QConnect	q(	AWS_ENDPOINT_URL_QCONNECT
QLDB	q:	AWS_ENDPOINT_URL_QLDB
QLDB Session	q: i(	AWS_ENDPOINT_URL_QLDB_SESSION
QuickSight	qu t	AWS_ENDPOINT_URL_QUICKSIGHT

serviceId	Cl id ac de se pa cc ha A\ **cc** fil	**AWS_ENDPOINT_URL_<SERVICE>** variável de ambiente
RAM	ra	AWS_ENDPOINT_URL_RAM
rbin	rb	AWS_ENDPOINT_URL_RBIN
RDS	rc	AWS_ENDPOINT_URL_RDS
RDS Data	rc	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 or	AWS_ENDPOINT_URL_REKOGNITION
repostspace	re ce	AWS_ENDPOINT_URL_REPOSTSPACE
resiliencehub	re el	AWS_ENDPOINT_URL_RESILIENCEHUB

serviceId	Cl id ac de se pa cc ha A\ **cc** fil	**AWS_ENDPOINT_URL_<SERVICE>** variável de ambiente
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	Cl id ac de se pa cc ha A\ **cc** fil	**AWS_ENDPOINT_URL_<SERVICE>** variável de ambiente
Route53 Recovery Control Config	rc ec or nf	AWS_ENDPOINT_URL_ROUTE53_RECOVERY_CO NTROL_CONFIG
Route53 Recovery Readiness	rc ec ea	AWS_ENDPOINT_URL_ROUTE53_RECOVERY_RE ADINESS
Route 53 Domains	rc dc	AWS_ENDPOINT_URL_ROUTE_53_DOMAINS
Route53Resolver	rc sc	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

serviceId	Cl id ac de se pa cc ha A\ **cc** fil	**AWS_ENDPOINT_URL_<SERVICE>** variável de ambiente
SageMaker	sa	AWS_ENDPOINT_URL_SAGEMAKER
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 a:	AWS_ENDPOINT_URL_SAGEMAKER_GEOSPATIAL
SageMaker Metrics	sa _m	AWS_ENDPOINT_URL_SAGEMAKER_METRICS
SageMaker Runtime	sa _r	AWS_ENDPOINT_URL_SAGEMAKER_RUNTIME

serviceId	Cl id ac de se pa cc ha A\ **cc** fil	**AWS_ENDPOINT_URL_<SERVICE>** variável de ambiente
savingsplans	sa ar	AWS_ENDPOINT_URL_SAVINGSPLANS
Scheduler	sc	AWS_ENDPOINT_URL_SCHEDULER
schemas	sc	AWS_ENDPOINT_URL_SCHEMAS
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 uc	AWS_ENDPOINT_URL_SERVICE_QUOTAS

serviceId	Cl id ac de se pa cc ha A\ **cc** fil	**AWS_ENDPOINT_URL_<SERVICE>** variável de ambiente
Service Catalog	se at	AWS_ENDPOINT_URL_SERVICE_CATALOG
Service Catalog AppRegistry	se at p:	AWS_ENDPOINT_URL_SERVICE_CATALOG_APP REGISTRY
ServiceDiscovery	se sc	AWS_ENDPOINT_URL_SERVICEDISCOVERY
SES	se	AWS_ENDPOINT_URL_SES
SESv2	se	AWS_ENDPOINT_URL_SESV2
Shield	sh	AWS_ENDPOINT_URL_SHIELD
signer	s:	AWS_ENDPOINT_URL_SIGNER
SimSpaceWeaver	s: ea	AWS_ENDPOINT_URL_SIMSPACEWEAVER
SMS	sr	AWS_ENDPOINT_URL_SMS
Snow Device Managemen t	sr ce me	AWS_ENDPOINT_URL_SNOW_DEVICE_MANAGEMENT

serviceId	Cl id ac de se pa cc ha A\ **cc** fil	**AWS_ENDPOINT_URL_<SERVICE>** variável de ambiente
Snowball	sr	AWS_ENDPOINT_URL_SNOWBALL
SNS	sr	AWS_ENDPOINT_URL_SNS
SQS	sc	AWS_ENDPOINT_URL_SQS
SSM	ss	AWS_ENDPOINT_URL_SSM
SSM Contacts	ss ct	AWS_ENDPOINT_URL_SSM_CONTACTS
SSM Incidents	ss er	AWS_ENDPOINT_URL_SSM_INCIDENTS
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

serviceId	Cl id ac de se pa cc ha A\ **cc** fil	**AWS_ENDPOINT_URL_<SERVICE>** variável de ambiente
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	sv	AWS_ENDPOINT_URL_SWF
synthetics	sy s	AWS_ENDPOINT_URL_SYNTHETICS
Textract	te	AWS_ENDPOINT_URL_TEXTRACT
Timestream InfluxDB	ti m_ b	AWS_ENDPOINT_URL_TIMESTREAM_INFLUXDB
Timestream Query	ti m_	AWS_ENDPOINT_URL_TIMESTREAM_QUERY
Timestream Write	ti m_	AWS_ENDPOINT_URL_TIMESTREAM_WRITE

serviceId	Cl id ac de se pa cc ha A\ **cc** fil	**AWS_ENDPOINT_URL_<SERVICE>** variável de ambiente
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
VPC Lattice	vp ce	AWS_ENDPOINT_URL_VPC_LATTICE
WAF	wa	AWS_ENDPOINT_URL_WAF
WAF Regional	wa na	AWS_ENDPOINT_URL_WAF_REGIONAL

serviceId	Cl id ac de se pa cc ha A\ **cc** fil	**AWS_ENDPOINT_URL_<SERVICE>** variável de ambiente
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
WorkSpaces Thin Client	wo s_ ie	AWS_ENDPOINT_URL_WORKSPACES_THIN_CLIENT
WorkSpaces Web	wo s_	AWS_ENDPOINT_URL_WORKSPACES_WEB

serviceId	Cl id ac de se pa cc ha A\ **cc** fil	**AWS_ENDPOINT_URL_<SERVICE>** variável de ambiente
XRay	x:	AWS_ENDPOINT_URL_XRAY

Padrões de configuração inteligente

Com o recurso de padrões de configuração inteligente, AWS SDKspode fornecer valores padrão predefinidos e otimizados para outras configurações.

Configure essa funcionalidade usando o seguinte:

defaults_mode- compartilhado AWS **config**configuração de arquivo, **AWS_DEFAULTS_MODE**: variável de ambiente, **aws.defaultsMode**- propriedade JVM do sistema: somente Java/Kotlin

Com essa configuração, você pode escolher um modo que se alinhe à arquitetura do aplicativo e, em seguida, forneça valores padrão otimizados para o aplicativo. Se um AWS SDKA configuração tem um valor definido explicitamente, então esse valor sempre tem precedência. Se um AWS SDKA configuração não tem um valor definido explicitamente e não `defaults_mode` é igual à legada, então esse recurso pode fornecer valores padrão diferentes para várias configurações otimizadas para seu aplicativo. As configurações podem incluir o seguinte: configurações de HTTP comunicação, comportamento de repetição, configurações de endpoint regional do serviço e, potencialmente, qualquer configuração SDK relacionada. Os clientes que usam esse atributo podem obter novos padrões de configuração personalizados para cenários de uso comuns. Se você não `defaults_mode` for igual `alegacy`, recomendamos realizar testes de seu aplicativo ao atualizar oSDK, pois os valores padrão fornecidos podem mudar à medida que as melhores práticas evoluem.

Valor padrão: `legacy`

Nota: As novas versões principais do SDKs terão como padrão`standard`.

Valores válidos:

- `legacy`— Fornece configurações padrão que variam SDK e existiam antes do estabelecimento do`defaults_mode`.
- `standard`: fornece os valores padrão recomendados mais recentes que devem ser executados com segurança na maioria dos cenários.
- `in-region`— Baseia-se no modo padrão e inclui otimização personalizada para aplicativos que chamam Serviços da AWS de dentro do mesmo Região da AWS.
- `cross-region`— Baseia-se no modo padrão e inclui otimização personalizada para aplicativos que chamam Serviços da AWS em uma região diferente.
- `mobile`: baseia-se no modo padrão e inclui otimização personalizada para aplicativos móveis.
- `auto`: baseia-se no modo padrão e inclui atributos experimentais. As SDK tentativas de descobrir o ambiente de tempo de execução para determinar automaticamente as configurações apropriadas. A detecção automática é baseada em heurísticas e não fornece 100% de precisão. Se o ambiente de runtime não puder ser determinado, o modo `standard` será usado. A detecção automática pode consultar os metadados da instância, o que pode introduzir latência. Se a latência de inicialização for fundamental para seu aplicativo, recomendamos escolher um `defaults_mode` explícito.

Exemplo de configuração desse valor no arquivo `config`:

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

Os parâmetros a seguir podem ser otimizados com base na seleção de `defaults_mode`:

- `retryMode`— Especifica como as SDK tentativas são repetidas. Consulte Comportamento de repetição.
- `stsRegionalEndpoints`— Especifica como o SDK determina o AWS service (Serviço da AWS) endpoint que ele usa para falar com o AWS Security Token Service (AWS STS). VejaAWS STS Endpoints regionais.
- `s3UsEast1RegionalEndpoints`— Especifica como o SDK determina o AWS endpoint de serviço que ele usa para se comunicar com o Amazon S3 da região. `us-east-1`
- `connectTimeoutInMillis`: depois de fazer uma tentativa inicial de conexão em um soquete, a quantidade de tempo antes do tempo limite. Se o cliente não receber a conclusão do handshake de conexão, ele desiste e falhará na operação.

- `tlsNegotiationTimeoutInMillis`— O tempo máximo que um TLS handshake pode levar desde o momento em que a CLIENT HELLO mensagem é enviada até o momento em que o cliente e o servidor negociaram totalmente as cifras e trocaram as chaves.

O valor padrão para cada configuração muda dependendo da `defaults_mode` selecionada para seu aplicativo. Atualmente, esses valores são definidos da seguinte forma (sujeitos a alterações):

Parâmetro	Modo **`standard`**	Modo **`in-region`**	Modo **`cross-region`**	Modo **`mobile`**
`retryMode`	`standard`	`standard`	`standard`	`standard`
`stsRegionalEndpoints`	`regional`	`regional`	`regional`	`regional`
`s3UsEast1RegionalEndpoints`	`regional`	`regional`	`regional`	`regional`
`connectTimeoutInMillis`	3100	1100	3100	30000
`tlsNegotiationTimeoutInMillis`	3100	1100	3100	30000

Por exemplo, se o `defaults_mode` que você selecionou fosse `standard`, o valor de `standard` seria atribuído a `retry_mode` (das opções `retry_mode` válidas) e o valor de `regional` seria atribuído a `stsRegionalEndpoints` (das opções `stsRegionalEndpoints` válidas).

Compatibilidade com AWS SDKs

Os itens a seguir SDKs oferecem suporte aos recursos e configurações descritos neste tópico. Quaisquer exceções parciais estão anotadas. Todas as configurações de propriedade do JVM sistema são suportadas pelo AWS SDK for Java e o AWS SDK para Kotlin somente.

SDK	Compatível	Notas ou mais informações
AWS CLI v2	Não	
SDKpara C++	Sim	Parâmetros não otimizados:`stsRegionalEndpoints` , `s3UsEast1RegionalEndpoints` , `tlsNegotiationTimeoutInMillis` .
SDKpara Go V2 (1.x)	Sim	Parâmetros não otimizados:`retryMode` , `stsRegionalEndpoints` , `s3UsEast1RegionalEndpoints` .
SDKpara Go 1.x (V1)	Não	
SDKpara Java 2.x	Sim	Parâmetros não otimizados:`stsRegionalEndpoints` .
SDKpara Java 1.x	Não	
SDKpara JavaScript 3.x	Sim	Parâmetros não otimizados:`stsRegionalEndpoints` , `s3UsEast1RegionalEndpoints` , `tlsNegotiationTimeoutInMillis` . `connectTimeoutInMi`

SDK	Compatível	Notas ou mais informações
		`llis` é chamado `connectionTimeout` .
SDKpara JavaScript 2.x	Não	
SDKpara Kotlin	Não	
SDKpara. NET3.x	Sim	Parâmetros não otimizados: `connectTimeoutInMillis` , `tlsNegotiationTimeoutInMillis` .
SDKpara PHP 3.x	Sim	Parâmetros não otimizados: `tlsNegotiationTimeoutInMillis` .
SDKpara Python (Boto3)	Sim	Parâmetros não otimizados: `tlsNegotiationTimeoutInMillis` .
SDKpara Ruby 3.x	Sim	
SDKpara Rust	Não	
SDKpara Swift	Não	
Ferramentas para PowerShell	Sim	Parâmetros não otimizados: `connectTimeoutInMillis` , `tlsNegotiationTimeoutInMillis` .

AWS Bibliotecas Common Runtime (CRT)

As bibliotecas AWS Common Runtime (CRT) são uma biblioteca base doSDKs. CRTÉ uma família modular de pacotes independentes, escrita em C. Cada pacote oferece bom desempenho e ocupa pouco espaço para as diferentes funcionalidades necessárias. Essas funcionalidades são comuns e compartilhadas entre todos, SDKs proporcionando melhor reutilização, otimização e precisão do código. Os pacotes são:

- `awslabs/aws-c-auth`: autenticação AWS do lado do cliente (provedores de credenciais padrão e assinatura (sigv4))
- `awslabs/aws-c-cal`: tipos criptográficos primitivos, hashes (,, SHA256HMAC)MD5, SHA256 signatários, AES
- `awslabs/aws-c-common`: estruturas de dados básicas, tipos primitivos de encadeamento/sincronização, gerenciamento de buffer, funções relacionadas ao stdlib
- `awslabs/aws-c-compression`: algoritmos de compressão (codificação/decodificação Huffman)
- `awslabs/aws-c-event-stream`: processamento de mensagens de fluxo de eventos (cabeçalhos, prelúdio, carga útil, crc/trailer), implementação de chamada de procedimento remoto () em fluxos de eventos RPC
- `awslabs/aws-c-http`: implementação C99 das especificações HTTP /1.1 e HTTP /2
- `awslabs/aws-c-io`: Soquetes (TCP,UDP), tubosDNS, circuitos de eventos, canais,/SSLTLS
- `awslabs/aws-c-iot`: Implementação C99 da integração de serviços de nuvem de AWS IoT com dispositivos
- `awslabs/aws-c-mqtt`: protocolo de mensagens leve e padrão para a Internet das Coisas (IoT)
- `awslabs/aws-c-s3`: Implementação da biblioteca C99 para comunicação com o serviço Amazon S3, projetada para maximizar a taxa de transferência em instâncias Amazon de alta largura de banda EC2
- `awslabs/aws-c-sdkutils`: Uma biblioteca de utilitários para analisar e gerenciar perfis AWS
- `awslabs/aws-checksums`: acelerado por hardware multiplataforma CRC32c e CRC32 com retorno a implementações eficientes de software
- `awslabs/aws-lc`: biblioteca criptográfica de uso geral mantida pela equipe de AWS criptografia AWS e seus clientes, com base no código do projeto Google Boring SSL e do projeto Open SSL
- `awslabs/s2n`: Implementação C99 dos SSL protocolosTLS/, projetados para serem pequenos e rápidos, com a segurança como prioridade

O CRT está disponível em todosSDKs, exceto Go e Rust.

CRTdependências

As CRT bibliotecas formam uma rede complexa de relacionamentos e dependências. Conhecer essas relações é útil se você precisar CRT construí-las diretamente da fonte. No entanto, a maioria dos usuários acessa a CRT funcionalidade por meio de sua linguagem SDK (como AWS SDK para C++ ou AWS SDK Java) ou de seu SDK dispositivo de IoT de linguagem (como AWS IoT SDK para C++ ou AWS IoT para Java). SDK No diagrama a seguir, a caixa CRT Ligações de idioma se refere ao pacote que agrupa as CRT bibliotecas de um idioma específico. SDK Essa é uma coleção de pacotes do formulárioaws-crt-*, em que '*' é um SDK idioma (como aws-crt-cppou aws-crt-java).

A seguir está uma ilustração das dependências hierárquicas das bibliotecas. CRT

AWS Política de manutenção de SDKs e ferramentas

Visão geral

Este documento descreve a política de manutenção de kits de desenvolvimento de AWS software (SDKs) e ferramentas, incluindo SDKs móveis e de IoT, e suas dependências subjacentes. AWS fornece regularmente aos AWS SDKs e às ferramentas atualizações que podem conter suporte para AWS APIs novas ou atualizadas, novos recursos, aprimoramentos, correções de bugs, patches de segurança ou atualizações de documentação. As atualizações também podem abordar alterações nas dependências, nos tempos de execução da linguagem e nos sistemas operacionais. AWS As versões do SDK são publicadas em gerenciadores de pacotes (por exemplo, Maven, NuGet PyPI) e estão disponíveis como código-fonte em. GitHub

Recomendamos que os usuários continuem up-to-date com as versões do SDK para acompanhar os recursos, as atualizações de segurança e as dependências subjacentes mais recentes. O uso contínuo de uma versão não compatível do SDK não é recomendado e é feito a critério do usuário.

Versionamento

As versões de lançamento do AWS SDK estão na forma de X.Y.Z, onde X representa a versão principal. O aumento da versão principal de um SDK indica que esse SDK passou por mudanças significativas e substanciais para oferecer suporte a novos idiomas e padrões na linguagem. As versões principais são introduzidas quando interfaces públicas (por exemplo, classes, métodos, tipos etc.), comportamentos ou semânticas mudam. Os aplicativos precisam ser atualizados para que funcionem com a versão mais recente do SDK. É importante atualizar as versões principais com cuidado e de acordo com as diretrizes de atualização fornecidas pelo AWS.

Ciclo de vida da versão principal do SDK

O ciclo de vida das principais versões de SDKs e Ferramentas consiste em 5 fases, descritas abaixo.

- Developer Preview (Fase 0) - Durante essa fase, os SDKs não são suportados, não devem ser usados em ambientes de produção e são destinados apenas para fins de acesso antecipado e feedback. É possível que versões futuras introduzam mudanças significativas. Depois de AWS identificar uma versão como um produto estável, ela pode marcá-la como candidata a lançamento.

Os candidatos a lançamento estão prontos para o lançamento do GA, a menos que surjam bugs significativos, e receberão suporte total para AWS .

- Disponibilidade geral (GA) (Fase 1) - Durante essa fase, os SDKs são totalmente suportados. AWS fornecerá lançamentos regulares do SDK que incluem suporte para novos serviços, atualizações de API para serviços existentes, bem como correções de bugs e segurança. Para Ferramentas, AWS fornecerá lançamentos regulares que incluem novas atualizações de recursos e correções de erros. AWS suportará a versão GA de um SDK por pelo menos 24 meses.
- Anúncio de manutenção (Fase 2) - AWS fará um anúncio público pelo menos 6 meses antes de um SDK entrar no modo de manutenção. Durante esse período, o SDK continuará sendo totalmente suportado. Normalmente, o modo de manutenção é anunciado ao mesmo tempo em que a próxima versão principal é transferida para GA.
- Manutenção (Fase 3) - Durante o modo de manutenção, AWS limita as versões do SDK para tratar apenas de correções críticas de bugs e problemas de segurança. Um SDK não receberá atualizações de API para serviços novos ou existentes, nem será atualizado para oferecer suporte a novas regiões. O modo de manutenção tem uma duração padrão de 12 meses, a menos que especificado de outra forma.
- Fim do suporte (Fase 4) - Quando um SDK chega ao fim do suporte, ele não receberá mais atualizações ou lançamentos. As versões publicadas anteriormente continuarão disponíveis por meio de gerenciadores de pacotes públicos e o código permanecerá ativado GitHub. O GitHub repositório pode ser arquivado. O uso de um SDK alcançado end-of-support é feito a critério do usuário. Recomendamos que os usuários atualizem para a nova versão principal.

Veja a seguir uma ilustração visual do ciclo de vida da versão principal do SDK. Observe que os cronogramas mostrados abaixo são ilustrativos e não vinculativos.

Ciclo de vida da dependência

A maioria dos AWS SDKs tem dependências subjacentes, como tempos de execução de linguagem, sistemas operacionais ou bibliotecas e estruturas de terceiros. Essas dependências geralmente estão vinculadas à comunidade linguística ou ao fornecedor que possui esse componente específico. Cada comunidade ou fornecedor publica sua própria end-of-support programação para seu produto.

Os termos a seguir são usados para classificar as dependências subjacentes de terceiros:

- Sistema operacional (SO): exemplos incluem Amazon Linux AMI, Amazon Linux 2, Windows 2008, Windows 2012, Windows 2016, etc.

- Language Runtime: exemplos incluem Java 7, Java 8, Java 11, .NET Core, .NET Standard, .NET PCL etc.
- Biblioteca/estrutura de terceiros: exemplos incluem OpenSSL, .NET Framework 4.5, Java EE etc.

Nossa política é continuar oferecendo suporte às dependências do SDK por pelo menos 6 meses após a comunidade ou o fornecedor encerrar o suporte para a dependência. Essa política, no entanto, pode variar dependendo da dependência específica.

> **Note**
>
> AWS reserva o direito de interromper o suporte para uma dependência subjacente sem aumentar a versão principal do SDK

Métodos de comunicação

Os anúncios de manutenção são comunicados de várias maneiras:

- Um anúncio por e-mail é enviado às contas afetadas, anunciando nossos planos de encerrar o suporte para a versão específica do SDK. O e-mail descreverá o caminho end-of-support, especificará os cronogramas da campanha e fornecerá orientações de atualização.
- AWS A documentação do SDK, como documentação de referência da API, guias do usuário, páginas de marketing de produtos do SDK e GitHub readme (s), é atualizada para indicar o cronograma da campanha e fornecer orientação sobre a atualização dos aplicativos afetados.
- É publicada uma postagem no AWS blog que descreve o caminho e reitera os cronogramas da campanha. end-of-support
- Os avisos de depreciação são adicionados aos SDKs, descrevendo o caminho end-of-support e vinculando à documentação do SDK.

Para ver a lista das principais versões disponíveis dos AWS SDKs e das ferramentas e onde elas estão em seu ciclo de vida de manutenção, consulte. Suporte à versão

AWS SDKse suporte à versão Tools

A tabela abaixo mostra a lista de disponíveis AWS Versões principais do Software Development Kit (SDK) e onde elas estão no ciclo de vida de manutenção com os cronogramas associados. Para obter informações detalhadas sobre o ciclo de vida das principais versões do AWS SDKse Ferramentas e suas dependências subjacentes, consulte Política de manutenção.

SDK	Versão principal	Fase atual	Data da disponibilidade geral	Observações
AWS CLI	1.x	Disponibilidade geral	02/09/2013	
AWS CLI	2.x	Disponibilidade geral	2/10/2020	
SDKpara C++	1.x	Disponibilidade geral	02/09/2015	
SDKpara Go V2	V2 1.x	Disponibilidade geral	19/01/2021	
SDKpara Go	1.x	Manutenção	19/11/2015	Veja o anúncio para obter detalhes e datas
SDKpara Java	1.x	Manutenção	25/03/2010	Veja o anúncio para obter detalhes e datas
SDKpara Java	2.x	Disponibilidade geral	20/11/2018	
SDKpara JavaScript	1.x	Fim do suporte	6/5/2013	

SDK	Versão principal	Fase atual	Data da disponibilidade geral	Observações
SDKpara JavaScript	2.x	Manutenção	19/06/2014	Veja o anúncio para obter detalhes e datas
SDKpara JavaScript	3.x	Disponibilidade geral	15/12/2020	
SDKpara Kotlin	1.x	Disponibilidade geral	27/11/2023	
SDKpara. NET	1.x	Fim do suporte	11/2009	
SDKpara. NET	2.x	Fim do suporte	08/11/2013	
SDKpara. NET	3.x	Disponibilidade geral	28/07/2015	
SDK para PHP	2.x	Fim do suporte	02/11/2012	
SDK para PHP	3.x	Disponibilidade geral	27/05/2015	
SDKpara Python (Boto2)	1.x	Fim do suporte	13/07/2011	
SDKpara Python (Boto3)	1.x	Disponibilidade geral	22/06/2015	
SDKpara Python (Botocore)	1.x	Disponibilidade geral	22/06/2015	
SDKpara Ruby	1.x	Fim do suporte	14/07/2011	
SDKpara Ruby	2.x	Fim do suporte	15/02/2015	

SDK	Versão principal	Fase atual	Data da disponibilidade geral	Observações
SDKpara Ruby	3.x	Disponibilidade geral	29/08/2017	
SDKpara Rust	1.x	Disponibilidade geral	27/11/2023	
SDKpara Swift	1.x	Disponibilidade geral	17/09/2024	
Ferramentas para PowerShell	2.x	Fim do suporte	08/11/2013	
Ferramentas para PowerShell	3.x	Fim do suporte	29/07/2015	
Ferramentas para PowerShell	4.x	Disponibilidade geral	21/11/2019	

Procurando por uma ferramenta SDK ou não mencionada? CriptografiaSDKs, dispositivo SDKs de IoT e dispositivos móveisSDKs, por exemplo, não estão incluídos neste guia. Para encontrar documentação sobre essas outras ferramentas, consulte Ferramentas para construir AWS.

Histórico do documento para AWS SDKsGuia de referência de ferramentas e ferramentas

A tabela a seguir descreve adições e atualizações importantes no AWS SDKsGuia de referência de ferramentas e ferramentas. Para receber notificações sobre atualizações desta documentação, você pode assinar o RSS feed.

Alteração	Descrição	Data
Adicionando Swift SDK à referência de configurações	Adicionando SDK suporte ao Swift a todas as referências de configuração Compatibilidade com AWS SDKsmesas.	17 de setembro de 2024
SDKpara propriedades do sistema Java 1.x	Adicione detalhes sobre as configurações do JVM sistema suportadas pelo AWS SDK for Java 1.x.	30 de maio de 2024
Atualizações de configurações	Adicione as configurações do JVM sistema.	27 de março de 2024
Atualizações da tabela de compatibilidade	Atualizações na compatibi lidade para SDK suporte, atualizações nos procedime ntos do IAM Identity Center.	20 de fevereiro de 2024
Atualização da credencial do contêiner. IMDSatualizar.	Adicionando suporte para a AmazonEKS. Adicionar configuração para desativar o IMDSv1 fallback.	29 de dezembro de 2023
Compactação de solicitações	Adicionar configurações para o recurso de compactação de solicitações.	27 de dezembro de 2023

Tabelas de compatibilidade	Tabelas de compatibilidade SDK e recursos de ferrament as atualizados SDK para incluir Kotlin, SDK Rust e AWS Tools for PowerShell.	10 de dezembro de 2023
Atualizações de autenticação	Atualizações dos métodos de autenticação SDKs e ferramentas compatíveis.	1º de julho de 2023
IAMatualizações de melhores práticas	Guia atualizado para se alinhar às IAM melhores práticas. Para obter mais informações, consulte Melhores práticas de segurança em IAM.	27 de fevereiro de 2023
SSOatualizações	Atualizações SSO nas credenciais da nova configura ção do SSO token.	19 de novembro de 2022
Atualizações de configurações	Atualizações na tabela de suporte para configuração geral e para pontos de acesso multirregionais do Amazon S3.	17 de novembro de 2022
Atualizações de configurações	Atualizações para clareza do IMDS cliente e das IMDS credenciais. Atualizações nas variáveis de ambiente.	4 de novembro de 2022
Atualização da página de boas-vindas	Anunciando a Amazon CodeWhisperer.	22 de setembro de 2022
Alteração do nome do serviço para login único	Atualizações para refletir isso AWS SSOagora é referido como AWS IAM Identity Center.	26 de julho de 2022

Atualização de configurações	Pequenas atualizações nos detalhes do arquivo de configuração e nas configura ções suportadas.	15 de junho de 2022
Atualização	Atualização massiva de quase todas as partes deste guia.	1º de fevereiro de 2022
Lançamento inicial	A primeira versão deste guia foi lançada ao público.	13 de março de 2020

Glossário do AWS

Para obter a terminologia mais recente da AWS, consulte o glossário da AWS na Referência do Glossário da AWS.

As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.