



Guia de referência

AWS SDKs e ferramentas



AWS SDKs e ferramentas: Guia de referência

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

As marcas comerciais e imagens de marcas da Amazon não podem ser usadas no contexto de nenhum produto ou serviço que não seja da Amazon, nem de qualquer maneira que possa gerar confusão entre os clientes ou que deprecie ou desprestige a Amazon. Todas as outras marcas comerciais que não pertencem à Amazon pertencem a seus respectivos proprietários, que podem ou não ser afiliados, patrocinados pela Amazon ou ter conexão com ela.

Table of Contents

AWS SDKs Guia de referência de ferramentas e ferramentas	1
Recursos para desenvolvedores	3
Notificação de telemetria de kit de ferramentas	3
Configuração	5
Arquivos config e credentials compartilhados	6
Perfis	6
Formato do arquivo de configuração	8
Formato do arquivo de credenciais	11
Localização de arquivos compartilhados	12
Resolução do diretório inicial	12
Alterar a localização padrão desses arquivos	13
Variáveis de ambiente	14
Como definir variáveis de ambiente	15
Configuração de variável de ambiente sem servidor	16
Propriedade do sistema de JVM	16
Como definir as propriedades do sistema em JVM	17
Autenticação e acesso	19
Escolher um método para autenticar o código da aplicação	19
Métodos de autenticação	23
ID do builder AWS	25
Faça login usando as credenciais do console	25
Como funciona	26
Autenticação do IAM Identity Center	26
Pré-requisitos	27
Configure o acesso programático usando o Centro de Identidade do IAM	27
Atualizar sessões de acesso ao portal	31
Entender a autenticação do IAM Identity Center	31
IAM Roles Anywhere	35
Etapa 1: configurar IAM Roles Anywhere	36
Etapa 2: usar IAM Roles Anywhere	36
Assumir uma função	37
Assumir um perfil do IAM	38
Assumir um perfil (web)	40
Federar com identidade da Web ou OpenID Connect	40

AWS chaves de acesso	42
Use credenciais de curto prazo	42
Use credenciais de longo prazo	42
Credenciais de curto prazo	44
Credenciais de longo prazo	45
Funções do IAM para EC2 instâncias	49
Criar um perfil do IAM	49
Inicie uma EC2 instância da Amazon e especifique sua função do IAM	49
Conecte-se à EC2 instância	50
Execute seu aplicativo na EC2 instância	50
Propagação de identidades confiáveis	51
Pré-requisitos para o uso do plug-in TIP	52
Para usar o plug-in TIP no código	52
Exemplos de código usando TIP	55
Referência de configurações	62
Criar clientes de serviço	62
Precedência de configurações	62
Entender as páginas de configurações deste guia	64
Lista de configurações de arquivo Config	65
Lista de configurações de arquivo Credentials	70
Lista de variáveis de ambiente	70
Lista de propriedades de sistema de JVM	75
Provedores de credenciais padronizados	79
Entender a cadeia de provedores de credenciais	80
Cadeias de provedores de credenciais específicas do SDK e da ferramenta	81
AWS teclas de acesso	82
Provedor de login	85
Assuma o provedor de perfil	88
Provedor de contêiner	95
Provedor do IAM Identity Center	99
Provedor de IMDS	106
Provedor de processo	111
Atributos padronizados	116
Endpoints baseados em conta	117
ID da aplicação	119
Metadados da instância do Amazon EC2	122

Pontos de acesso Amazon S3	125
Pontos de acesso de várias regiões do Amazon S3	127
Autenticação de sessão da S3 Express One Zone	130
Esquema de autenticação	133
Região da AWS	136
AWS STS Endpoints regionais	139
Proteções de integridade de dados	145
Endpoints de pilha dupla e FIPS	150
Descoberta de endpoint	153
Configuração geral	156
Injeção de prefixo de host	160
Cliente de IMDS	164
Comportamento de repetição	167
Compactação de solicitações	182
Endpoints específicos de serviço	184
Padrões de configuração inteligentes	233
Segurança	239
Habilitando o TLS híbrido pós-quântico	239
SDKs com PQ TLS habilitado por padrão	240
Opt-in Suporte para PQ TLS	241
SDKs que dependem do System OpenSSL	241
AWS SDKs e ferramentas que não planejam oferecer suporte ao PQ TLS	243
Commmom runtime	244
Adicionar dependências	245
Política de manutenção	246
Visão geral do	246
Versionamento	246
Ciclo de vida da versão principal do SDK	246
Ciclo de vida de dependência	247
Métodos de comunicação	248
Ciclo de vida da versão	249
Histórico do documento	252
.....	cclvi

O que é abordado no Guia de referência de ferramentas AWS SDKs 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 AWS SDKs e ferramentas globais](#)— Como usar os `credentials` arquivos config compartilhados ou variáveis de ambiente para configurar suas AWS SDKs ferramentas.
- [Uso e ferramentas de autenticação AWS SDKs e acesso](#)— Estabeleça como seu código ou ferramenta se autentica AWS quando você desenvolve com Serviços da AWS.
- [AWS SDKs referência de configurações e ferramentas](#): referência para todas as configurações padronizadas disponíveis para autenticação e configuração.
- [AWS Bibliotecas do Common Runtime \(CRT\)](#)— Visão geral das bibliotecas compartilhadas do AWS Common Runtime (CRT) que estão disponíveis para quase todos SDKs.
- [AWS SDKs e política de manutenção de ferramentas](#) abrange 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 para o SDK ou ferramenta que você está usando deve ser usado além de qualquer informação apresentada aqui. A seguir estão o SDK e as ferramentas que têm seções relevantes do material neste guia:

Se você estiver usando:	As seções relevantes deste guia para você são:
<ul style="list-style-type: none">• Qualquer SDK ou ferramenta	AWS SDKs e política de manutenção de ferramentas
<ul style="list-style-type: none">• AWS Cloud9• AWS CDK• Kit de ferramentas da AWS para Azure DevOps	Configuração AWS SDKs e ferramentas globais

Se você estiver usando:	As seções relevantes deste guia para você são:
<ul style="list-style-type: none"> • AWS Toolkit for JetBrains • 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 	<p>Uso e ferramentas de autenticação AWS SDKs e acesso</p> <p>AWS SDKs e política de manutenção de ferramentas</p>
<ul style="list-style-type: none"> • AWS CLI • AWS SDK para C++ • AWS SDK para Go • AWS SDK para Java • AWS SDK para JavaScript • AWS SDK para Kotlin • AWS SDK para .NET • AWS SDK para PHP • AWS SDK para Python (Boto3) • AWS SDK para Ruby • AWS SDK para Rust • AWS SDK para Swift • AWS Tools for Windows PowerShell 	<p>Configuração AWS SDKs e ferramentas globais</p> <p>Uso e ferramentas de autenticação AWS SDKs e acesso</p> <p>AWS SDKs referência de configurações e ferramentas</p> <p>AWS Bibliotecas do Common Runtime (CRT)</p> <p>AWS SDKs e política de manutenção de ferramentas</p> <p>AWS SDKs e ciclo de vida da versão Tools</p>

- 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](#).

- Para obter a AWS terminologia, consulte o [AWS glossário](#) na Glossário da AWS Referência.

Recursos para desenvolvedores

O Amazon Q Developer é um assistente conversacional generativo baseado em IA que pode ajudar você a entender, criar, estender e operar aplicativos. 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 [O que é Amazon Q Developer?](#) no Guia do usuário do Amazon Q Developer.

Notificação de telemetria de kit de ferramentas

AWS Os kits de ferramentas do Ambiente de Desenvolvimento Integrado (IDE) são plug-ins e extensões que permitem o acesso aos AWS serviços em seu IDE. Os plug-ins e extensões do Amazon Q para IDE permitem assistência de IA generativa no IDE. Para obter informações detalhadas sobre cada um dos Toolkits para IDE, consulte os Guias de usuário de kits de ferramentas na tabela anterior. Para saber mais sobre como usar o Amazon Q no IDE, consulte o tópico [Using Amazon Q in the IDE](#) no Amazon Q Developer Guide.

AWS Os kits de ferramentas do IDE e o Amazon Q podem coletar e armazenar dados de telemetria do lado do cliente para informar decisões sobre futuras versões 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 ferramentas do AWS IDE 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 kits de ferramentas do AWS IDE e extensões do Amazon Q, consulte os documentos de recursos nos seguintes AWS repositórios do kit de ferramentas: 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 AWS SDKs e ferramentas globais

Com AWS SDKs outras ferramentas de AWS desenvolvedor, como a AWS Command Line Interface (AWS CLI), você pode interagir com o AWS serviço APIs. 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, as ferramentas AWS SDKs e oferecem suporte a configurações comuns de fontes primárias além do próprio código. Esta seção abrange os seguintes tópicos:

Tópicos

- [Usando credentials arquivos compartilhados config para configurar AWS SDKs e ferramentas globalmente](#)
- [Localizando e alterando a localização dos arquivos compartilhados, dos credentials arquivos config AWS SDKs e das ferramentas](#)
- [Usando variáveis de ambiente para configuração AWS SDKs e ferramentas globais](#)
- [Usando propriedades do sistema JVM para configurar e AWS SDK para Java AWS SDK para Kotlin](#)

Usando **credentials** arquivos compartilhados **config** para configurar AWS SDKs e ferramentas globalmente

O compartilhamento AWS `config` e `credentials` os arquivos são a forma mais comum de especificar a autenticação e a configuração em um AWS SDK ou ferramenta.

Os arquivos compartilhados `config` e `credentials` contêm um conjunto de perfis. Um perfil é um conjunto de definições de configuração, em pares chave-valor, usado pelo AWS SDKs AWS Command Line Interface (AWS CLI) e por outras ferramentas. Os valores de configuração são anexados a um perfil para configurar algum aspecto do SDK/tool 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 `config` quanto `credentials` compartilhados são arquivos de texto simples que contêm somente caracteres ASCII (codificados em UTF-8). Eles assumem a forma do que geralmente é chamado de [arquivos INI](#).

Perfis

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

O perfil `[default]` contém os valores que são usados por uma operação de SDK ou ferramenta 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 a aplicação e o 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 perfil `[default]` e não definir em um perfil nomeado, o valor não estará definido quando você usar o perfil nomeado.

Definir um perfil nomeado

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

Para o SDK para Kotlin na JVM e o SDK para Java 2.x, você pode [definir a propriedade do sistema aws.profile](#). Quando o SDK cria um cliente de serviço, ele usa as configurações no perfil nomeado, a menos que a configuração seja substituída no código. O SDK para Java 1.x não é compatível com essa propriedade do sistema.

Note

Recomendamos que você sempre use perfis nomeados em vez do perfil padrão se a aplicação estiver em um servidor executando várias aplicações. O perfil padrão é automaticamente selecionado por qualquer AWS aplicativo no ambiente e compartilhado entre eles. Portanto, se alguém atualizar o perfil padrão de sua própria aplicação, isso poderá involuntariamente afetar as outras. Para evitar isso, defina um perfil nomeado no arquivo compartilhado `config` e use esse perfil nomeado em sua aplicação ao configurar o perfil

nomeado em seu código. Use a variável de ambiente ou a propriedade do sistema de JVM para definir o perfil nomeado se souber que seu escopo afeta apenas sua aplicação.

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, hífen (`-`) 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 de seção `profile`.

O exemplo a seguir mostra um arquivo `config` básico com um perfil `[default]`. Ele define a configuração [region](#). Todas as configurações depois dessa linha, até outra definição de seção ser 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 de sessão `profile` é um agrupamento nomeado de configurações 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 arquivo `config` com uma linha de definição de seção `profile` e um perfil nomeado denominado `foo`. Todas as configurações depois dessa linha, até outra definição de seção ser 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 Centro de Identidade do AWS IAM. Para obter mais informações sobre como configurar a autenticação de login único, consulte [Usando o IAM Identity Center para autenticar o AWS SDK e as ferramentas](#). 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 função do IAM "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-ss0]
sso_region = us-east-1
sso_start_url = https://my-ss0-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 atributo está disponível para seu SDK, consulte [Support by AWS SDKs and tools](#) para ver os 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 modelo de API, 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]
```


Localizando e alterando a localização dos arquivos compartilhados, dos **credentials** arquivos **config** AWS SDKs e das ferramentas

Os **credentials** arquivos compartilhados AWS **config** e são arquivos de texto simples que contêm informações de configuração das ferramentas AWS SDKs e. 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 denominada `.aws` que é colocada na sua pasta home ou na pasta do usuário.

Sistema operacional	Local padrão e nome dos arquivos
Linux e macOS	<code>~/.aws/config</code> <code>~/.aws/credentials</code>
Windows	<code>%USERPROFILE%\aws\config</code> <code>%USERPROFILE%\aws\credentials</code>

Resolução do diretório inicial

`~` é usado para resolução do diretório inicial apenas quando:

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

Ao definir 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 das variáveis de ambiente HOMEDRIVE e HOMEPATH (\$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 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 a configuração do sistemas usando variáveis de ambiente, consulte [Usando variáveis de ambiente para configuração AWS SDKs e ferramentas globais](#).

Usar propriedade do sistema de JVM

No SDK para Kotlin em execução na JVM e no SDK para Java 2.x, você pode definir as seguintes propriedades de sistema de JVM para alterar a localização ou o nome desses arquivos do valor padrão para um valor personalizado:

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

Para obter instruções sobre como definir as propriedades e sistema de JVM, consulte [the section called “Como definir as propriedades do sistema em JVM”](#). O SDK para Java 1.x não é compatível com essas propriedades de sistema.

Usando variáveis de ambiente para configuração AWS SDKs e ferramentas globais

As variáveis de ambiente fornecem outra maneira de especificar as opções e credenciais de configuração ao usar AWS SDKs as ferramentas. As variáveis de ambiente podem ser úteis para criar scripts ou definir temporariamente um perfil nomeado como padrão. Para obter a lista das variáveis de ambiente suportadas pela maioria SDKs, consulte [Lista 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 e credentials.
- 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...zrkuWJ0gQs8IZZaIv2BXIa2R40Lgk
$ 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...zrkuWJ0gQs8IZZaIv2BXIa2R40Lgk
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...zrkuWJ0gQs8IZZaIv2BXIa2R40Lgk"
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, geralmente você pode definir variáveis de ambiente do SDK no arquivo `serverless.yml` sob a chave do provedor na 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.

Usando propriedades do sistema JVM para configurar e AWS SDK para Java AWS SDK para Kotlin

[As propriedades do sistema JVM](#) fornecem outra maneira de especificar as opções de configuração e as credenciais para SDKs execução na JVM, como a e a. AWS SDK para Java AWS SDK para Kotlin Para obter uma lista das propriedades do sistema JVM suportadas pelo SDKs, consulte Referência de [configurações](#).

Precedência de opções

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

Como definir as propriedades do sistema em JVM

Você pode definir propriedades de sistema em JV de várias maneiras.

Na linha de comando

Defina as propriedades de sistema de JVM na linha de comando ao invocar o comando `java` usando a opção `-D`. O comando a seguir configura a Região da AWS de maneira global 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 de sistema de JVM, especifique a opção `-D` várias vezes.

Com uma variável de ambiente

Se não conseguir acessar a linha de comando para invocar a JVM para executar a aplicação, você poderá usar a variável de ambiente `JAVA_TOOL_OPTIONS` para configurar as opções da linha de comando. Essa abordagem é útil em situações como a execução de uma função do AWS Lambda em runtime Java ou a execução de 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 runtime

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

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

Important

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

Uso e ferramentas de autenticação AWS SDKs e acesso

Quando você desenvolve um aplicativo AWS SDK ou usa AWS ferramentas para usar Serviços da AWS, você deve estabelecer como seu código ou ferramenta se autentica. AWS Você pode configurar o acesso programático aos AWS recursos de maneiras diferentes, dependendo do ambiente em que o código é executado e do AWS acesso disponível para você.

As opções abaixo fazem parte da [cadeia de fornecedores de credenciais](#). Isso significa que, ao configurar seus `credentials` arquivos compartilhados AWS `config` e compartilhados adequadamente, seu AWS SDK ou ferramenta descobrirá e usará automaticamente esse método de autenticação.

Escolher um método para autenticar o código da aplicação

Escolha um método para autenticar as chamadas feitas AWS pelo seu aplicativo.

Você está executando código DENTRO de um AWS service (Serviço da AWS) (como Amazon EC2, Lambda, Amazon ECS, Amazon EKS,)? CodeBuild

Se seu código for executado AWS, as credenciais poderão ser disponibilizadas automaticamente para seu aplicativo. Por exemplo, se a aplicação estiver hospedada no Amazon Elastic Compute Cloud e houver um perfil do IAM associado a esse recurso, as credenciais serão disponibilizadas automaticamente para a aplicação. Da mesma forma, se você usar contêineres do Amazon ECS ou do Amazon EKS, as credenciais definidas para o perfil do IAM poderão ser obtidas automaticamente pelo código sendo executado dentro do contêiner por meio da [cadeia de fornecedores de credenciais](#) do SDK.

O código está em uma instância do Amazon Elastic Compute Cloud?

[Usando funções do IAM para autenticar aplicativos implantados na Amazon EC2](#): use o perfil do IAM para executar sua aplicação com segurança em uma instância do Amazon EC2.

Seu código está em uma AWS Lambda função?

O Lambda cria um perfil de execução com permissões mínimas quando você [cria uma função do Lambda](#). O AWS SDK ou a ferramenta então usa automaticamente a função do IAM anexada ao Lambda em tempo de execução, por meio do ambiente de execução do Lambda.

Seu código está no Amazon Elastic Container Service (no Amazon EC2 ou no AWS Fargate Amazon ECS)?

Use o perfil do IAM para a tarefa. Você deve [criar um perfil de tarefa](#) e especificar esse perfil na [definição de tarefa do Amazon ECS](#). O SDK ou a ferramenta da AWS então usam automaticamente o perfil do IAM atribuído à tarefa no runtime por meio dos metadados do Amazon ECS.

O código está no Amazon Elastic Kubernetes Service?

Recomendamos que você use as [identidades de Pods do Amazon EKS](#).

Observação: se você achar que os [perfis do IAM para contas de serviço](#) (IRSA) podem atender melhor às suas necessidades específicas, consulte [Comparação entre o EKS Pod Identity e o IRSA](#) no Amazon EKS User Guide.

Seu código está sendo executado em AWS CodeBuild

Consulte [Uso de políticas baseadas em identidade](#) para CodeBuild

Seu código está em outro AWS service (Serviço da AWS)?

Veja o guia dedicado ao seu AWS service (Serviço da AWS). Quando você executa o código no AWS, a [cadeia de fornecedores de credenciais](#) do SDK pode obter e atualizar automaticamente as credenciais para você.

Você está criando aplicações móveis ou aplicações web baseadas em cliente?

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 identidades (IdP) externo conhecido, como Login with 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 função do IAM com permissões para usar os recursos em seu Conta da AWS.

Para saber mais sobre como configurar isto para o seu SDK ou ferramenta, consulte [Assumir uma função com identidade da web ou OpenID Connect AWS SDKs para autenticação e ferramentas](#).

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 [Usar Amazon Cognito para aplicações móveis](#) no Guia do usuário do IAM.

Você está desenvolvendo e executando o código LOCALMENTE?

Nós recomendamos [Usando credenciais do console para autenticação e ferramentas AWS SDKs](#).

Depois de um rápido fluxo de autenticação baseado em navegador, gera AWS automaticamente credenciais temporárias que funcionam em ferramentas de desenvolvimento locais, como a CLI e AWS Ferramentas da AWS para PowerShell AWS SDKs

Se você usa o Identity Center para acessar a AWS conta

Use o IAM Identity Center para autenticar o AWS SDK e as ferramentas se você já tiver acesso às AWS contas and/or necessárias para gerenciar o acesso da sua força de trabalho. Como prática recomendada de segurança, recomendamos o uso AWS Organizations com o IAM Identity Center para gerenciar o acesso em todas as suas AWS contas. Você pode criar usuários no IAM Identity Center, usar o Microsoft Active Directory, usar um provedor de identidade (IdP) SAML 2.0 ou federar individualmente seu AWS IdP às contas. Para verificar se sua região é compatível com o IAM Identity Center, consulte os endpoints e cotas do [Usando o IAM Identity Center para autenticar o AWS SDK e as ferramentas](#) IAM Identity Center na Referência geral da Amazon Web Services.

Se você estiver procurando outras formas de se autenticar

Crie um usuário do IAM com menos privilégios com permissões para `sts:AssumeRole` e entrar na sua função de destino. Em seguida, configure seu perfil para assumir uma função usando uma `source_profile` configuração para esse usuário.

Você também pode usar credenciais temporárias do IAM por meio de variáveis de ambiente ou do arquivo de AWS credenciais compartilhado. Consulte [Uso de credenciais de curto prazo para autenticação AWS SDKs e ferramentas](#).

Observação: somente em ambientes de sandbox ou de aprendizado, você pode considerar o uso de credenciais de longo prazo para autenticação AWS SDKs e ferramentas.

Esse código está sendo executado on-premises ou em uma VM híbrida/sob demanda (como, por exemplo, um servidor que lê ou grava no Amazon S3, ou o Jenkins implantando na nuvem)?

Você está usando certificados de cliente X.509?

Sim: consulte [Usando o IAM Roles Anywhere para autenticação AWS SDKs e ferramentas](#). Você pode usar o IAM Roles Anywhere para obter credenciais de segurança temporárias no IAM para cargas de trabalho, como servidores, contêineres e aplicativos executados fora do. AWS Para usar o IAM Roles Anywhere, seu workload deve usar certificados X.509.

O ambiente pode se conectar com segurança a um provedor de identidade federado (como Microsoft Entra ou Okta) para solicitar credenciais temporárias? AWS

Sim: use [Provedor de credenciais de processo](#)

Use [Provedor de credenciais de processo](#) para recuperar credenciais automaticamente no runtime. Esses sistemas podem usar uma ferramenta auxiliar ou um plug-in para obter as credenciais e podem assumir um perfil do IAM nos bastidores usando `sts:AssumeRole`.

Não: use credenciais temporárias injetadas via AWS Secrets Manager

Use credenciais temporárias injetadas via. AWS Secrets Manager Para ver opções para obter chaves de acesso de curta duração, consulte [Solicitar credenciais de segurança temporárias](#) no Guia do usuário do IAM. Para ver opções para armazenar essas credenciais temporárias, consulte [AWS teclas de acesso](#).

Você pode usar essas credenciais para recuperar com segurança permissões de aplicação mais amplas do [Secrets Manager](#), onde seus segredos de produção ou credenciais de longa duração baseadas em perfil podem ser armazenados.

Você está usando uma ferramenta de terceiros que não está disponível AWS?

Use a documentação escrita por seu provedor externo para obter a melhor orientação sobre como obter credenciais.

Se seu provedor externo não tiver fornecido a documentação, você poderá injetar credenciais temporárias com segurança?

Sim: use variáveis de ambiente e AWS STS credenciais temporárias.

Não: use as chaves de acesso estáticas armazenadas no gerenciador de segredos criptografados (último recurso).

Métodos de autenticação

Métodos 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 a aplicação estiver hospedada no Amazon Elastic Compute Cloud e houver um perfil do IAM associado a esse recurso, as credenciais serão disponibilizadas automaticamente para a aplicação. Da mesma forma, se você usar contêineres do Amazon ECS ou do Amazon EKS, as credenciais definidas para o perfil do IAM poderão ser obtidas automaticamente pelo código executado dentro do contêiner por meio da cadeia de fornecedores de credenciais do SDK.

- [Usando funções do IAM para autenticar aplicativos implantados na Amazon EC2](#): use o perfil do IAM para executar sua aplicação com segurança em uma instância do Amazon EC2.
- Você pode interagir programaticamente com o AWS uso do IAM Identity Center das seguintes formas:
 - Use [AWS CloudShell](#) para executar AWS CLI comandos a partir do 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 identidades (IdP) externo conhecido, como Login with 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 função do IAM com permissões para usar os recursos em seu Conta da AWS.

Para saber mais sobre como configurar isto para o seu SDK ou ferramenta, consulte [Assumir uma função com identidade da web ou OpenID Connect AWS SDKs para autenticação e ferramentas](#).

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 [Usar Amazon Cognito para aplicações móveis](#) no Guia do usuário do IAM.

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

- [Usando credenciais do console para autenticação e ferramentas AWS SDKs](#)— Esse recurso funciona com a interface de linha de AWS comando e com as ferramentas para PowerShell e fornece credenciais atualizáveis que funcionam em ferramentas de desenvolvimento locais, como AWS CLI, Tools for e. PowerShell AWS
- [Usando o IAM Identity Center para autenticar o AWS SDK e as ferramentas](#)— 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 Centro de Identidade do AWS IAM, 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 é compatível com o IAM Identity Center, consulte [Endpoints e cotas do Centro de Identidade do AWS IAM](#) na Referência geral da Amazon Web Services.
- [Usando o IAM Roles Anywhere para autenticação AWS SDKs e ferramentas](#)— Você pode usar o IAM Roles Anywhere para obter credenciais de segurança temporárias no IAM para cargas de trabalho, como servidores, contêineres e aplicativos executados fora do. AWS Para usar o IAM Roles Anywhere, seu workload deve usar certificados X.509.
- [Assumindo uma função com AWS credenciais para autenticação AWS SDKs e ferramentas](#)— Você pode assumir uma função do IAM para acessar temporariamente AWS recursos aos quais talvez não tivesse acesso de outra forma.
- [Usando chaves de AWS acesso para autenticação AWS SDKs e ferramentas](#)— Outras opções que podem ser menos convenientes ou aumentar o risco de segurança de seus AWS recursos.

Mais informações sobre gerenciamento de acesso

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

- [Identidades do IAM \(usuários, grupos de usuários e funções\)](#) — Entenda os fundamentos das identidades em. AWS

- [Melhores práticas de segurança no IAM](#): recomendações de segurança a serem seguidas ao desenvolver aplicativos da AWS de acordo com o [modelo de responsabilidade compartilhada](#).

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

- [Entender e obter suas credenciais AWS](#): opções de chave de acesso e práticas de gerenciamento para acesso programático e de console.

Plug-in de propagação de identidades confiáveis (TIP) do IAM Identity Center para acessar Serviços da AWS

- [Usando o plugin TIP para acessar Serviços da AWS](#)— Se você estiver criando um aplicativo para o Amazon Q Business ou outro serviço que ofereça suporte à propagação de identidade confiável e estiver usando o AWS SDK para Java ou o AWS SDK para JavaScript, poderá usar o plug-in TIP para uma experiência de autorização simplificada.

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 Q 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
- [CodeCatalystconceitos - ID do builder AWS](#) no Guia CodeCatalyst do usuário da Amazon — Saiba como CodeCatalyst usa um ID do builder AWS.

Usando credenciais do console para autenticação e ferramentas AWS SDKs

Usar as credenciais do console é o método recomendado para fornecer AWS credenciais ao desenvolver um AWS aplicativo em seu ambiente local ou em outros ambientes de serviços que não sejam de AWS computação. Se você estiver desenvolvendo em um AWS recurso, como o Amazon

Elastic Compute Cloud (Amazon EC2) AWS CloudShell ou, recomendamos que você obtenha credenciais desse serviço.

Você também pode se autenticar por meio do IAM Identity Center [Usando o IAM Identity Center para autenticar o AWS SDK e as ferramentas](#). Essa opção é uma forma comum de as organizações gerenciarem o acesso de sua força de trabalho e exige que o Identity Center esteja ativado.

Como funciona?

O [login para desenvolvimento AWS local usando as credenciais do console](#) permite que você use suas credenciais de login existentes do AWS Management Console para acesso programático aos serviços. AWS Depois de um fluxo de autenticação baseado em navegador, AWS gera credenciais temporárias que funcionam em ferramentas de desenvolvimento locais, como AWS CLI, Tools for e. PowerShell AWS SDKs Esse recurso simplifica o processo de configuração e gerenciamento de credenciais de AWS CLI, especialmente se você preferir a autenticação interativa em vez de gerenciar chaves de acesso de longo prazo.

Com esse processo, você pode se autenticar usando suas credenciais raiz criadas durante a configuração inicial da conta, usuários do IAM ou uma identidade federada do seu provedor de identidade.

Se você usar SDKs para desenvolvimento, os clientes do SDK usarão as credenciais temporárias por meio do. [AWS SDKs e ferramentas: provedores de credenciais padronizados](#) Você também pode configurar [Provedor de credenciais de login](#) o.

A autenticação por meio do comando de login é suportada pela AWS CLI e pelas ferramentas para: PowerShell

- [Faça login para desenvolvimento AWS local usando as credenciais do console](#)
- [Faça login usando as credenciais do console](#) no guia do Ferramentas da AWS para PowerShell usuário

Usando o IAM Identity Center para autenticar o AWS SDK e as ferramentas

Centro de Identidade do AWS IAM pode ser usado para fornecer AWS credenciais ao desenvolver um AWS aplicativo em ambientes de serviços não AWS computacionais. Se você estiver

desenvolvendo em um AWS recurso, como o Amazon Elastic Compute Cloud (Amazon EC2) AWS Cloud9 ou, recomendamos que você obtenha credenciais desse serviço.

Use a autenticação do IAM Identity Center se você já usa o Identity Center para acessar a AWS conta ou precisa gerenciar o acesso de uma organização.

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

- O portal de AWS acesso é o local da web em que você faz login manualmente no IAM Identity Center. O formato da 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 SDK/tool autenticação.
- O AWS CLI é usado para configurar seu SDK ou ferramenta para usar a autenticação do IAM Identity Center para chamadas de API feitas pelo seu código. Esse processo único atualiza seu AWS config arquivo compartilhado, que é usado pelo SDK ou pela ferramenta quando você executa o código.

Pré-requisitos

Antes de iniciar este procedimento, você deverá ter concluído o seguinte:

- Se você não tiver um Conta da AWS, [inscreva-se em um Conta da AWS](#).
- Se você ainda não o fez, [habilite o IAM Identity Center](#) seguindo as instruções do Centro de Identidade do AWS IAM User Guide.

Configure o acesso programático usando o Centro de Identidade do IAM

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

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

Não estabeleci acesso por meio do IAM Identity Center

1. Adicione um usuário e permissões administrativas seguindo o procedimento [Configure user access with the default IAM Identity Center directory](#) do Centro de Identidade do AWS IAM User Guide.
2. O conjunto de permissões `AdministratorAccess` não deve ser usado para desenvolvimento normal. Em vez disso, recomendamos usar o conjunto predefinido de permissões `PowerUserAccess`, a menos que seu empregador tenha criado um conjunto de permissões personalizado com essa finalidade.

Siga o mesmo procedimento [Configure user access with the default IAM Identity Center directory](#), mas desta vez:

- Em vez de criar o grupo *Admin team*, crie um grupo *Dev team* e faça essa substituição nas instruções de agora em diante.
- Você pode usar o usuário existente, mas o usuário deve ser adicionado ao novo grupo *Dev team*.
- Em vez de criar o grupo permissões *AdministratorAccess*, crie um grupo *PowerUserAccess* e faça essa substituição nas instruções de agora em diante.

Quando terminar, você deverá ter seguinte:

- Um grupo `Dev team`.
 - Um conjunto de permissões `PowerUserAccess` anexado ao grupo `Dev team`.
 - O usuário foi adicionado ao grupo `Dev team`.
3. Saia do portal e entre novamente para ver suas opções Contas da AWS e para `Administrator` ou `PowerUserAccess`. Selecione `PowerUserAccess` ao trabalhar com a 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 esse URL ou você pode recebê-lo em um e-mail seguindo a Etapa 1: Estabelecer acesso. Caso contrário, encontre a URL do seu portal de AWS acesso 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. Os perfis do usuário serão exibidos. Escolha Chaves de acesso para obter credenciais de acesso programático ou por linha de comando para o conjunto apropriado de permissões. 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. Selecione o método Credenciais IAM Identity Center para obter os valores `Issuer URL` e `SSO Region` necessários para a próxima etapa. Observação: `SSO Start URL` pode ser usado de modo 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. No prompt `SSO Start URL`, insira o valor obtido para `Issuer URL`.
 - b. Para o nome do perfil CLI, 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 seu SDK em seu ambiente de desenvolvimento.
 - a. Para alguns SDKs, pacotes adicionais, como `SSO` e `SSO0IDC` devem ser adicionados ao seu aplicativo antes que você possa usar a autenticação do IAM Identity Center. Para obter detalhes, consulte seu SDK específico.
 - b. Se você configurou anteriormente o acesso ao AWS, revise o `AWS credentials` arquivo compartilhado para verificar se há algum [AWS teclas de acesso](#). Você deve remover todas as credenciais estáticas antes que o SDK ou a ferramenta usem as credenciais do IAM Identity Center devido à precedência [Entender a cadeia de provedores de credenciais](#).

Para saber mais sobre como as ferramentas SDKs e usam e atualizam as credenciais usando essa configuração, consulte. [Como a autenticação do IAM Identity Center é resolvida AWS SDKs e as ferramentas](#)

Para definir as configurações do provedor do IAM Identity Center diretamente no arquivo `config` compartilhado, consulte [Provedor de credencial do IAM Identity Center](#) neste guia.

Atualizar sessões de acesso ao portal

Seu acesso acabará expirando e o SDK ou a ferramenta encontrarão um erro de autenticação. Quando essa expiração ocorrerá depende da duração da sessão configurada. 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 do 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 saber mais, consulte os seguintes tópicos no Manual do usuário do Centro de Identidade do AWS IAM :

- Duração da sessão do IAM Identity Center: [configure a duração das sessões do portal de acesso da AWS de seus usuários](#)
- Permissão definir duração da sessão: [definir duração da sessão](#)

Como a autenticação do IAM Identity Center é resolvida AWS SDKs e as ferramentas

Termos relevantes do Centro de Identidade do IAM

Os termos a seguir ajudam você a entender o processo e a configuração por trás do Centro de Identidade do AWS IAM. A documentação do AWS SDK APIs 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	Description
Identity Center	sso	Embora o AWS Single Sign-On tenha sido renomeado, os namespaces da sso API manterão seu nome original para fins de compatibilidade com versões anteriores. Para obter mais informações, consulte Renomear o

Nome do IAM Identity Center	Nome da API do SDK	Description
		IAM Identity Center no Guia do usuário do Centro de Identidade do AWS IAM .
Console do IAM Identity Center Console administrativo		O console que você usa para configurar o single sign-on.
AWS URL do portal de acesso		Um URL exclusivo para sua conta do IAM Identity Center, como <code>https://xxx.awsapps.com/start</code> . 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	Fornecer 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 AWS service (Serviço da AWS) chamadas. Em discussões informais, você pode ver isso incorretamente chamado de “sessão de funções”.

Nome do IAM Identity Center	Nome da API do SDK	Description
Credenciais do conjunto de permissões	AWS credenciais credenciais sigv4	As credenciais que o SDK realmente usa para a maioria das AWS service (Serviço da AWS) chamadas (especificamente, todas as chamadas sigv4 AWS service (Serviço da AWS)). 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 Serviços da AWS maioria é sigv4 APIs, com algumas exceções, como e. Amazon CodeWhisperer Amazon CodeCatalyst A seguir, descrevemos o processo de resolução de credenciais para dar suporte à maioria das AWS service (Serviço da AWS) chamadas para o código do seu aplicativo por meio Centro de Identidade do AWS IAM de.

Iniciar uma sessão do portal de AWS acesso

- Inicie o processo entrando na sessão com suas credenciais.
 - Use o `aws sso login` comando 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. Ele 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 AWS CLI comando 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 AWS acesso de seus usuários](#) no Guia do Centro de Identidade do AWS IAM usuário.

- 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)

SDKs forneça acesso a Serviços da AWS quando você instancia um objeto cliente por serviço. Quando o perfil selecionado do AWS config arquivo compartilhado 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 credenciais para sigv4 APIs 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 AWS acesso ao SDK assumindo uma função 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 do Centro de Identidade do AWS IAM .
- Lembre-se de que as credenciais do conjunto de permissões também são chamadas de credenciais e AWS credenciais sigv4 na maioria das documentações da API do SDK. AWS

As credenciais do conjunto de permissões são retornadas de uma chamada [getRoleCredentials](#) da API do IAM Identity Center para o SDK. O objeto cliente do SDK usa essa função assumida do IAM 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 [Configuração do provedor de token do SSO](#), 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 serço IAM Identity Center. Esse token é concedido se sua sessão do portal de acesso ao IAM Identity Center não tiver expirado.

Usando o IAM Roles Anywhere para autenticação AWS SDKs e ferramentas

Você pode usar o IAM Roles Anywhere para obter credenciais de segurança temporárias no IAM para cargas de trabalho, 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 uma carga de trabalho 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, a função do IAM e o perfil do IAM Roles Anywhere, consulte [Como criar uma âncora de confiança e um perfil em AWS Identity and Access Management 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 AWS config arquivo 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 auxiliar de credenciais, consulte [Obter credenciais de segurança temporárias do AWS Identity and Access Management Roles Anywhere no Guia](#) do usuário do IAM Roles Anywhere.

Para usar credenciais de segurança temporárias do IAM Roles Anywhere with AWS SDKs and the AWS CLI, você pode definir a `credential_process` configuração no AWS config arquivo compartilhado. O SDKs e AWS CLI suporta 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 Console de gerenciamento da AWS. 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 [IAM Roles Anywhere Credential Helper](#) on. 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.

Assumindo uma função com AWS credenciais para autenticação

AWS SDKs e ferramentas

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 as solicitações de API AWS Security Token Service (AWS STS), consulte [Ações](#) na Referência da API do AWS Security Token Service .

Para configurar seu SDK ou ferramenta para assumir um perfil, você deve primeiro criar ou identificar um perfil específico a ser assumido. Os perfis do IAM são identificados exclusivamente por um perfil do nome do recurso da Amazon ([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 obter mais informações sobre perfis do IAM, consulte [Perfis do IAM](#) no Guia do usuário do IAM.

Depois que perfil do IAM for identificado, se você tiver a confiança desse perfil, poderá configurar seu SDK ou ferramenta para usar as permissões concedidas pelo perfil.

Note

É uma prática AWS recomendada usar endpoints regionais sempre que possível e configurar seus [Região da AWS](#).

Assumir um perfil do IAM

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. O uso mais comum desse método de assumir um perfil é quando você tem credenciais da AWS para uma conta, mas a aplicação precisa acessar recursos em outra conta.

Etapa 1: Configurar um perfil do IAM

Para configurar seu SDK ou ferramenta para assumir um perfil, você deve primeiro criar ou identificar um perfil específico a ser assumido. Os perfis do IAM são identificados exclusivamente usando um [ARN](#) de perfil. Os perfis estabelecem relações de confiança com outra entidade, normalmente dentro da sua conta ou para acesso entre contas. Para saber mais, consulte [Criar perfis do IAM](#) no Guia do usuário do IAM.

Etapa 2: Configurar o SDK ou a ferramenta

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

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

Assumir uma função com identidade da web ou OpenID Connect

AWS SDKs para autenticação e ferramentas

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 as solicitações de API AWS Security Token Service (AWS STS), consulte [Ações](#) na Referência da API do AWS Security Token Service .

Para configurar seu SDK ou ferramenta para assumir um perfil, você deve primeiro criar ou identificar um perfil específico a ser assumido. Os perfis do IAM são identificados exclusivamente por um perfil do nome do recurso da Amazon ([ARN](#)). Os perfis estabelecem as relações de confiança com uma outra entidade. A entidade de confiança que usa o perfil pode ser um provedor de identidades de web ou OpenID Connect (OIDC), ou uma federação SAML. Para saber mais sobre perfis do IAM, consulte [Métodos para assumir um perfil](#) no Guia do usuário do IAM.

Depois que a função do IAM for configurada em seu SDK, se essa função estiver configurada para confiar em seu provedor de identidade, você poderá configurar ainda mais seu SDK para assumir essa função a fim de obter credenciais temporárias AWS .

Note

É uma prática AWS recomendada usar endpoints regionais sempre que possível e configurar seus [Região da AWS](#).

Federar com identidade da Web ou OpenID Connect

Você pode usar os JSON Web Tokens (JWTs) de provedores de identidade públicos, 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 o protocolo de descoberta do OIDC, como EntraId ou. PingFederate

Se você estiver usando o Amazon Elastic Kubernetes Service, esse atributo permite especificar diferentes perfis do IAM para cada uma das suas contas de serviço no cluster do Amazon EKS.

Esse recurso do Kubernetes é distribuído JWTs para seus pods, que são usados por esse provedor de credenciais para obter credenciais temporárias. AWS Para obter mais informações sobre essa configuração do Amazon EKS, consulte [Perfis do IAM para contas de serviço](#) no Guia do usuário do Amazon EKS. No entanto, como uma opção mais simples, recomendamos que você use o [Amazon EKS Pod Identities](#) se seu [SDK for compatível](#).

Etapa 1: Configurar um provedor de identidades e um perfil do IAM

Para configurar a federação com um IdP externo, use um provedor de identidade do IAM para informar AWS sobre o IdP externo e sua configuração. Isso estabelece confiança entre o seu Conta da AWS e o IdP externo. Antes de configurar o SDK para usar o JSON Web Token (JWT), você deve primeiro configurar o provedor de identidades (IdP) e o perfil do IAM usado para acessá-lo. Para configurá-los, consulte [Criar um perfil para identidades da Web ou federação do OpenID Connect \(console\)](#) no Guia do usuário do IAM.

Etapa 2: Configurar o SDK ou a ferramenta

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

Quando você especifica isso em um perfil, o SDK ou a ferramenta faz automaticamente a chamada de AWS STS [AssumeRoleWithWebIdentity](#) API 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`: a partir do perfil do IAM que você criou na Etapa 1
- `web_identity_token_file`: do IdP externo
- (Optional) `duration_seconds`
- (Optional) `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 a API do Amazon Cognito, inclua o cliente do serviço Amazon Cognito na compilação ou nas bibliotecas do seu SDK ou ferramenta. 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](#).

Usando chaves de AWS acesso para autenticação AWS SDKs e ferramentas

Usar chaves de AWS acesso é uma opção para autenticação ao usar AWS SDKs ferramentas.

Use credenciais de curto prazo

Recomendamos configurar o seu SDK ou ferramenta para usar [Usando o IAM Identity Center para autenticar o AWS SDK e as ferramentas](#) para usar as opções de duração de sessão estendida.

No entanto, para configurar diretamente as credenciais temporárias do SDK ou da ferramenta, consulte [Usando credenciais de curto prazo para autenticação e ferramentas AWS SDKs](#).

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 [Centro de Identidade do AWS IAM](#).

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 AWS Usando 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 da 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 na AWS](#).

Garanta chaves de acesso para usuários do IAM

Você deve alternar chaves de acesso de usuário do IAM regularmente. Siga as orientações em [Alternar 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 ajudar 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 [Usando credenciais de longo prazo para autenticação e ferramentas AWS SDKs](#).

Usando credenciais de curto prazo para autenticação e ferramentas AWS SDKs

Recomendamos configurar seu AWS SDK ou ferramenta para uso [Usando o IAM Identity Center para autenticar o AWS SDK e as ferramentas](#) 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 a aplicação 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 da cadeia de fornecedores de credenciais, como [Usando o IAM Identity Center para autenticar o AWS SDK e as ferramentas](#).
- Ou use [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>
```


Se você usa um usuário do IAM para executar seu código, o SDK ou a ferramenta em seu ambiente de desenvolvimento é autenticado usando credenciais de usuário do IAM de longo prazo no arquivo compartilhado. `AWS credentials` Analise o tópico [Melhores práticas de segurança no IAM](#) e faça a transição para o IAM Identity Center ou outras credenciais temporárias assim que possível.

Avisos e orientações importantes para credenciais

Avisos para credenciais

- NÃO use as credenciais de raiz da sua conta para acessar os recursos da AWS . Estas credenciais fornecem acesso ilimitado à conta e são difíceis de revogar.
- NÃO coloque chaves de acesso literais ou informações de credenciais em arquivos de aplicações. Se colocar, criará um risco de exposição acidental das credenciais se, por exemplo, fizer upload do projeto em um repositório público.
- NÃO inclua 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 [perfis do IAM para tarefas](#) do Amazon Elastic Container Service (Amazon ECS).
- Use [perfis do IAM](#) para aplicações em execução nas instâncias do Amazon EC2.

Pré-requisitos: Crie uma conta AWS

Para usar um usuário do IAM 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 Gerenciamento de contas 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 do usuário do IAM.

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 [Aplicar permissões de privilégio mínimo](#), no Guia do usuário do IAM.


Etapa 1: criar o usuário do IAM

- Crie o usuário do IAM seguindo o procedimento de [Criação de usuários do IAM \(console\)](#) no Guia do usuário do IAM. Ao criar o usuário do IAM:
 - Recomendamos que você selecione Fornecer acesso de usuário ao Console de gerenciamento da AWS. Isso permite visualizar os Serviços da AWS relacionados ao código que você está executando em um ambiente visual, como verificação de logs de diagnóstico do AWS CloudTrail ou upload de arquivos para o Amazon Simple Storage Service, que é útil ao depurar código.
 - Em Definir permissões: Opções de permissão, selecione Anexar políticas diretamente para definir como você deseja atribuir permissões a esse usuário.
 - A maioria dos tutoriais de “Conceitos básicos” do SDK 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 deste procedimento em relação à definição de limites de permissão ou tags.

Etapa 2: obter as chaves de acesso

1. No painel de navegação do console do IAM, selecione Usuários e depois selecione o **User name** do 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 comandos (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 arquivo .csv com a chave de acesso e a chave de acesso secreta do usuário do IAM. 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 teclas de acesso](#) para ver os nomes das variáveis de ambiente. Essa é uma forma de 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 funções do IAM para autenticar aplicativos implantados na Amazon EC2

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 do Amazon Elastic Compute Cloud.

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

Criar um perfil do IAM

O aplicativo AWS SDK que você desenvolve provavelmente acessa pelo menos um AWS service (Serviço da AWS) para realizar ações. Crie um perfil do IAM que conceda as permissões necessárias para a execução da aplicação.

Esse procedimento cria um perfil que concede acesso somente leitura ao Amazon S3, por exemplo. Muitos dos guias do AWS SDK têm tutoriais de “introdução” que são lidos no Amazon S3.

1. Faça login no Console de gerenciamento da AWS e abra o console do IAM 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 EC2 e 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 função do IAM

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

- Siga [Execute rapidamente uma instância](#) no Guia do EC2 usuário da Amazon. Porém, antes da etapa de envio final, faça também o seguinte:
 - Em Detalhes avançados, em Perfil de instância do IAM, escolha o perfil criado na etapa anterior.

Com essa EC2 configuração do IAM e da 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 que usou em Par de chaves (login) quando criou a instância, ou seja, o arquivo PEM.

Você pode fazer isso seguindo as orientações para o tipo de instância: [Conectar-se à instância do Linux](#) ou [Conectar-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 do Linux ou do macOS, você pode usar o comando secure copy para copiar a aplicação. Para usar scp com um par de chaves, 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. Inicie o aplicativo e verifique se ele é executado com os mesmos resultados da sua máquina de desenvolvimento.
3. (Opcional) Verifique se o aplicativo usa as credenciais fornecidas pelo perfil do IAM.

- a. Faça login no Console de gerenciamento da AWS e abra o EC2 console da Amazon em <https://console.aws.amazon.com/ec2/>.
- b. Selecione a instância.
- c. Escolha Ações, Segurança e, depois, Modificar perfil do IAM.
- d. Em Perfil do IAM, desanexe o perfil do IAM escolhendo Nenhum perfil do IAM.
- e. Escolha Atualizar perfil do IAM.
- f. Execute o aplicativo novamente e confirme se ele retorna um erro de autorização.

Usando o plugin TIP para acessar Serviços da AWS

A propagação de identidade confiável (TIP) é um recurso Centro de Identidade do AWS IAM que permite que os administradores concedam permissões com base nos atributos do usuário, como associações de grupos. Serviços da AWS Com a propagação de identidade confiável, o contexto de identidade é adicionado a uma função do IAM para identificar o usuário que está solicitando acesso aos AWS recursos. Esse contexto é propagado para outros Serviços da AWS.

O contexto de identidade compreende informações que são Serviços da AWS usadas para tomar decisões de autorização ao receber solicitações de acesso. Essas informações incluem metadados que identificam o solicitante (por exemplo, um usuário do IAM Identity Center), o acesso AWS service (Serviço da AWS) ao qual o acesso é solicitado (por exemplo, Amazon Redshift) e o escopo do acesso (por exemplo, acesso somente para leitura). O destinatário AWS service (Serviço da AWS) usa esse contexto e todas as permissões atribuídas ao usuário para autorizar o acesso aos seus recursos. Para obter mais informações, consulte a [visão geral da propagação de identidade confiável](#) no Guia do Centro de Identidade do AWS IAM usuário.

O plug-in TIP pode ser usado com Serviços da AWS esse suporte à propagação confiável de identidade. Como caso de uso de referência, consulte [Configuring an Amazon Q Business application using Centro de Identidade do AWS IAM](#) no Amazon Q Business User Guide.

Note

Se você estiver usando o Amazon Q Business, consulte [Configuring an Amazon Q Business application using Centro de Identidade do AWS IAM](#) para obter instruções específicas para o serviço.

Pré-requisitos para o uso do plug-in TIP

Os seguintes recursos são necessários para o plug-in funcionar:

1. Você deve estar usando o AWS SDK para Java ou AWS SDK para JavaScript o.
2. Verifique se o serviço que você está usando é compatível com propagação de identidades confiáveis.

Consulte a coluna Enables trusted identity propagation through IAM Identity Center da tabela [AWS managed applications that integrate with IAM Identity Center](#) no Centro de Identidade do AWS IAM User Guide.

3. Habilite o IAM Identity Center e a propagação de identidades confiáveis.

Consulte [TIP prerequisites and considerations](#) no Centro de Identidade do AWS IAM User Guide.

4. Você deve ter um Identity-Center-integrated aplicativo.

Consulte [AWS managed applications](#) ou [Customer managed applications](#) no Centro de Identidade do AWS IAM User Guide.

5. Você deve configurar um emissor de token confiável (TTI) e conectar seu serviço ao IAM Identity Center.

Consulte [Prerequisites for trusted token issuers](#) e [Tasks for setting up a trusted token issuer](#) no Centro de Identidade do AWS IAM User Guide.

Para usar o plug-in TIP no código

1. Crie uma instância do plug-in de propagação de identidades confiáveis.
2. Crie uma instância de cliente de serviço para interagir com você AWS service (Serviço da AWS) e personalize o cliente de serviço adicionando o plug-in confiável de propagação de identidade.

O plug-in TIP aceita os seguintes parâmetros de entrada:

- **webTokenProvider**: uma função que o cliente implementa para obter um token do OpenID do respectivo provedor de identidades externo.
- **accessRoleArn**: o ARN do perfil do IAM a ser assumido pelo plug-in com o contexto de identidade do usuário para obter as credenciais aprimoradas com identidade.

- **applicationArn**: string do identificador específico do cliente ou da aplicação. Esse valor é um ARN do aplicativo que tem OAuth concessões configuradas.
- **ssoOidcClient**: (Opcional) Um cliente OIDC SSO, como [SsoOidcClient](#) para Java ou for JavaScript, com configurações [client-sso-oidc](#) definidas pelo cliente. Se não for fornecido, um cliente do OIDC usando o `applicationRoleArn` será instanciado e usado.
- **stsClient**: (opcional) um cliente do AWS STS com configurações definidas pelo cliente, usado para assumir o `accessRoleArn` com o contexto da identidade do usuário. Se não for fornecido, um AWS STS cliente usando `applicationRoleArn` será instanciado e usado.
- **applicationRoleArn**: (Opcional) O ARN da função do IAM a ser assumido para que o OIDC e os AWS STS clientes `AssumeRoleWithWebIdentity` possam ser inicializados.
 - Se não forem fornecidos, ambos os parâmetros `ssoOidcClient` e `stsClient` deverão ser fornecidos.
 - Se fornecido, o `applicationRoleArn` não poderá ser igual ao valor do parâmetro `accessRoleArn`. `applicationRoleArn` é usado para compilar o `STSCient`, que é usado para assumir o `accessRole`. Se o mesmo papel for usado para ambos `applicationRole` e `accessRole`, isso significaria usar um papel para assumir a si mesmo (suposição de papel próprio), o que é desencorajado por. AWS Consulte [announcement](#) para obter mais detalhes.

Considerações sobre os parâmetros **ssoOidcClient**, **stsClient** e **applicationRoleArn**

Ao configurar o plug-in TIP, considere os seguintes requisitos de permissão com base nos parâmetros que você fornecer:

- Se você fornecer `ssoOidcClient` e `stsClient`:
 - As credenciais no `ssoOidcClient` devem ter permissão de `oauth:CreateTokenWithIAM` para chamar a central de identidade para obter o contexto do usuário específico da central de identidade.
 - As credenciais no `stsClient` devem ter as permissões `sts:AssumeRole` e `sts:SetContext` no `accessRole`. O `accessRole` também precisa ser configurado com uma relação de confiança com as credenciais no `stsClient`.
- Se você estiver fornecendo `applicationRoleArn`:
 - O `applicationRole` deve ter as permissões `oauth:CreateTokenWithIAM`, `sts:AssumeRole` e `sts:SetContext` nos recursos necessários (instância do IdC, `accessRole`), pois ele será usado para compilar os clientes do OIDC e do STS.

- `applicationRole` deve ter uma relação de confiança com o provedor de identidade usado para gerar `webToken`, pois `webToken` será usado para assumir o `ApplicationRole` por meio da [AssumeRoleWithWebIdentity](#) chamada do plug-in.

Exemplo `ApplicationRole` de configuração:

Política de confiança com o provedor de tokens Web:

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

Política de permissão:

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

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

Exemplos de código usando TIP

Os exemplos abaixo mostram como implementar o plug-in TIP em seu código usando o AWS SDK para Java ou AWS SDK para JavaScript o.

Java

Para usar o plug-in TIP em seu AWS SDK para Java projeto, você precisa declará-lo como uma dependência no arquivo do `pom.xml` seu projeto.

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

No código-fonte, inclua a instrução de pacote necessária para `software.amazon.awssdk.trustedidentitypropagation`.

Os exemplos a seguir mostram duas maneiras de criar uma instância do plug-in de propagação de identidades confiáveis e adicioná-lo a um cliente de serviço. Ambos os exemplos usam o Amazon S3 como serviço e o utilizam `S3AccessGrantsPlugin` para gerenciar permissões específicas do usuário, mas podem ser aplicados a qualquer um AWS service (Serviço da AWS) que ofereça suporte à propagação de identidade confiável (TIP).

Note

Para esses exemplos, você precisa configurar as permissões específicas do usuário de Concessão de Acesso do S3. Consulte a [documentação de Concessão de Acesso do S3](#) para obter mais detalhes.

Opção 1: criar e transmitir clientes do OIDC e do STS

```
Sso0idcClient oidcClient = Sso0idcClient.builder()
    .region(Region.US_EAST_1)
    .credentialsProvider(credentialsProvider).build();

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

TrustedIdentityPropagationPlugin trustedIdentityPropagationPlugin =
    TrustedIdentityPropagationPlugin.builder()
        .webTokenProvider(() -> webToken)
        .applicationArn(idcApplicationArn)
        .accessRoleArn(accessRoleArn)
        .sso0idcClient(oidcClient)
        .stsClient(stsClient)
        .build();

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

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

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

Opção 2: passar applicationRoleArn e adiar a criação do cliente para o plug-in

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

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

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

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

Para obter detalhes adicionais e fontes, consulte [trusted-identity-propagation-java](#) em GitHub.

JavaScript

Execute o comando a seguir para instalar o pacote do plug-in de autenticação TIP em seu AWS SDK para JavaScript projeto:

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

O package . json final deve incluir uma dependência semelhante à seguinte:

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

No seu código-fonte, importe a dependência de `TrustedIdentityPropagationExtension` necessária.

Os exemplos a seguir mostram duas maneiras de criar uma instância do plug-in de propagação de identidades confiáveis e adicioná-lo a um cliente de serviço. Ambos os exemplos usam o Amazon S3 como serviço e utilizam os Amazon S3 Access Grants para gerenciar permissões específicas do usuário, mas podem ser aplicados a AWS service (Serviço da AWS) qualquer um que ofereça suporte à propagação de identidade confiável (TIP).

Note

Para esses exemplos, você precisa configurar as permissões específicas do usuário de Concessão de Acesso do Amazon S3. Consulte a documentação de [Concessão de Acesso do Amazon S3](#) para obter mais detalhes.

Opção 1: criar e transmitir clientes do OIDC e do STS

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

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

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

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

  const credentials = response.Credentials;

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

  // Use the temporary S3 client to perform the operation
  const s3Params = {
    Bucket: "BUCKET_NAME",
    Key: "S3_OBJECT_KEY",
  };
  const getObjectCommand = new GetObjectCommand(s3Params);
  const s3Object = await temporaryS3Client.send(getObjectCommand);

  const fileContent = await s3Object.Body.transformToString();

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

Opção 2: passar applicationRoleArn e adiar a criação do cliente para o plug-in

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

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

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

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

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

  const credentials = response.Credentials;

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

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

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

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

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

Para obter detalhes adicionais e fontes, consulte [trusted-identity-propagation-jsem](#) em GitHub.

AWS SDKs referência de configurações e ferramentas

SDKs forneça um idioma específico para APIs . Serviços da AWS Eles cuidam de parte do trabalho pesado necessário para fazer chamadas de API 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 ajustes de configuração nas seguintes seções:

- [AWS SDKs e ferramentas: provedores de credenciais padronizados](#)— Provedores de credenciais comuns padronizados em vários SDKs
- [AWS SDKs e ferramentas, recursos padronizados](#)— Recursos comuns padronizados em vários SDKs.

Criar clientes de serviço

Para acessar programaticamente Serviços da AWS, SDKs use um cliente class/object para cada um. AWS service (Serviço da AWS) Por exemplo, se seu aplicativo precisa acessar a Amazon EC2, 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 vezes SDKs, 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 Ferramentas da AWS

para PowerShell, eles assumem a forma de parâmetros por operação que você insere na linha de comando. Para um SDK, 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 uma API individual.

2. Java/Kotlin apenas: a propriedade do sistema de JVM para a configuração é verificada. Se estiver definida esse valor será usado para configurar o cliente.
3. A variável de ambiente está marcada. Se estiver definida esse valor será usado para configurar o cliente.
4. O SDK verifica a configuração no arquivo compartilhado `credentials`. Se estiver definida, o cliente a usará.
5. O arquivo compartilhado `config` para a configuração. Se a configuração estiver presente, o SDK a usará.
 - A variável de ambiente `AWS_PROFILE` ou a propriedade do sistema de JVM `aws.profile` podem ser usadas para personalizar o perfil que o SDK carrega.
6. Qualquer valor padrão fornecido pelo próprio código-fonte do SDK 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, o AWS SDK para .NET suporta uma fonte adicional chamada [SDK Store](#). Para obter mais informações sobre provedores exclusivos de um SDK ou ferramenta, consulte o guia específico do SDK ou da ferramenta que você está usando.

A ordem determina quais métodos têm precedência e substituem outros. Por exemplo, se você configurar um perfil no arquivo `config` compartilhado, ele só será encontrado e usado depois que o SDK ou a ferramenta verificarem 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 da API ou parâmetro da linha de comando da AWS CLI) ou no código substituiria todos os outros valores desse comando.

Entender as páginas de configurações deste guia

As páginas na seção Referência de configurações deste guia detalham as configurações disponíveis que podem ser definidas por vários mecanismos. As tabelas a seguir listam as configurações do arquivo de configuração e credencial, as variáveis de ambiente e (para Java e Kotlin SDKs) as configurações da JVM 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](#)
- [Lista de propriedades de sistema de JVM](#)

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

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

Por exemplo, vejamos uma configuração da página do atributo [Compactação de solicitações](#).

As informações do exemplo de configuração `disable_request_compression` documentam o seguinte:

- Há três maneiras equivalentes de controlar a compactação de solicitações fora da base de código. Você também pode:
 - Defini-la no arquivo de configuração usando `disable_request_compression`
 - Defini-la como uma variável de ambiente usando `AWS_DISABLE_REQUEST_COMPRESSION`
 - Ou, se você estiver usando o SDK do Java ou Kotlin, defina-a como uma propriedade do sistema de JVM usando `aws.disableRequestCompression`

Note

Pode haver também uma maneira de configurar a mesma funcionalidade diretamente no código, mas esta referência não trata disso, pois ela é específica para cada SDK. Se você quiser definir a configuração no código em si, consulte o guia específico do SDK ou a referência de API.

- 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 tabela de ferramentas AWS SDKs e Support by.

Essa tabela mostra se o SDK é compatível com as configurações listadas na página. A coluna `Supported` indica o nível de compatibilidade com os seguintes valores:

- `Yes`: as configurações são totalmente compatíveis com o SDK conforme escrito.
- `Partial`: algumas das configurações são compatíveis ou o comportamento se desvia da descrição. Para `Partial`, uma nota adicional indica o desvio.
- `No`: nenhuma das configurações é compatível. Isso não significa que a mesma funcionalidade pode ser alcançada em código, indica apenas que as configurações externas listadas não são compatíveis.

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 Serviços da AWS. SDKs e as ferramentas também podem oferecer suporte a configurações e variáveis de ambiente exclusivas. Para ver as configurações e as variáveis de ambiente compatíveis apenas com um SDK ou uma ferramenta individual, consulte o guia específica do SDK ou da ferramenta.

Nome da configuração	Detalhes
account_id_endpoint_mode	Endpoints baseados em contas
api_versions	Definições gerais de configuração
auth_scheme_preference	Esquema de autenticação
aws_access_key_id	AWS teclas de acesso
aws_account_id	Endpoints baseados em contas
aws_secret_access_key	AWS teclas de acesso
aws_session_token	AWS teclas 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_host_prefix_injection	Injeção de prefixo de host

Nome da configuração	Detalhes
<code>disable_request_compression</code>	Compactação de solicitações
<code>duration_seconds</code>	Assuma a função de provedor de credenciais
<code>ec2_metadata_service_endpoint</code>	Provedor de credenciais IMDS
<code>ec2_metadata_service_endpoint_mode</code>	Provedor de credenciais IMDS
<code>ec2_metadata_v1_disabled</code>	Provedor de credenciais IMDS
<code>endpoint_discovery_enabled</code>	Descoberta de endpoint
<code>endpoint_url</code>	Endpoints específicos de serviço
<code>external_id</code>	Assuma a função de provedor de credenciais
<code>ignore_configured_endpoint_urls</code>	Endpoints específicos de serviço
<code>max_attempts</code>	Comportamento de repetição

Nome da configuração	Detalhes
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
output	Definições gerais de configuração
parameter_validation	Definições gerais de configuração
region	Região da AWS
request_checksum_calculation	Proteções de integridade de dados para o Amazon S3
request_min_compression_size_bytes	Compactação de solicitações
response_checksum_validation	Proteções de integridade de dados para o Amazon S3
retry_mode	Comportamento de repetição
role_arn	Assuma a função de provedor de credenciais
role_session_name	Assuma a função de provedor de credenciais

Nome da configuração	Detalhes
s3_disable_express_session_auth	Autenticação de sessão da S3 Express One Zone
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
sigv4a_signing_region_set	Esquema de autenticação
source_profile	Assuma a função de provedor de credenciais
sso_account_id	Provedor de credenciais do IAM Identity Center
sso_region	Provedor de credenciais do IAM Identity Center
sso_registration_scopes	Provedor de credenciais do IAM Identity Center
sso_role_name	Provedor de credenciais do IAM Identity Center
sso_start_url	Provedor de credenciais do IAM Identity Center
sts_regional_endpoints	AWS STS Endpoints regionais
use_dualstack_endpoint	Endpoints de pilha dupla e FIPS

Nome da configuração	Detalhes
use_fips_endpoint	Endpoints de pilha dupla e 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 Serviços da AWS. SDKs e as ferramentas também podem oferecer suporte a configurações e variáveis de ambiente exclusivas. Para ver as configurações e as variáveis de ambiente compatíveis apenas com um SDK ou uma ferramenta individual, consulte o guia específica do SDK ou da ferramenta.

Nome da configuração	Detalhes
aws_access_key_id	AWS teclas de acesso
aws_secret_access_key	AWS teclas de acesso
aws_session_token	AWS teclas 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 Serviços da AWS. SDKs e as ferramentas também podem oferecer suporte a configurações e variáveis de ambiente exclusivas. Para ver as configurações e as variáveis de ambiente compatíveis apenas com um SDK ou uma ferramenta individual, consulte o guia específica do SDK ou da ferramenta.

Nome da configuração	Detalhes
AWS_ACCESS_KEY_ID	AWS teclas de acesso
AWS_ACCOUNT_ID	Endpoints baseados em contas
AWS_ACCOUNT_ID_ENDPOINT_MODE	Endpoints baseados em contas
AWS_AUTH_SCHEME_PREFERENCE	Esquema de autenticação
AWS_CA_BUNDLE	Definições gerais de configuração
AWS_CONFIG_FILE	Localizando e alterando a localização dos arquivos compartilhados, dos <code>credentials</code> arquivos <code>config</code> AWS SDKs e das ferramentas
AWS_CONTAINER_AUTHORIZATION_TOKEN	Provedor de credenciais de contêiner
AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE	Provedor de credenciais de contêiner
AWS_CONTAINER_CREDENTIALS_FULL_URI	Provedor de credenciais de contêiner
AWS_CONTAINER_CREDENTIALS	Provedor de credenciais de contêiner

Nome da configuração	Detalhes
ENTIALS_RELATIVE_URI	
AWS_DEFAULTS_MODE	Padrões de configuração inteligente
AWS_DISABLE_HOST_PREFIX_INJECTION	Injeção de prefixo de host
AWS_DISABLE_REQUEST_COMPRESSION	Compactação de solicitações
AWS_EC2_METADATA_DISABLED	Provedor de credenciais IMDS
AWS_EC2_METADATA_SERVICE_ENDPOINT	Provedor de credenciais IMDS
AWS_EC2_METADATA_SERVICE_ENDPOINT_MODE	Provedor de credenciais IMDS
AWS_EC2_METADATA_V1_DISABLED	Provedor de credenciais IMDS
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	Usando credentials arquivos config e arquivos compartilhados para configurar AWS SDKs e ferramentas globalmente
AWS_REGION	Região da AWS
AWS_REQUEST_CHECKSUM_CALCULATION	Proteções de integridade de dados para o Amazon S3

Nome da configuração	Detalhes
AWS_REQUEST_MIN_COMPRESSION_SIZE_BYTES	Compactação de solicitações
AWS_RESPONSE_CHECKSUM_VALIDATION	Proteções de integridade de dados para o Amazon S3
AWS_RETRY_MODE	Comportamento de repetição
AWS_ROLE_ARN	Assuma a função de provedor de credenciais
AWS_ROLE_SESSION_NAME	Assuma a função de provedor de credenciais
AWS_S3_DISABLE_EXPRESS_SESSION_AUTH	Autenticação de sessão da S3 Express One Zone
AWS_S3_DISABLE_MULTIREGION_ACCESS_POINTS	Pontos de acesso de várias regiões do Amazon S3
AWS_S3_US_E_ARN_REGION	Pontos de acesso Amazon S3
AWS_SDK_UA_APP_ID	ID do aplicativo
AWS_SECRET_ACCESS_KEY	AWS teclas de acesso

Nome da configuração	Detalhes
AWS_SESSION_TOKEN	AWS teclas de acesso
AWS_SHARED_CREDENTIALS_FILE	Localizando e alterando a localização dos arquivos compartilhados, dos <code>credentials</code> arquivos <code>config</code> AWS SDKs e das ferramentas
AWS_SIGV4_A_SIGNING_REGION_SET	Esquema de autenticação
AWS_STS_REGIONAL_ENDPOINTS	AWS STS Endpoints regionais
AWS_USE_DUALSTACK_ENDPOINT	Endpoints de pilha dupla e FIPS
AWS_USE_FIPS_ENDPOINT	Endpoints de pilha dupla e FIPS
AWS_WEB_IDENTITY_TOKEN_FILE	Assuma a função de provedor de credenciais

Lista de propriedades de sistema de JVM

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

Nome da configuração	Detalhes
<code>aws.accessKeyId</code>	AWS teclas de acesso
<code>aws.accountId</code>	Endpoints baseados em contas
<code>aws.accountIdEndpointMode</code>	Endpoints baseados em contas
<code>aws.authSchemePreference</code>	Esquema de autenticação
<code>aws.configFile</code>	Localizando e alterando a localização dos arquivos compartilhados, dos <code>credentials</code> arquivos <code>config</code> AWS SDKs e das ferramentas
<code>aws.defaultsMode</code>	Padrões de configuração inteligente
<code>aws.disableEc2MetadataV1</code>	Provedor de credenciais IMDS
<code>aws.disableHostPrefixInjection</code>	Injeção de prefixo de host
<code>aws.disableRequestCompression</code>	Compactação de solicitações
<code>aws.disableS3ExpressAuth</code>	Autenticação de sessão da S3 Express One Zone

Nome da configuração	Detalhes
<code>aws.ec2MetadataServiceEndpoint</code>	Provedor de credenciais IMDS
<code>aws.ec2MetadataEndpointMode</code>	Provedor de credenciais IMDS
<code>aws.endpointDiscoveryEnabled</code>	Descoberta de endpoint
<code>aws.endpointUrl</code>	Endpoints específicos de serviço
<code>aws.endpointUrl<ServiceName></code>	Endpoints específicos de serviço
<code>aws.ignoreConfiguredEndpointUrls</code>	Endpoints específicos de serviço
<code>aws.maxAttempts</code>	Comportamento de repetição
<code>aws.profile</code>	Usando <code>credentials</code> arquivos <code>config</code> e arquivos compartilhados para configurar AWS SDKs e ferramentas globalmente
<code>aws.region</code>	Região da AWS
<code>aws.requestChecksumCalculation</code>	Proteções de integridade de dados para o Amazon S3

Nome da configuração	Detalhes	
<code>aws.requestMinCompressionSizeBytes</code>	Compactação de solicitações	
<code>aws.responseChecksumValidation</code>	Proteções de integridade de dados para o Amazon S3	
<code>aws.retryMode</code>	Comportamento de repetição	
<code>aws.roleArn</code>	Assuma a função de provedor de credenciais	
<code>aws.roleSessionName</code>	Assuma a função de provedor de credenciais	
<code>aws.s3DisableMultiRegionAccessPoints</code>	Pontos de acesso de várias regiões do Amazon S3	
<code>aws.s3UseArnRegion</code>	Pontos de acesso Amazon S3	
<code>aws.secretAccessKey</code>	AWS teclas de acesso	
<code>aws.sessionToken</code>	AWS teclas de acesso	
<code>aws.shareCredentialsFile</code>	Localizando e alterando a localização dos arquivos compartilhados, dos <code>credentials</code> arquivos <code>config</code> AWS SDKs e das ferramentas	

Nome da configuração	Detalhes
<code>aws.useDualstackEndpoint</code>	Endpoints de pilha dupla e FIPS
<code>aws.useFipsEndpoint</code>	Endpoints de pilha dupla e FIPS
<code>aws.webIdentityTokenFile</code>	Assuma a função de provedor de credenciais
<code>sdk.ua.appId</code>	ID do aplicativo

AWS SDKs e 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 vários SDKs. Todas as configurações podem ser substituídas no código. Para obter detalhes, consulte a API específica de seu SDK.

Important

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

Tópicos

- [Entender a cadeia de provedores de credenciais](#)
- [Cadeias de provedores de credenciais específicas do SDK e da ferramenta](#)
- [AWS teclas de acesso](#)
- [Provedor de credenciais de login](#)
- [Assuma o perfil de provedor de credenciais](#)

- [Provedor de credenciais de contêiner](#)
- [Provedor de credencial do IAM Identity Center](#)
- [Provedor de credenciais IMDS](#)
- [Provedor de credenciais de processo](#)

Entender a cadeia de provedores 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 denominada cadeia de provedores 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 à aplicação a capacidade de atualizar suas credenciais, independentemente do provedor que você está usando na cadeia. Nenhum código adicional é necessário para que o SDK faça isso.

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

Provedor de credencial	Description
AWS teclas de acesso	AWS chaves de acesso para um usuário do IAM (como <code>AWS_ACCESS_KEY_ID</code> e <code>AWS_SECRET_ACCESS_KEY</code>).
Federar com identidade da Web ou OpenID Connect : assumir a função de provedor de credenciais	Fazer login usando um provedor de identidades (IdP) externo conhecido, como Login with Amazon, Facebook, Google ou qualquer outro IdP compatível com OpenID Connect (OIDC). Assuma as permissões de uma função do IAM usando um JSON Web Token (JWT) de AWS Security Token Service (AWS STS).
Provedor de credenciais de login	Obtenha credenciais para uma sessão de console nova ou existente na qual você está conectado.

Provedor de credencial	Description
Provedor de credencial do IAM Identity Center	Obtenha credenciais de Centro de Identidade do AWS IAM.
Assuma o perfil de provedor de credenciais	Tenha acesso a outros recursos assumindo as permissões de um perfil do IAM. (Recupere e use credenciais temporárias para uma função).
Provedor de credenciais de contêiner	Credenciais do Amazon Elastic Container Service (Amazon ECS) e do Amazon Elastic Kubernetes Service (Amazon EKS). O provedor de credenciais de contêiner busca credenciais para a aplicação containerizada do cliente.
Provedor de credenciais de processo	Provedores de credenciais personalizados. Obtenha suas credenciais de uma fonte ou processo externo, incluindo o IAM Roles Anywhere.
Provedor de credenciais IMDS	Credenciais do perfil de instância do Amazon Elastic Compute Cloud (Amazon EC2). Associe um perfil do IAM a cada uma das suas instâncias do EC2. As credenciais temporárias para essa função estão disponíveis para o código em execução na instância. As credenciais são fornecidas por meio do serviço de metadados do Amazon EC2.

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 [Usando credentials arquivos compartilhados config para configurar AWS SDKs e ferramentas globalmente](#). Para obter mais informações, consulte [Precedência de configurações](#).

Cadeias de provedores de credenciais específicas do SDK e da ferramenta

Para acessar diretamente os detalhes específicos da cadeia de fornecedores de credenciais do SDK ou da ferramenta, escolha o SDK ou a ferramenta entre as seguintes opções:

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

AWS teclas de acesso

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 [Centro de Identidade do AWS IAM](#).

AWS as chaves de acesso de um usuário do IAM podem ser usadas como suas AWS credenciais. O AWS SDK usa automaticamente essas AWS credenciais para assinar solicitações de API 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. O uso de credenciais de longo prazo não é recomendado.

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 específicas do SDK do arquivo compartilhado. `config`

Para saber mais sobre AWS credenciais e o uso de chaves de acesso, consulte [Credenciais AWS de segurança](#) e [Gerenciamento de chaves de acesso para usuários do IAM no Guia](#) do usuário do IAM.

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 do sistema JVM: 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 do sistema JVM: 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 do sistema JVM: 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 obter recomendações de segurança, consulte [Melhores práticas de segurança no IAM](#).

Para obter instruções sobre como obter esses valores, consulte [Usando credenciais de curto prazo para autenticação e ferramentas AWS SDKs](#).

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...zrkuWJ0gQs8IZZaIv2BXIa2R40lgk
```

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...zrkuWJ0gQs8IZZaIv2BXIa2R40lgk
```

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...zrkuWJ0gQs8IZZaIv2BXIa2R40lgk
```

Support by AWS SDKs and tools

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 sistema JVM são suportadas pelo AWS SDK para Java e pelo AWS SDK para Kotlin único.

SDK	Ci	Notas ou mais informações
AWS CLI v2	Sim	
SDK para C++	Sim	arquivo compartilhado config não suportado.
SDK para Go V2 (1.x)	Sim	
SDK para Go 1.x (V1)	Sim	Para usar as configurações do arquivo config compartilhado, você deve ativar o carregamento do arquivo de configuração; consulte Sessões .
SDK para Java 2.x	Sim	

SDK	Compatibilidade	Notas ou mais informações
SDK para Java 1.x	Sim	
SDK para 3.x JavaScript	Sim	
SDK para 2.x JavaScript	Sim	
SDK para Kotlin	Sim	
SDK para .NET 4.x	Sim	
SDK para .NET 3.x	Sim	
SDK para PHP 3.x	Sim	
SDK para Python (Boto3)	Sim	
SDK para Ruby 3.x	Sim	
SDK para Rust	Sim	
SDK para Swift	Sim	
Ferramentas para PowerShell V5	Sim	
Ferramentas para PowerShell V4	Sim	As variáveis de ambiente não são compatíveis.

Provedor de credenciais de login

Você pode [usar suas credenciais de login existentes do AWS Management Console para adquirir credenciais](#) de curto prazo que podem ser usadas para acesso programático. Depois de concluir o fluxo de autenticação baseado em navegador, AWS gera credenciais temporárias que funcionam em ferramentas de desenvolvimento locais, como AWS CLI, Tools for e. AWS PowerShell AWS SDKs

Para gerar essas credenciais, execute o `aws login` comando na AWS CLI ou o cmdlet `AWS Invoke-AWSLogin` Tools for. PowerShell As credenciais de curto prazo resultantes serão

armazenadas em cache localmente, onde poderão ser reutilizadas pelo. AWS SDKs As credenciais de curto prazo expiram em 15 minutos, mas a CLI as SDKs atualizará automaticamente conforme necessário em até 12 horas. Quando o token de atualização expirar, você será solicitado a fazer login novamente por meio da CLI ou. PowerShell

O comando de login atualizará o perfil especificado com a `login_session` configuração, que armazena a identidade da sessão do console de gerenciamento que você selecionou durante o fluxo de trabalho de login.

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

Por padrão, as credenciais de curto prazo e o token de atualização são armazenados em um arquivo JSON no `~/.aws/login/cache` diretório no Linux e no macOS ou no Windows. `%USERPROFILE%\aws\login\cache` O nome do arquivo é baseado no nome da sessão de login. Você pode substituir o diretório definindo a variável de `AWS_LOGIN_CACHE_DIRECTORY` ambiente.

Configurações do provedor de login

Configure essa funcionalidade usando o seguinte:

AWS_LOGIN_CACHE_DIRECTORY: variável de ambiente

Diretório alternativo em que a CLI SDKs armazenará as credenciais em cache que são mapeadas para um perfil de sessão de login.

Valor padrão: `~/.aws/login/cache` no Linux e no macOS ou `%USERPROFILE%\aws\login\cache` no Windows.

Support by AWS SDKs and tools

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 sistema JVM são suportadas pelo AWS SDK para Java e pelo AWS SDK para Kotlin único.

SDK	Cl	Notas ou mais informações
AWS CLI v2	Sim	
SDK para C++	Sim	
SDK para Go V2 (1.x)	Não	
SDK para Go 1.x (V1)	Sim	
SDK para Java 2.x	Sim	
SDK para Java 1.x	Não	
SDK para 3.x JavaScript	Sim	
SDK para 2.x JavaScript	Não	
SDK para Kotlin	Sim	
SDK para .NET 4.x	Sim	
SDK para .NET 3.x	Sim	
SDK para PHP 3.x	Sim	
SDK para Python (Boto3)	Sim	Requer CRT
SDK para Ruby 3.x	Sim	
SDK para Rust	Sim	
Ferramentas para PowerShell V5	Sim	
Ferramentas para PowerShell V4	Não	

Assuma o perfil de provedor de credenciais

Note

Para obter ajuda na compreensão do layout das páginas de configurações ou na interpretação da tabela Support by AWS SDKs and tools a seguir, consulte [Entender as páginas de configurações deste guia](#).

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 seu SDK ou ferramenta para assumir um perfil, você deve primeiro criar ou identificar um perfil específico a ser assumido. Os perfis do IAM são identificados exclusivamente por um perfil do nome do recurso da Amazon ([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 uma federação OIDC ou SAML.

Depois que perfil do IAM for identificado, se você tiver a confiança desse perfil, poderá configurar seu SDK ou ferramenta para usar as permissões concedidas pelo perfil. Para fazer isso, execute as configurações a seguir.

Para obter orientação sobre como começar a usar essas configurações, consulte este guia [Assumindo uma função com AWS credenciais para autenticação AWS SDKs e ferramentas](#).

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 instâncias ou contêineres do Amazon EC2 ou Amazon Elastic Container Service para especificar onde o SDK ou ferramenta pode encontrar credenciais que tem permissão para assumir o perfil que você especificar com o parâmetro `role_arn`.

Valor padrão: nenhum

Valores válidos:

- **Ambiente:** especifica que o SDK ou a ferramenta deve recuperar credenciais de origem de variáveis de ambiente [AWS_ACCESS_KEY_ID](#) e [AWS_SECRET_ACCESS_KEY](#).
- **Ec2 InstanceMetadata** — Especifica que o SDK ou a ferramenta deve usar a [função do IAM anexada ao perfil da instância do EC2 para obter as](#) credenciais de origem.
- **EcsContainer**— Especifica que o SDK ou a ferramenta deve usar a [função do IAM anexada ao contêiner do Amazon ECS ou a função do IAM anexada ao contêiner do Amazon EKS para obter as credenciais](#) de origem.

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

Exemplo de configuração em um arquivo `config` para indicar que as credenciais devem ser provenientes do 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 um perfil](#) no Guia do usuário do IAM.

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 usuário do IAM.

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 um perfil.

Obrigatório ao assumir um perfil em que a política de confiança para o perfil inclui uma condição que exige autenticação de MFA. Para obter mais informações sobre MFA, consulte [Código da autenticação multifator no IAM da AWS](#) no Guia do usuário do IAM.

Valor padrão: nenhum.

Valores válidos: o valor pode ser um número de série de um dispositivo de hardware (como GAHT12345678) ou um nome do recurso da Amazon (ARN) de um dispositivo MFA 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 dispositivo virtual de MFA, denominado 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 do sistema JVM: somente Java/Kotlin

Especifica o nome do recurso da Amazon (ARN) de um perfil do IAM que você deseja usar para realizar operações solicitadas usando esse perfil.

Valor padrão: nenhum.

Valores válidos: o valor deve ser o ARN de um perfil do IAM, 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 do Amazon EC2 ou a uma instância de contêiner do Amazon ECS.
- `web_identity_token_file`: usar provedores de identidades públicas ou qualquer provedor de identidades compatível com OpenID Connect (OIDC) para usuários que foram autenticados em um aplicativo móvel ou aplicativo web.

role_session_name- configuração de AWS **config** arquivo compartilhado,

AWS_ROLE_SESSION_NAME: variável de ambiente, **aws.roleSessionName**- Propriedade do sistema JVM: 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 [elemento CloudTrail userIdentity](#) no Guia do AWS CloudTrail 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 a AWS API AWS CLI ou 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, consulte [Arquivos 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 o SDK encontra um perfil com credenciais. O encadeamento de funções limita sua sessão de função AWS CLI ou de AWS API a no máximo uma hora e não pode ser aumentado. Para obter mais informações, consulte [Termos e conceitos de funções](#) no Guia do usuário do IAM.

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 perfil A instrui o SDK ou a ferramenta a pesquisar automaticamente as credenciais do perfil B vinculado. Nesse caso, o perfil B usa a ferramenta assistente de credenciais fornecida por [Usando o IAM Roles Anywhere para autenticação AWS SDKs e ferramentas](#) para obter credenciais para o SDK da AWS. Essas credenciais temporárias são então usadas pelo código para acessar recursos da AWS. O papel especificado deve ter políticas de permissões do IAM anexadas que permitam a execução do código solicitado, como o comando ou o método da API. AWS service (Serviço da AWS) Cada ação realizada pelo perfil A tem o nome da sessão da função incluído nos CloudTrail registros.

Como um segundo exemplo de encadeamento de perfis, a configuração a seguir poderá ser usada se você tiver uma aplicação em uma instância do Amazon Elastic Compute Cloud e quiser que ela assuma outro perfil.

```
[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 de instância do Amazon EC2 para assumir o perfil especificado 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 do sistema JVM: 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. O SDK ou a ferramenta do desenvolvedor carrega o conteúdo deste arquivo e o transmite como argumento `WebIdentityToken` quando chama a operação `AssumeRoleWithWebIdentity` 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.

Support by AWS SDKs and tools

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 sistema JVM são suportadas pelo AWS SDK para Java e pelo AWS SDK para Kotlin único.

SDK	Compatibilidade	Notas ou mais informações
AWS CLI v2	Sim	
SDK para C++	Parcial	<code>credential_source</code> não suportado. <code>duration_seconds</code> não suportado. <code>mfa_serial</code> não suportado.
SDK para Go V2 (1.x)	Sim	
SDK para Go 1.x (V1)	Sim	Para usar as configurações do arquivo <code>config</code> compartilhado, você deve ativar o carregamento do arquivo de configuração; consulte Sessões .
SDK para Java 2.x	Parcial	<code>mfa_serial</code> não compatível. <code>duration_seconds</code> não compatível.
SDK para Java 1.x	Parcial	<code>credential_source</code> não compatível. <code>mfa_serial</code> não compatível. Propriedades do sistema JVM não compatíveis.
SDK para 3.x JavaScript	Sim	
SDK para 2.x JavaScript	Parcial	<code>credential_source</code> incompatível.
SDK para Kotlin	Sim	
SDK para .NET 4.x	Sim	
SDK para .NET 3.x	Sim	
SDK para PHP 3.x	Sim	
SDK para Python (Boto3)	Sim	
SDK para Ruby 3.x	Sim	
SDK para Rust	Sim	
SDK para Swift	Sim	

SDK	C	Notas ou mais informações
Ferramentas para PowerShell I V5	Sim	
Ferramentas para PowerShell I V4	Sim	

Provedor de credenciais de contêiner

Note

Para obter ajuda na compreensão do layout das páginas de configurações ou na interpretação da tabela Support by AWS SDKs and tools a seguir, consulte [Entender as páginas de configurações deste guia](#).

O provedor de credenciais de contêiner busca credenciais para a aplicação containerizada do cliente. Esse provedor de credenciais é útil para clientes do Amazon Elastic Container Service (Amazon ECS) e do Amazon Elastic Kubernetes Service (Amazon EKS). SDKs tente carregar credenciais do endpoint HTTP especificado por meio de uma solicitação GET.

Se você usa o Amazon ECS, recomendamos que você use um perfil do IAM de tarefa para melhorar o isolamento, a autorização e a auditabilidade das credenciais. Quando configurado, o Amazon ECS define a variável de ambiente `AWS_CONTAINER_CREDENTIALS_RELATIVE_URI` que as ferramentas SDKs e ferramentas usam para obter credenciais. Para configurar o Amazon ECS para essa funcionalidade, consulte a [Função do IAM de tarefa](#) no Amazon Elastic Container Service Developer Guide.

Se você usa o Amazon EKS, recomendamos usar o Amazon EKS Pod Identity para melhorar o isolamento de credenciais, privilégios mínimos, auditabilidade, operação independente, reutilização e escalabilidade. Tanto seu Pod quanto um perfil do IAM 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 usuário do Amazon EKS. Quando configurado, o Amazon EKS define as variáveis de ambiente `AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE`

`AWS_CONTAINER_CREDENTIALS_FULL_URI` e as variáveis de ambiente que as ferramentas SDKs e as ferramentas usam para obter credenciais. Para obter informações de configuração, consulte [Configurar o Amazon EKS Pod Identity Agent](#) no Guia do usuário do Amazon EKS ou o [Amazon EKS Pod Identity simplifica as permissões do IAM para aplicativos em clusters do Amazon EKS](#) no site do AWS blog.

Configure essa funcionalidade usando o seguinte:

`AWS_CONTAINER_CREDENTIALS_FULL_URI`: variável de ambiente

Contém o endpoint de URL HTTP relativo para o SDK usar ao fazer uma solicitação de credenciais. Isso inclui o esquema e o host.

Valor padrão: nenhum.

Valores válidos: URL válido.

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

Contém o endpoint de URL HTTP completo para o SDK usar ao fazer uma solicitação de credenciais. O valor é anexado ao nome de host padrão do Amazon ECS de `169.254.170.2`.

Valor padrão: Nenhum.

Valores válidos: URL relativo válido.

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 solicitação HTTP 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
```

Support by AWS SDKs and tools

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 sistema JVM são suportadas pelo AWS SDK para Java e pelo AWS SDK para Kotlin único.

SDK	CI	Notas ou mais informações
AWS CLI v2	Sim	
SDK para C++	Sim	
SDK para Go V2 (1.x)	Sim	
SDK para Go 1.x (V1)	Sim	
SDK para Java 2.x	Sim	Quando o Lambda SnapStart é ativado AWS_CONTAINER_CREDENTIALS_FULL_URI e AWS_CONTAINER_AUTHORIZATION_TOKEN é usado automaticamente para autenticação.
SDK para Java 1.x	Sim	Quando o Lambda SnapStart é ativado AWS_CONTAINER_CREDENTIALS_FULL_URI e AWS_CONTAINER_AUTHORIZATION_TOKEN é usado automaticamente para autenticação.
SDK para 3.x JavaScript	Sim	
SDK para 2.x JavaScript	Sim	
SDK para Kotlin	Sim	
SDK para .NET 4.x	Sim	Quando o Lambda SnapStart é ativado AWS_CONTAINER_CREDENTIALS_FULL_URI e AWS_CONTAINER_AUTHORIZATION_TOKEN é usado automaticamente para autenticação.
SDK para .NET 3.x	Sim	Quando o Lambda SnapStart é ativado AWS_CONTAINER_CREDENTIALS_FULL_URI e AWS_CONTAINER_AUTHORIZATION_TOKEN é usado automaticamente para autenticação.
SDK para PHP 3.x	Sim	

SDK	C	Notas ou mais informações
SDK para Python (Boto3)	Sim	Quando o Lambda SnapStart é ativado AWS_CONTAINER_CREDENTIALS_FULL_URI e AWS_CONTAINER_AUTHORIZATION_TOKEN é usado automaticamente para autenticação.
SDK para Ruby 3.x	Sim	
SDK para Rust	Sim	
SDK para Swift	Sim	
Ferramentas para PowerShell V5	Sim	
Ferramentas para PowerShell V4	Sim	

Provedor de credencial do IAM Identity Center

Note

Para obter ajuda na compreensão do layout das páginas de configurações ou na interpretação da tabela Support by AWS SDKs and tools a seguir, consulte [Entender as páginas de configurações deste guia](#).

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

Note

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

Depois de habilitar o IAM Identity Center, você define um perfil para suas configurações no AWS `config` arquivo compartilhado. Este 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 o perfil do IAM associado a esse usuário. Para saber como o SDK obtém credenciais temporárias da configuração e as usa para AWS service (Serviço da AWS) solicitações, consulte [Como a autenticação do IAM Identity Center é resolvida AWS SDKs e as ferramentas](#)

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

- (Recomendado) Configuração do provedor de token SSO: durações de sessão estendidas. Inclui compatibilidade com 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:

- [Centro de Identidade do AWS IAM Guia do usuário](#)
- [Centro de Identidade do AWS IAM Referência da API do portal](#)

Para saber mais sobre como as ferramentas SDKs e usam e atualizam as credenciais usando essa configuração, consulte [Como a autenticação do IAM Identity Center é resolvida AWS SDKs e as ferramentas](#)

Pré-requisitos

É necessário primeiro habilitar o IAM Identity Center. Para obter detalhes sobre como habilitar a autenticação do IAM Identity Center, consulte [Enabling Centro de Identidade do AWS IAM](#) no Centro de Identidade do AWS IAM User Guide.

Note

Como alternativa, para obter os pré-requisitos completos e a configuração do arquivo `config` compartilhado necessário que são detalhados nesta página, consulte as instruções guiadas para a configuração do [Usando o IAM Identity Center para autenticar o AWS SDK e as ferramentas](#).

Configuração do provedor de token do SSO

Quando você usa a configuração do provedor de token SSO, seu AWS SDK ou ferramenta 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 Centro de Identidade do AWS IAM do usuário.

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

O exemplo de arquivo compartilhado `config` a seguir configura o SDK ou a ferramenta usando um perfil `dev` 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 sessão do `sso-session` e a associa a um perfil. Normalmente, `sso_account_id` e `sso_role_name` deve ser definido na `profile` seção para que o SDK possa solicitar AWS credenciais. `sso_region`, `sso_start_url`, e `sso_registration_scopes` deve ser definido dentro da `sso-session` seção.

No entanto, `sso_account_id` e `sso_role_name` não são necessários para todos os cenários de configuração do token do SSO. 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 autenticação HTTP que usa tokens de segurança chamados tokens de 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 no OAuth 2.0 para limitar o acesso de um aplicativo à conta de um usuário. O exemplo anterior define `sso_registration_scopes` para fornecer o acesso necessário para listar contas e perfis.

O exemplo a seguir mostra como é possível reutilizar a mesma configuração `sso-session` 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 [Configuração do provedor de token do SSO](#).

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`.

Configurações do provedor de credenciais do IAM Identity Center

Configure essa funcionalidade usando o seguinte:

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

A URL que aponta para a URL do emissor do IAM Identity Center ou para a URL do portal de acesso da sua organização. Para obter informações consulte [Sign in to the AWS access portal](#) no Centro de Identidade do AWS IAM User Guide.

Para encontrar esse valor, abra o [console do IAM Identity Center](#), visualize o painel e encontre a URL do portal de acesso da AWS .

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

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 página de Contas da AWS. Você também pode ver a lista de contas disponíveis usando o método [ListAccounts](#)API na Referência da API do Centro de Identidade do AWS IAM 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 um perfil do IAM que define as permissões resultantes do usuário. A função deve existir no Conta da AWS especificado por `sso_account_id`. Use o nome do perfil, não o nome do recurso da Amazon (ARN) do perfil.

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

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 AWS página. 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 método [ListAccountRoles](#)API na Referência da API do Centro de Identidade do AWS IAM 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 [Access scopes](#) no Centro de Identidade do AWS IAM User Guide.

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

Support by AWS SDKs and tools

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 sistema JVM são suportadas pelo AWS SDK para Java e pelo AWS SDK para Kotlin único.

SDK	Compatível	Notas ou mais informações
AWS CLI v2	Sim	
SDK para C++	Sim	
SDK para Go V2 (1.x)	Sim	
SDK para Go 1.x (V1)	Sim	Para usar as configurações do arquivo config compartilhado, você deve ativar o carregamento do arquivo de configuração; consulte Sessões .
SDK para Java 2.x	Sim	Valores de configuração também compatíveis no arquivo <code>credentials</code> .
SDK para Java 1.x	Não	
SDK para 3.x JavaScript	Sim	
SDK para 2.x JavaScript	Sim	
SDK para Kotlin	Sim	
SDK para .NET 4.x	Sim	
SDK para .NET 3.x	Sim	
SDK para PHP 3.x	Sim	
SDK para Python (Boto3)	Sim	
SDK para Ruby 3.x	Sim	
SDK para Rust	Parcial	Somente configuração herdada não atualizável.

SDK	CI	Notas ou mais informações
SDK para Swift	Sim	
Ferramentas para PowerShell I V5	Sim	
Ferramentas para PowerShell I V4	Sim	

Provedor de credenciais IMDS

Note

Para obter ajuda na compreensão do layout das páginas de configurações ou na interpretação da tabela Support by AWS SDKs and tools a seguir, consulte [Entender as páginas de configurações deste guia](#).

O serviço de metadados de instância (IMDS) fornece dados sobre sua instância que é possível usar para configurar ou gerenciar a instância em execução. Para obter mais informações, consulte [Trabalhar com metadados de instância](#) no Guia do usuário do Amazon EC2. O Amazon EC2 fornece um endpoint local disponível para instâncias que podem 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 do 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 erro HTTP 403, 404, 405), IMDSv1 é usado como alternativa.

Configure essa funcionalidade usando o seguinte:

AWS_EC2_METADATA_DISABLED: variável de ambiente

Se o serviço de metadados de instância (IMDS) do Amazon EC2 (IMDS) deve ou não ser usado para obter credenciais.


Valor padrão: `false`.

Valores válidos:

- **true** – Não use o IMDS para obter credenciais.
- **false** – Use o 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 do sistema JVM: 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 [Support by AWS SDKs and tools](#).

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 do sistema JVM: somente Java/Kotlin

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

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: URL válido.

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

O modo de endpoint do IMDS.

Valor padrão: IPv4.

Valores válidos: IPv4, IPv6.

Note

O provedor de credenciais do IMDS faz parte do [Entender a cadeia de provedores de credenciais](#). No entanto, o provedor de credenciais do IMDS 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 provedores de credenciais para descobrir automaticamente qual provedor retorna credenciais válidas, especifique o uso do provedor de credenciais IMDS no código. Você pode especificar fontes de credenciais diretamente ao criar clientes de serviço.

Segurança para credenciais do IMDS

Por padrão, quando o AWS SDK não está configurado com credenciais válidas, o SDK 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 de rede desnecessárias e aumenta a segurança em redes não confiáveis nas quais o serviço de metadados de instância do Amazon EC2 pode ser representado.

Note

AWS Clientes SDK configurados com credenciais válidas nunca usarão o IMDS para recuperar credenciais, independentemente de qualquer uma dessas configurações.

Desabilitando o uso das credenciais do 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.

Support by AWS SDKs and tools

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 sistema JVM são suportadas pelo AWS SDK para Java e pelo AWS SDK para Kotlin único.

SDK	Compartilhado	Notas ou mais informações
AWS CLI v2	Sim	
SDK para C++	Sim	
SDK para Go V2 (1.x)	Sim	
SDK para Go 1.x (V1)	Sim	Para usar as configurações do arquivo config compartilhado, você deve ativar o carregamento do arquivo de configuração; consulte Sessões .
SDK para Java 2.x	Sim	
SDK para Java 1.x	Parcial	Propriedades de sistema JVM: use <code>com.amazonaws.sdk.disableEc2MetadataV1</code> em vez de <code>aws.disableEc2MetadataV1</code> ; <code>aws.ec2MetadataServiceEndpoint</code> e <code>aws.ec2MetadataServiceEndpointMode</code> não compatíveis.
SDK para 3.x JavaScript	Sim	
SDK para 2.x JavaScript	Sim	
SDK para Kotlin	Sim	Não usa IMDSv1 fallback.
SDK para .NET 4.x	Sim	
SDK para .NET 3.x	Sim	
SDK para PHP 3.x	Sim	
SDK para Python (Boto3)	Sim	
SDK para Ruby 3.x	Sim	
SDK para Rust	Sim	Não usa IMDSv1 fallback.
SDK para Swift	Sim	

SDK	C	Notas ou mais informações
Ferramentas para PowerShell V5	Sim	Você pode desativar o IMDSv1 fallback explicitamente no código usando. <code>[Amazon.Util.EC2InstanceMetadata]::EC2MetadataV1Disabled = \$true</code>
Ferramentas para PowerShell V4	Sim	Você pode desativar o IMDSv1 fallback explicitamente no código usando. <code>[Amazon.Util.EC2InstanceMetadata]::EC2MetadataV1Disabled = \$true</code>

Provedor de credenciais de processo

Note

Para obter ajuda na compreensão do layout das páginas de configurações ou na interpretação da tabela Support by AWS SDKs and tools a seguir, consulte [Entender as páginas de configurações deste guia](#).

SDKs fornecem 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 recuperação de credenciais de um repositório de credenciais on-premises ou integração com o provedor de identificação on-premises.

Por exemplo, o IAM Roles Anywhere usa o `credential_process` para obter credenciais temporárias para sua aplicação. Para configurar `credential_process` para esse uso, consulte [Usando o IAM Roles Anywhere para autenticação AWS SDKs e ferramentas](#).

Note

A seguir, descrevemos um método de obtenção de credenciais de um processo externo que pode ser usado se você estiver executando software fora da AWS. Se você estiver criando um recurso AWS computacional, use outros provedores de credenciais. Se usar esta opção, certifique-se de que o arquivo config 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 secretaStdErr, pois ela

AWS CLI pode capturar SDKs e registrar essas informações, potencialmente expondo-as a usuários não autorizados.

Configure essa funcionalidade usando o seguinte:

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

Especifica um comando externo que o SDK ou uma ferramenta executa em seu nome para gerar ou recuperar credenciais de autenticação a serem usadas. A configuração especifica o nome de um program/command que o SDK invocará. Quando o SDK invoca o processo, ele espera que o processo grave dados JSON em stdout. O provedor personalizado deve retornar informações em um formato específico. Essas informações contêm as credenciais que o SDK ou a ferramenta podem usar para autenticar você.

Note

O provedor de credenciais do processo faz parte do [Entender a cadeia de provedores 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 o SDK ou a ferramenta de desenvolvimento 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 dependendo 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, deverá gerar a saída JSON em STDOUT que corresponde 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 chave `Expiration` não estiver presente na saída da ferramenta, o SDK vai supor que as credenciais são 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 como faz com credenciais `assume-role`. 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.

Support by AWS SDKs and tools

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 sistema JVM são suportadas pelo AWS SDK para Java e pelo AWS SDK para Kotlin único.

SDK	Compartilhe Notas ou mais informações
AWS CLI v2	Sim
SDK para C++	Sim
SDK para Go V2 (1.x)	Sim

SDK	CI	Notas ou mais informações
SDK para Go 1.x (V1)	Sim	Para usar as configurações do arquivo <code>config</code> compartilhado, você deve ativar o carregamento do arquivo de configuração; consulte Sessões .
SDK para Java 2.x	Sim	
SDK para Java 1.x	Sim	
SDK para 3.x JavaScript	Sim	
SDK para 2.x JavaScript	Sim	
SDK para Kotlin	Sim	
SDK para .NET 4.x	Sim	
SDK para .NET 3.x	Sim	
SDK para PHP 3.x	Sim	
SDK para Python (Boto3)	Sim	
SDK para Ruby 3.x	Sim	
SDK para Rust	Sim	
SDK para Swift	Sim	
Ferramentas para PowerShell V5	Sim	
Ferramentas para PowerShell V4	Sim	

AWS SDKs e 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 vários SDKs. Todas as configurações podem ser substituídas no código. Consulte sua API específica do SDK 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 conta](#)
- [ID da aplicação](#)
- [Metadados da instância do Amazon EC2](#)
- [Pontos de acesso Amazon S3](#)
- [Pontos de acesso de várias regiões do Amazon S3](#)
- [Autenticação de sessão da S3 Express One Zone](#)
- [Esquema de autenticação](#)
- [Região da AWS](#)
- [AWS STS Endpoints regionais](#)
- [Proteções de integridade de dados para o Amazon S3](#)
- [Endpoints de pilha dupla e FIPS](#)
- [Descoberta de endpoint](#)
- [Definições gerais da configuração](#)
- [Injeção de prefixo de host](#)
- [Cliente de IMDS](#)
- [Comportamento de repetição](#)
- [Compactação de solicitações](#)
- [Endpoints específicos de serviço](#)
- [Padrões de configuração inteligente](#)

Endpoints baseados em conta

Note

Para obter ajuda na compreensão do layout das páginas de configurações ou na interpretação da tabela Support by AWS SDKs and tools a seguir, consulte [Entender as páginas de configurações deste guia](#).

Os endpoints baseados em conta ajudam a garantir alta performance e escalabilidade usando o seu ID da Conta da AWS para rotear solicitações para serviços compatíveis com esse atributo. Quando você usa um SDK da AWS e um serviço compatível com endpoints baseados em conta, o cliente do SDK cria e usa um endpoint baseado em conta em vez de um endpoint regional. Se o ID da conta não estiver visível para o cliente do SDK, ele usará o endpoint regional. Os endpoints baseados em conta assumem a forma de `https://<account-id>.ddb.<region>.amazonaws.com`, onde `<account-id>` e onde `<region>` está seu ID e. Conta da AWS Região da AWS

Configure essa funcionalidade usando o seguinte:

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

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

O roteamento de endpoints baseado em conta fornece melhor performance em 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 do sistema JVM: somente Java/Kotlin

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

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, o SDK lançará um erro.

Support by AWS SDKs and tools

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 sistema JVM são suportadas pelo AWS SDK para Java e pelo AWS SDK para Kotlin único.

SDK	Com l	Lançado na versão do SDK	Notas ou mais informações
AWS CLI v2	Sim	2.25.0	
AWS CLI v1	Sim	1.38,0	
SDK para C++	Não		
SDK para Go V2 (1.x)	Sim	v1.35.0	
SDK para Go 1.x (V1)	Não		
SDK para Java 2.x	Sim	v2.28.4	
SDK para Java 1.x	Sim	v1.12.771	
SDK para 3.x JavaScript	Sim	v3.656.0	
SDK para 2.x JavaScript	Não		
SDK para Kotlin	Sim	v1.3.37	
SDK para .NET 4.x	Sim	4.0.0	

SDK	Com I	Lançado na versão do SDK	Notas ou mais informações
SDK para .NET 3.x	Não		
SDK para PHP 3.x	Sim	v3.318.0	
SDK para Python (Boto3)	Sim	1.37.0	
SDK para Ruby 3.x	Sim	v1.123.0	
SDK para Rust	Sim	lançament o-2025-04 -24	
SDK para Swift	Sim	1.2.0	
Ferramentas para PowerShell V5	Não		
Ferramentas para PowerShell V4	Não		

ID da aplicação

Note

Para obter ajuda na compreensão do layout das páginas de configurações ou na interpretação da tabela Support by AWS SDKs and Tools a seguir, consulte. [Entender as páginas de configurações deste guia](#)

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 Os SDKs e os

serviços não usam nem interpretam esse valor a não ser para trazê-lo de volta às comunicações com os clientes. 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- configuração de AWS **config** arquivo compartilhado, **AWS_SDK_UA_APP_ID**: variável de ambiente, **sdk.ua.appId**- Propriedade do sistema JVM: somente Java/Kotlin

Essa configuração é uma string exclusiva que você atribui ao seu aplicativo para identificar para quais aplicativos em um determinado aplicativo Conta da AWS fazem chamadas 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
```

Linux/macOS exemplo de configuração de variáveis de ambiente via 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 o shell que está sendo usado, escape o valor conforme apropriado.

Support by AWS SDKs e ferramentas

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

SDK	Compatível	Notas ou mais informações
AWS CLI v2	Sim	
SDK para C++	Sim	arquivo compartilhado config não suportado.
SDK para Go V2 (1.x)	Sim	
SDK para Go 1.x (V1)	Não	
SDK para Java 2.x	Sim	
SDK para Java 1.x	Não	
SDK para 3.x JavaScript	Sim	
SDK para 2.x JavaScript	Não	
SDK para Kotlin	Sim	A propriedade do sistema de JVM é <code>aws.userAgentAppId</code> .
SDK para .NET 4.x	Sim	
SDK para .NET 3.x	Sim	
SDK para PHP 3.x	Sim	
SDK para Python (Boto3)	Sim	
SDK para Ruby 3.x	Sim	
SDK para Rust	Sim	
SDK para Swift	Sim	

SDK	C	Notas ou mais informações
Ferramentas para PowerShell I V5	Sim	
Ferramentas para PowerShell I V4	Sim	

Metadados da instância do Amazon EC2

Note

Para obter ajuda na compreensão do layout das páginas de configurações ou na interpretação da tabela Support by AWS SDKs and tools a seguir, consulte [Entender as páginas de configurações deste guia](#).

O Amazon EC2 fornece um serviço em instâncias chamado serviço de metadados de instância (IMDS). Para saber mais sobre esse serviço, consulte [Trabalhar com metadados de instância](#) no Guia do usuário do Amazon EC2. Ao tentar recuperar credenciais em uma instância do Amazon EC2 que foi configurada com um perfil do IAM, uma conexão ao serviço de metadados da instância é ajustável.

Configure essa funcionalidade usando o seguinte:

metadata_service_num_attempts- configuração de AWS **config** arquivo compartilhado,
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- configuração de AWS **config** arquivo compartilhado,
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
```

Support by AWS SDKs and tools

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 sistema JVM são suportadas pelo AWS SDK para Java e pelo AWS SDK para Kotlin único.

SDK	Cl	Notas ou mais informações
AWS CLI v2	Sim	
SDK para C++	Não	

SDK	Compartilhado	Notas ou mais informações
SDK para Go V2 (1.x)	Não	
SDK para Go 1.x (V1)	Não	
SDK para Java 2.x	Parcial	Somente AWS_METADATA_SERVICE_TIMEOUT é suportado.
SDK para Java 1.x	Parcial	Somente AWS_METADATA_SERVICE_TIMEOUT é suportado.
SDK para 3.x JavaScript	Não	
SDK para 2.x JavaScript	Não	
SDK para Kotlin	Não	
SDK para .NET 4.x	Não	
SDK para .NET 3.x	Não	
SDK para PHP 3.x	Sim	
SDK para Python (Boto3)	Sim	
SDK para Ruby 3.x	Não	
SDK para Rust	Não	
SDK para Swift	Não	
Ferramentas para PowerShell I V5	Não	
Ferramentas para PowerShell I V4	Não	

Pontos de acesso Amazon S3

Note

Para obter ajuda na compreensão do layout das páginas de configurações ou na interpretação da tabela Support by AWS SDKs and tools a seguir, consulte [Entender as páginas de configurações deste guia](#).

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 operações de API em vez de especificar o nome do bucket explicitamente. Eles são usados para operações específicas, como usar um ponto de acesso ARN com o [GetObject](#) para buscar um objeto de um bucket ou usar um ponto de acesso ARN com o [PutObject](#) para adicionar um objeto a um bucket.

Para saber mais sobre os pontos de acesso do Amazon S3 e ARNs, consulte [Uso de pontos de acesso no Guia](#) do usuário do Amazon S3.

Configure essa funcionalidade usando o seguinte:

s3_use_arn_region- configuração de AWS **config** arquivo compartilhado,

AWS_S3_USE_ARN_REGION: variável de ambiente, **aws.s3UseArnRegion**- Propriedade do sistema JVM: somente Java/Kotlin , Para configurar o valor diretamente no código, consulte diretamente seu SDK específico.

Essa configuração controla se o SDK usa o Região da AWS ARN do ponto de acesso para construir o endpoint regional para a solicitação. O SDK valida que o ARN Região da AWS é servido pela mesma AWS partição configurada 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**— O SDK usa os ARNs Região da AWS ao construir o endpoint em vez do configurado pelo cliente. Região da AWS Exceção: se a configuração do cliente Região da AWS for FIPS

Região da AWS, ela deverá corresponder aos ARNs. Caso contrário, ocorrerá um erro.

- **false**: o SDK usa a Região da AWS configurada pelo cliente ao construir o endpoint.

Support by AWS SDKs and tools

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 sistema JVM são suportadas pelo AWS SDK para Java e pelo AWS SDK para Kotlin único.

SDK	Compatível	Notas ou mais informações
AWS CLI v2	Sim	
SDK para C++	Sim	
SDK para Go V2 (1.x)	Sim	
SDK para Go 1.x (V1)	Sim	Para usar as configurações do arquivo config compartilhado, você deve ativar o carregamento do arquivo de configuração; consulte Sessões .
SDK para Java 2.x	Sim	
SDK para Java 1.x	Sim	Propriedade do sistema de JVM não compatível.
SDK para 3.x JavaScript	Sim	
SDK para 2.x JavaScript	Sim	
SDK para Kotlin	Sim	
SDK para .NET 4.x	Sim	
SDK para .NET 3.x	Sim	Não segue a precedência padrão, o valor do arquivo compartilhado config tem precedência sobre a variável de ambiente.
SDK para PHP 3.x	Sim	

SDK	C	Notas ou mais informações
SDK para Python (Boto3)	Sim	
SDK para Ruby 3.x	Sim	
SDK para Rust	Não	
SDK para Swift	Não	
Ferramentas para PowerShell V5	Sim	Não segue a precedência padrão, o valor do arquivo compartilhado config tem precedência sobre a variável de ambiente.
Ferramentas para PowerShell V4	Sim	Não segue a precedência padrão, o valor do arquivo compartilhado config tem precedência sobre a variável de ambiente.

Pontos de acesso de várias regiões do Amazon S3

Note

Para obter ajuda na compreensão do layout das páginas de configurações ou na interpretação da tabela Support by AWS SDKs and tools a seguir, consulte [Entender as páginas de configurações deste guia](#).

Os pontos de acesso multirregionais do Amazon S3 fornecem um endpoint global que as aplicações podem usar para atender a solicitações de buckets do S3 localizados em várias Regiões da AWS. Você pode usar pontos de acesso multirregionais para criar aplicações de várias regiões com a mesma arquitetura usada em uma única região e, em seguida, executar essas aplicações 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. Este algoritmo é obtido pelo SDK por meio de uma dependência em [AWS Bibliotecas do Common Runtime \(CRT\)](#).

Configure essa funcionalidade usando o seguinte:

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

Esta configuração controla se o SDK pode tentar 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.

Support by AWS SDKs and tools

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 sistema JVM são suportadas pelo AWS SDK para Java e pelo AWS SDK para Kotlin único.

SDK	Cl	Notas ou mais informações
AWS CLI v2	Sim	
SDK para C++	Sim	
SDK para Go V2 (1.x)	Sim	

SDK	Cl	Notas ou mais informações
SDK para Go 1.x (V1)	Nãc	
SDK para Java 2.x	Sim	
SDK para Java 1.x	Nãc	
SDK para 3.x JavaScript	Sim	
SDK para 2.x JavaScript	Nãc	
SDK para Kotlin	Sim	
SDK para .NET 4.x	Sim	
SDK para .NET 3.x	Sim	
SDK para PHP 3.x	Sim	
SDK para Python (Boto3)	Sim	
SDK para Ruby 3.x	Sim	
SDK para Rust	Sim	
SDK para Swift	Nãc	
Ferramentas para PowerShell V5	Sim	
Ferramentas para PowerShell V4	Sim	

Autenticação de sessão da S3 Express One Zone

Note

Para obter ajuda na compreensão do layout das páginas de configurações ou na interpretação da tabela Support by AWS SDKs and tools a seguir, consulte [Entender as páginas de configurações deste guia](#).

O S3 Express One Zone é a classe de armazenamento de alta performance do Amazon S3 que fornece latência abaixo de dez milissegundos para dados acessados com frequência. Quando você usa buckets AWS SDKs e ferramentas do S3 Express One Zone, usa automaticamente a autenticação baseada em sessão, otimizada para autorização de baixa latência de solicitações de dados. Você usa tokens de sessão com operações zonais (nível de objeto) para distribuir a latência associada à autorização em várias solicitações em uma sessão, reduzindo a sobrecarga de autenticação e melhorando a performance geral de solicitação.

Os buckets do S3 Express One Zone usam um formato de nomenclatura específico que inclui o ID da zona de disponibilidade, como `bucket-name--usw2-az1--x-s3`. Quando o SDK detecta esse padrão de nomenclatura, encaminha automaticamente as solicitações para os endpoints apropriados do S3 Express One Zone e aplica o fluxo de autenticação otimizado. A autenticação da sessão cria credenciais temporárias específicas do bucket que fornecem acesso de baixa latência ao bucket e são armazenadas em cache e atualizadas automaticamente pelo SDK. Consulte [S3 Express One Zone](#) no Guia do usuário do Amazon S3 para saber mais.

Por padrão, a autenticação de sessão está habilitada para buckets do S3 Express One Zone.

Configure essa funcionalidade usando o seguinte:

s3_disable_express_session_auth- configuração de AWS **config** arquivo compartilhado, **AWS_S3_DISABLE_EXPRESS_SESSION_AUTH**: variável de ambiente, **aws.disableS3ExpressAuth**- Propriedade do sistema JVM: somente Java/Kotlin

Controla se a autenticação da sessão da S3 Express One Zone está desabilitada. Quando definido como `true`, o SDK usa a autenticação SigV4 padrão para buckets do S3 Express One Zone em vez da autenticação de sessão.

Valor padrão: `false`

Valores válidos:

- **true**: desabilitar a autenticação de sessão da S3 Express One Zone.
- **false**: habilitar a autenticação de sessão da S3 Express One Zone.

Exemplo de configuração desse valor no arquivo `config`:

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

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

```
export AWS_S3_DISABLE_EXPRESS_SESSION_AUTH=true
```

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

```
setx AWS_S3_DISABLE_EXPRESS_SESSION_AUTH true
```

Support by AWS SDKs and tools

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 sistema JVM são suportadas pelo AWS SDK para Java e pelo AWS SDK para Kotlin único.

SDK	Compatível	Notas ou mais informações
AWS CLI v2	Sim	
AWS CLI v1	Sim	
SDK para C++	Sim	
SDK para Go V2 (1.x)	Sim	
SDK para Go 1.x (V1)	Não	Para usar as configurações do arquivo <code>config</code> compartilhado, você deve ativar o carregamento do arquivo de configuração; consulte Sessões .
SDK para Java 2.x	Sim	

SDK	Compl	Notas ou mais informações
SDK para Java 1.x	Não	
SDK para 3.x JavaScript	Sim	
SDK para 2.x JavaScript	Não	
SDK para Kotlin	Sim	A propriedade do sistema de JVM é <code>aws.s3DisableExpressSessionAuth</code> .
SDK para .NET 4.x	Sim	
SDK para .NET 3.x	Sim	
SDK para PHP 3.x	Sim	
SDK para Python (Boto3)	Sim	
SDK para Ruby 3.x	Sim	
SDK para Rust	Sim	
SDK para Swift	Sim	
Ferramentas para PowerShell V5	Sim	
Ferramentas para PowerShell V4	Sim	

Esquema de autenticação

Note

Para obter ajuda na compreensão do layout das páginas de configurações ou na interpretação da tabela Support by AWS SDKs and tools a seguir, consulte [Entender as páginas de configurações deste guia](#).

AWS os serviços oferecem suporte a vários esquemas de autenticação, como AWS Signature Version 4 (SigV4) e AWS Signature Version 4a (SigV4a). Por padrão, SDKs seleciona esquemas de autenticação com base nas definições do modelo de serviço e prioriza esquemas que ofereçam a melhor compatibilidade. Porém, você pode configurar seu esquema de autenticação preferido para otimizá-lo para requisitos específicos.

Ao contrário do SigV4, as solicitações assinadas com o SigV4a são válidas em várias Regiões da AWS. O SigV4a fornece maior disponibilidade por meio da assinatura de solicitação inter-regional, o que permite o failover automático para regiões de reserva durante interrupções regionais. Isso é particularmente benéfico para serviços globais como AWS Identity and Access Management e Amazon CloudFront.

Para ter mais informações sobre esses dois esquemas de autenticação, consulte [AWS Signature Version 4 para solicitações de API](#) no Guia do usuário do IAM.

Configure essa funcionalidade usando o seguinte:

auth_scheme_preference- configuração de AWS **config** arquivo compartilhado,
AWS_AUTH_SCHEME_PREFERENCE: variável de ambiente, **aws.authSchemePreference**-
Propriedade do sistema JVM: somente Java/Kotlin

Especifica uma lista separada por vírgulas de esquemas de autenticação preferenciais em ordem de prioridade. Quando um serviço é compatível com vários esquemas de autenticação, o SDK tenta usar os esquemas dessa lista na ordem especificada, voltando ao comportamento padrão se nenhum dos esquemas preferenciais estiver disponível.

Valor padrão: Nenhum.

Valores válidos: uma lista separada por vírgulas de um ou mais dos seguintes itens:

- **sigv4**: Signature Version 4 (performance mais rápida, região única)

- **sigv4a**: Signature Version 4a (disponibilidade aprimorada, compatibilidade inter-regional, tem uma performance de assinatura mais lenta que o SigV4)
- **httpBearerAuth**: Autenticação com token de portador HTTP

Os caracteres espaço e tabulação entre nomes de esquemas são ignorados.

Exemplo de configuração desse valor no arquivo `config` para preferir o SigV4a:

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

sigv4a_signing_region_set- configuração de AWS **config** arquivo compartilhado, **AWS_SIGV4A_SIGNING_REGION_SET**: variável de ambiente

Especifica uma lista separada por vírgulas Regiões da AWS para assinatura multirregional SigV4a. Isso é usado como a região padrão definida para a solicitação se o SigV4a for o esquema de autenticação selecionado.

Valor padrão: determinado pela solicitação.

Valores válidos: lista separada por vírgulas de Regiões da AWS. Os caracteres espaço e tabulação entre nomes de regiões são ignorados.

Support by AWS SDKs and tools

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 sistema JVM são suportadas pelo AWS SDK para Java e pelo AWS SDK para Kotlin único.

SDK	Compartilhado	Notas ou mais informações
AWS CLI v2	Sim	
SDK para C++	Sim	
SDK para Go V2 (1.x)	Sim	
SDK para Go 1.x (V1)	Não	

SDK	Categorias Notas ou mais informações
SDK para Java 2.x	Sim
SDK para Java 1.x	Não
SDK para 3.x JavaScript	Sim
SDK para 2.x JavaScript	Não
SDK para Kotlin	Sim
SDK para .NET 4.x	Sim
SDK para .NET 3.x	Não
SDK para PHP 3.x	Sim
SDK para Python (Boto3)	Sim
SDK para Ruby 3.x	Sim
SDK para Rust	Sim
SDK para Swift	Sim
Ferramentas para PowerShell V5	Sim
Ferramentas para PowerShell V4	Não

Região da AWS

Note

Para obter ajuda na compreensão do layout das páginas de configurações ou na interpretação da tabela Support by AWS SDKs and tools a seguir, consulte [Entender as páginas de configurações deste guia](#).

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, como o IAM, 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 um cliente class/object para cada um. AWS service (Serviço da AWS) Se precisar acessar o Amazon EC2, por exemplo, seu aplicativo criará um objeto cliente do Amazon EC2 para interagir com esse serviço.

Se nenhuma região for especificada explicitamente para o cliente, ele usará por padrão a região definida por meio da configuração `region` a seguir. 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 definida durante a instanciação desse cliente, específica para seu SDK (consulte o Guia do seu SDK ou a base de código do seu SDK).

Configure essa funcionalidade usando o seguinte:

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

Especifica o padrão Região da AWS a ser usado para AWS solicitações. Essa região é usada para solicitações de serviço do SDK 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 desenvolvedores do AWS SDK.

Support by AWS SDKs and tools

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 sistema JVM são suportadas pelo AWS SDK para Java e pelo AWS SDK para Kotlin único.

SDK	C	Notas ou mais informações
AWS CLI v2	Sim	AWS CLI v2 usa qualquer valor em <code>AWS_REGION</code> antes de qualquer valor em <code>AWS_DEFAULT_REGION</code> (ambas as variáveis são verificadas).
AWS CLI v1	Sim	AWS CLI v1 usa uma variável de ambiente nomeada <code>AWS_DEFAULT_REGION</code> para essa finalidade.
SDK para C++	Sim	
SDK para Go V2 (1.x)	Sim	
SDK para Go 1.x (V1)	Sim	Para usar as configurações do arquivo <code>config</code> compartilhado, você deve ativar o carregamento do arquivo de configuração; consulte Sessões .
SDK para Java 2.x	Sim	
SDK para Java 1.x	Sim	
SDK para 3.x JavaScript	Sim	
SDK para 2.x JavaScript	Sim	
SDK para Kotlin	Sim	
SDK para .NET 4.x	Sim	

SDK	Cl	Notas ou mais informações
SDK para .NET 3.x	Sim	
SDK para PHP 3.x	Sim	
SDK para Python (Boto3)	Sim	Este SDK usa uma variável de ambiente nomeada <code>AWS_DEFAULT_REGION</code> para essa finalidade.
SDK para Ruby 3.x	Sim	
SDK para Rust	Sim	
SDK para Swift	Sim	
Ferramentas para PowerShell V5	Sim	
Ferramentas para PowerShell V4	Sim	

AWS STS Endpoints regionais

Note

Para obter ajuda na compreensão do layout das páginas de configurações ou na interpretação da tabela Support by AWS SDKs and tools a seguir, consulte [Entender as páginas de configurações deste guia](#).

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}`). Nas regiões [habilitadas por padrão](#), as solicitações para o endpoint AWS STS global são atendidas automaticamente na mesma região de origem da solicitação. Nas regiões opcionais, as solicitações para o endpoint AWS STS global são atendidas por um único Região da AWS Leste dos EUA (Norte da Virgínia). Para obter mais

informações sobre AWS STS endpoints, consulte [Endpoints](#) na Referência da AWS Security Token Service API ou [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 seus [Região da AWS](#). Clientes em [partições](#) que não são 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 ter mais informações.

Note

AWS fez alterações no endpoint global AWS Security Token Service (AWS STS) (<https://sts.amazonaws.com>) em regiões [habilitadas por padrão](#) para aprimorar sua resiliência e desempenho. AWS STS as solicitações para o endpoint global são atendidas automaticamente da Região da AWS mesma forma que suas cargas de trabalho. Essas alterações não serão implantadas em regiões opt-in. Recomendamos que você use os endpoints AWS STS regionais apropriados. Para obter mais informações, consulte [Alterações em endpoints globais do AWS STS](#) no Guia do usuário do AWS Identity and Access Management .

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 o SDK ou a ferramenta determina o AWS service (Serviço da AWS) endpoint que ele usa para se comunicar com o AWS Security Token Service ().AWS STS

Valor padrão: `regional`, veja as exceções na tabela a seguir.

Note

Todas as novas versões principais do SDK lançadas após julho de 2022 terão como padrão `regional`. As novas versões principais do SDK podem remover essa configuração e usar o comportamento `regional`. 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**— O SDK ou a ferramenta 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 `regionalsts.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
```

Support by AWS SDKs and tools

Note

É uma prática AWS recomendada usar endpoints regionais sempre que possível e configurar seus [Região da AWS](#).

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

- **Compatível com a configuração:** se a variável do arquivo compartilhado `config` e a variável de ambiente para endpoints regionais do STS são compatíveis.
- **Valor da configuração padrão:** o valor padrão da configuração, se for compatível.
- **Endpoint do STS 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 o 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 usa também os seguintes valores:

- **Endpoint global:** `https://sts.amazonaws.com`.
- **Endpoint regional:** com base na [Região da AWS](#) configurada usada pela aplicação.
- **us-east-1 (Regional):** usa o endpoint da região `us-east-1`, mas com tokens de sessão mais longos que as solicitações globais comuns.

SDK	Valor padrão da configuração	Endpoint do STS de destino de cliente de serviço padrão	Comportamento de fallback de cliente de serviço	Notas ou mais informações
AWS CLI v2	Não N/D	Endpoint regional	Endpoint global	
AWS CLI v1	Sim legacy	Endpoint global	Endpoint global	
SDK para C++	Não N/D	Endpoint regional	us-east-1 (Regional)	
SDK para Go V2 (1.x)	Não N/D	Endpoint regional	Falha na solicitação	

SDK		Valor padrão da configuração	Endpoint do STS de destino de cliente de serviço padrão	Comportamento de fallback de cliente de serviço	Notas ou mais informações
SDK para Go 1.x (V1)	Sim	legacy	Endpoint global	Endpoint global	Para usar as configurações do arquivo config compartilhado, você deve ativar o carregamento do arquivo de configuração; consulte Sessões .
SDK para Java 2.x	Não	N/D	Endpoint regional	Falha na solicitação	Se nenhuma região estiver configurada, AssumeRole e AssumeRoleWithWebIdentity usarão o endpoint global do STS.
SDK para Java 1.x	Sim	legacy	Endpoint global	Endpoint global	
SDK para JavaScript 3.x	Não	N/D	Endpoint regional	us-east-1 (Regional)	
SDK para JavaScript 2.x	Sim	legacy	Endpoint global	Endpoint global	
SDK para Kotlin	Não	N/D	Endpoint regional	Endpoint global	
SDK para .NET 4.x	Não	N/D	Endpoint regional	us-east-1 (Regional)	

SDK		Valor padrão da configuração	Endpoint do STS de destino de cliente de serviço padrão	Comportamento de fallback de cliente de serviço	Notas ou mais informações
SDK para .NET 3.x	Sim	regional	Endpoint global	Endpoint global	
SDK para PHP 3.x	Sim	regional	Endpoint global	Falha na solicitação	
SDK para Python (Boto3)	Sim	regional	Endpoint global	Endpoint global	
SDK para Ruby 3.x	Sim	regional	Endpoint regional	Falha na solicitação	
SDK para Rust	Não	N/D	Endpoint regional	Falha na solicitação	
SDK para Swift	Não	N/D	Endpoint regional	Falha na solicitação	
Ferramentas para PowerShell V5	Sim	regional	Endpoint global	Endpoint global	
Ferramentas para PowerShell V4	Sim	regional	Endpoint global	Endpoint global	

Proteções de integridade de dados para o Amazon S3

Note

Para obter ajuda na compreensão do layout das páginas de configurações ou na interpretação da tabela Support by AWS SDKs and tools a seguir, consulte [Entender as páginas de configurações deste guia](#).

Há algum tempo, AWS SDKs suportamos verificações de integridade de dados ao carregar ou baixar dados do Amazon Simple Storage Service. Antes, essas verificações eram opcionais. Agora, habilitamos essas verificações por padrão, usando algoritmos baseados em CRC, como CRC32 ou CRC64 NVME. Embora cada SDK ou ferramenta tenha um algoritmo padrão, você pode escolher um algoritmo diferente. Se desejar, você também pode continuar fornecendo manualmente uma soma de verificação pré-calculada para os uploads. O comportamento consistente entre uploads, uploads em várias partes, downloads e modos de criptografia simplifica as verificações de integridade do lado do cliente.

As versões mais recentes do nosso AWS SDKs calculam AWS CLI automaticamente uma [soma de verificação baseada em verificação de redundância cíclica \(CRC\)](#) para cada upload e a enviam para o Amazon S3. Em seguida, o Amazon S3 calcula independentemente um valor de soma de verificação do objeto do lado do servidor e o valida com o valor fornecido antes de armazenar o objeto e o valor da soma de verificação nos metadados do objeto. Ao armazenar a soma de verificação nos metadados junto com o objeto, quando o objeto é baixado, a mesma soma de verificação também pode ser retornada automaticamente e usada para validar os downloads. Você também pode confirmar a soma de verificação armazenada nos metadados do objeto a qualquer momento.

Para saber mais sobre operações de soma de verificação, uploads com várias partes ou a lista de algoritmos de soma de verificação compatíveis, consulte [Verificar a integridade do objeto no Amazon S3](#) Guia do usuário do Amazon Simple Storage Service.

Uploads com várias partes:

O Amazon S3 também fornece aos desenvolvedores somas de verificação completas e consistentes entre os objetos em uploads com uma única parte e com várias partes.

Ao fazer upload de arquivos em várias partes, SDKs calcule as somas de verificação para cada parte. O Amazon S3 usa essas somas de verificação para verificar a integridade de cada parte por

meio da API `UploadPart`. Além disso, o Amazon S3 valida o tamanho e a soma de verificação do arquivo inteiro quando você chama a API `CompleteMultipartUpload`.

Se o SDK tiver um Amazon S3 Transfer Manager para auxiliar nos uploads de várias partes, as somas de verificação serão validadas para as partes usando o algoritmo padrão específico do SDK encontrado na tabela [Support by AWS SDKs and tools](#). Você pode optar por uma soma de verificação completa do objeto definindo `checksum_type` a configuração `FULL_OBJECT` ou optando por usar o algoritmo CRC64 NVME.

Se você estiver usando uma versão mais antiga de SDK ou AWS CLI:

Se seu aplicativo usa uma versão anterior a dezembro de 2024 do SDK ou da ferramenta, o Amazon S3 ainda computa CRC64 uma soma de verificação NVME em novos objetos e a armazena nos metadados do objeto para referência futura. Você pode comparar depois o CRC armazenado com um CRC calculado do seu lado e verificar se a transmissão da rede estava correta. Além disso, você ainda pode estender manualmente a proteção de integridade fornecendo suas próprias somas de verificação computadas previamente com suas solicitações de [PutObject](#) ou [UploadPart](#), que é a técnica padrão para lidar com isso nas versões mais antigas.

Configure essa funcionalidade usando o seguinte:

request_checksum_calculation- configuração de AWS **config** arquivo compartilhado, **AWS_REQUEST_CHECKSUM_CALCULATION**: variável de ambiente, **aws.requestChecksumCalculation**- Propriedade do sistema JVM: somente Java/Kotlin

Por padrão, os usuários optam por calcular a soma de verificação da solicitação ao enviar uma solicitação. O usuário pode escolher um dos [algoritmos de soma de verificação disponíveis](#) como parte da criação da solicitação. Caso contrário, é usado um algoritmo padrão específico do SDK. Consulte a tabela [Support by AWS SDKs and tools](#) para ver o algoritmo padrão para cada SDK ou ferramenta.

Valor padrão: `WHEN_SUPPORTED`

Valores válidos:

- **WHEN_SUPPORTED**: a validação da soma de verificação é realizada em todas as cargas úteis de solicitação quando compatível com a operação da API, como transferências de dados para o Amazon S3.
- **WHEN_REQUIRED**: a validação da soma de verificação é realizada somente quando exigido pela operação da API.

response_checksum_validation- configuração de AWS **config** arquivo compartilhado, **AWS_RESPONSE_CHECKSUM_VALIDATION**: variável de ambiente, **aws.responseChecksumValidation**- Propriedade do sistema JVM: somente Java/Kotlin

Por padrão, os usuários optam por uma validação da soma de verificação ao enviar uma solicitação. Uma soma de verificação para a carga útil da resposta é calculada e comparada com o cabeçalho da resposta da soma de verificação. Se houver falha na validação da soma de verificação, será gerado um erro para o usuário quando a carga útil for lida.

O cabeçalho da resposta da soma de verificação também indica o algoritmo para a soma de verificação. O cliente Amazon S3 tenta validar somas de verificação de resposta para todas as operações de API do Amazon S3 compatíveis com somas de verificação. Porém, se o SDK não tiver implementado o algoritmo de soma de verificação especificado, essa validação será ignorada.

Valor padrão: `WHEN_SUPPORTED`

Valores válidos:

- **WHEN_SUPPORTED**: a validação da soma de verificação é realizada em todas as cargas úteis de resposta quando compatível com a operação de API, como transferências de dados para o Amazon S3.
- **WHEN_REQUIRED**: a validação da soma de verificação é realizada somente quando compatível com a operação de API e quando o chamador habilitou explicitamente a soma de verificação para a operação. Por exemplo, quando a API `GetObject` do Amazon S3 é chamada e o parâmetro `ChecksumMode` é definido como habilitado.

Support by AWS SDKs and tools

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 sistema JVM são suportadas pelo AWS SDK para Java e pelo AWS SDK para Kotlin único.

Note

Na tabela a seguir, "CRT" se refere ao [AWS Bibliotecas do Common Runtime \(CRT\)](#) e pode exigir a adição de uma dependência adicional ao projeto.

SDK	Compilado	Algoritmo da soma de verificação padrão	Algoritmos de soma de verificação compatíveis	Notas ou mais informações
AWS CLI v2	Sim	CRC64NVME	CRC64NVME, CRC32 C,, CRC32, xHash3 SHA1, xHash64 SHA256, xHash128, SHA512	Para a AWS CLI v1, o algoritmo padrão e os algoritmos suportados serão idênticos ao Python (Boto3).
SDK para C++	Sim	CRC64NVME	CRC64NVME, CRC32 C,, CRC32, xHash3 SHA1, xHash64 SHA256, xHash128, SHA512	
SDK para Go V2 (1.x)	Sim	CRC32	CRC64NVME, CRC32 C CRC32, SHA1 SHA256	
SDK para Go 1.x (V1)	Não			
SDK para Java 2.x	Sim	CRC32	CRC32, CRC32 C, SHA1, SHA256 Somente via CRT: CRC64 NVME, xxHash3, xxHash64, xxHash128, SHA512	
SDK para Java 1.x	Não			
SDK para 3.x JavaScript	Sim	CRC32	CRC32, CRC32 C, SHA1, SHA256	

SDK	Compatível	Algoritmo da soma de verificação padrão	Algoritmos de soma de verificação compatíveis	Notas ou mais informações
SDK para 2.x JavaScript	Não			
SDK para Kotlin	Sim	CRC32	CRC32, CRC32 C, SHA1, SHA256	
SDK para .NET 4.x	Sim	CRC32	CRC32, CRC32 C, SHA1, SHA256, SHA512	
SDK para .NET 3.x	Sim	CRC32	CRC32, CRC32 C, SHA1, SHA256, SHA512	
SDK para PHP 3.x	Sim	CRC32	CRC32, SHA1, SHA256 Somente via CRT: CRC32 C	aws_crt extensão é necessária para usar CRC32 C.
SDK para Python (Boto3)	Sim	CRC32	CRC32, SHA1, SHA256 Somente via CRT: CRC32 C, CRC64 NVME, xxHash3, xxHash64, xxHash128, SHA512	

SDK	Compilado	Algoritmo da soma de verificação padrão	Algoritmos de soma de verificação compatíveis	Notas ou mais informações
SDK para Ruby 3.x	Sim	CRC32	CRC32, SHA1, SHA256 Somente via CRT: CRC64 NVME, C CRC32	
SDK para Rust	Sim	CRC32	CRC64NVME, CRC32 C CRC32, SHA1 SHA256	
SDK para Swift	Sim	CRC32	CRC64NVME, CRC32 C CRC32, SHA1 SHA256	Dependência de CRT requerida para todos os algoritmos.
Ferramentas para PowerShell V5	Sim	CRC32	CRC32, CRC32 C, SHA1 SHA256, xHash3, SHA512	
Ferramentas para PowerShell V4	Sim	CRC32	CRC32, CRC32 C, SHA1, SHA256, SHA512	

Endpoints de pilha dupla e FIPS

Note

Para obter ajuda na compreensão do layout das páginas de configurações ou na interpretação da tabela Support by AWS SDKs and tools a seguir, consulte [Entender as páginas de configurações deste guia](#).

Configure essa funcionalidade usando o seguinte:

use_dualstack_endpoint- configuração de AWS **config** arquivo compartilhado, **AWS_USE_DUALSTACK_ENDPOINT**: variável de ambiente, **aws.useDualstackEndpoint**- Propriedade do sistema JVM: 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**: o SDK ou a ferramenta 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**: o SDK ou a ferramenta não usarão endpoints de pilha dupla para fazer solicitações de rede.

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

Ativa ou desativa se o SDK enviará solicitações para endpoints de pilha dupla. Os Padrões federais de processo de informação (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 diretrizes do FIPS. Diferentemente dos AWS endpoints padrão, os endpoints FIPS usam uma biblioteca de software TLS validada em relação ao FIPS 140. Se essa configuração estiver ativada e não existir um endpoint FIPS para o serviço em sua Região da AWS, a AWS chamada poderá falhar. [Endpoints específicos de serviço](#) e a `--endpoint-url` opção de AWS Command Line Interface substituir essa configuração.

Para saber mais sobre outras formas de especificar endpoints FIPS por Região da AWS, consulte [FIPS Endpoints](#) por serviço. Para obter mais informações sobre os endpoints do serviço Amazon Elastic Compute Cloud, consulte endpoints de [pilha dupla \(IPv4 e IPv6\) na Referência de API do Amazon EC2](#).

Valor padrão: `false`

Valores válidos:

- **true**: o SDK ou a ferramenta enviará solicitações para endpoints compatíveis com FIPS.
- **false**: o SDK ou a ferramenta não enviará solicitações para endpoints compatíveis com FIPS.

Support by AWS SDKs and tools

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 sistema JVM são suportadas pelo AWS SDK para Java e pelo AWS SDK para Kotlin único.

SDK	Compatível	Notas ou mais informações
AWS CLI v2	Sim	
SDK para C++	Sim	
SDK para Go V2 (1.x)	Sim	
SDK para Go 1.x (V1)	Sim	Para usar as configurações do arquivo config compartilhado, você deve ativar o carregamento do arquivo de configuração; consulte Sessões .
SDK para Java 2.x	Sim	
SDK para Java 1.x	Não	
SDK para 3.x JavaScript	Sim	
SDK para 2.x JavaScript	Sim	
SDK para Kotlin	Sim	
SDK para .NET 4.x	Sim	
SDK para .NET 3.x	Sim	
SDK para PHP 3.x	Sim	

SDK	CI	Notas ou mais informações
SDK para Python (Boto3)	Sim	
SDK para Ruby 3.x	Sim	
SDK para Rust	Sim	
SDK para Swift	Sim	
Ferramentas para PowerShell V5	Sim	
Ferramentas para PowerShell V4	Sim	

Descoberta de endpoint

Note

Para obter ajuda na compreensão do layout das páginas de configurações ou na interpretação da tabela Support by AWS SDKs and tools a seguir, consulte [Entender as páginas de configurações deste guia](#).

SDKs use a descoberta de endpoints para acessar os endpoints de serviço (URLs para acessar vários recursos), mantendo a flexibilidade AWS para alterá-los 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, o SDK faz uma solicitação `DescribeEndpoints` para recuperar os endpoints disponíveis e, em seguida, usa esses endpoints para concluir operações específicas, como `CreateDatabase` ou `CreateTable`.

Configure essa funcionalidade usando o seguinte:

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

Ativa ou desativa a descoberta de endpoints para o DynamoDB.

A descoberta de endpoint é obrigatória no Timestream e opcional no Amazon DynamoDB. O padrão para essa configuração é `true` ou `false`, dependendo de o serviço exigir ou não descoberta de endpoint. O padrão para solicitações do Timestream é `true` e para as solicitações do Amazon DynamoDB é `false`.

Valores válidos:

- **true**: o SDK deve tentar descobrir automaticamente um endpoint para serviços em que a descoberta de endpoint é opcional.
- **false**: o SDK deve tentar descobrir automaticamente um endpoint para serviços em que a descoberta de endpoint é opcional.

Support by AWS SDKs and tools

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 sistema JVM são suportadas pelo AWS SDK para Java e pelo AWS SDK para Kotlin único.

SDK	C	Notas ou mais informações
AWS CLI v2	Sim	
SDK para C++	Sim	
SDK para Go V2 (1.x)	Sim	
SDK para Go 1.x (V1)	Sim	Para usar as configurações do arquivo <code>config</code> compartilhado, você deve ativar o carregamento do arquivo de configuração; consulte Sessões .

SDK	Compatibilidade	Notas ou mais informações
SDK para Java 2.x	Sim	O SDK para Java 2.x usa <code>AWS_ENDPOINT_DISCOVERY_ENABLED</code> para o nome da variável de ambiente.
SDK para Java 1.x	Parcial	Propriedade do sistema de JVM não compatível.
SDK para 3.x JavaScript	Sim	
SDK para 2.x JavaScript	Sim	
SDK para Kotlin	Sim	
SDK para .NET 4.x	Sim	
SDK para .NET 3.x	Sim	
SDK para PHP 3.x	Sim	
SDK para Python (Boto3)	Sim	
SDK para Ruby 3.x	Sim	
SDK para Rust	Parcial	Compatível somente com Timestream.
SDK para Swift	Não	
Ferramentas para PowerShell V5	Sim	
Ferramentas para PowerShell V4	Sim	

Definições gerais da configuração

Note

Para obter ajuda na compreensão do layout das páginas de configurações ou na interpretação da tabela Support by AWS SDKs and tools a seguir, consulte [Entender as páginas de configurações deste guia](#).

SDKs oferecem suporte a algumas configurações gerais que definem os comportamentos gerais do SDK.

Configure essa funcionalidade usando o seguinte:

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

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

Valor padrão: nenhum. (A versão mais recente da API é usada pelo SDK.)

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

O exemplo define uma versão específica da API para dois AWS serviços no `config` arquivo. Essas versões de API são usadas apenas para comandos que são executados sob o perfil que contém essas configurações. Os comandos para qualquer outro serviço usam a versão mais recente da API desse serviço.

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

ca_bundle- configuração de AWS **config** arquivo compartilhado, **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 SSL/TLS conexões.

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 a diferenças na forma como os sistemas operacionais lidam com caminhos e escape de caracteres de caminho, o seguinte é um exemplo de configuração desse valor no arquivo config do 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- configuração de AWS **config** arquivo compartilhado

Especifica como os resultados são formatados nas ferramentas AWS CLI e em outras AWS SDKs .

Valor padrão: json

Valores válidos:

- **json**: a saída é formatada como uma string [JSON](#).
- **yaml**: a saída é formatada como uma string [YAML](#).
- **yaml-stream**: a saída é transmitida e formatada como uma string [YAML](#). 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- configuração de AWS **config** arquivo compartilhado

Especifica se o SDK ou a ferramenta tentará validar os parâmetros da linha de comando antes de enviá-los para o endpoint de serviço da AWS .

Valor padrão: `true`

Valores válidos:

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

Support by AWS SDKs and tools

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 sistema JVM são suportadas pelo AWS SDK para Java e pelo AWS SDK para Kotlin único.

SDK	Compartilhado	Notas ou mais informações
AWS CLI v2	Parcial	<code>api_versions</code> incompatível.
SDK para C++	Sim	
SDK para Go V2 (1.x)	Parcial	<code>api_versions</code> e <code>parameter_validation</code> não são compatíveis.

SDK	Compatibilidade	Notas ou mais informações
SDK para Go 1.x (V1)	Parcial	<code>api_versions</code> e <code>parameter_validation</code> não são compatíveis. Para usar as configurações <code>config</code> do arquivo compartilhado, você deve ativar o carregamento do arquivo de configuração; consulte Sessões .
SDK para Java 2.x	Não	
SDK para Java 1.x	Não	
SDK para 3.x JavaScript	Sim	
SDK para 2.x JavaScript	Sim	
SDK para Kotlin	Não	
SDK para .NET 4.x	Não	
SDK para .NET 3.x	Não	
SDK para PHP 3.x	Sim	
SDK para Python (Boto3)	Sim	
SDK para Ruby 3.x	Sim	
SDK para Rust	Não	
SDK para Swift	Não	
Ferramentas para PowerShell V5	Não	
Ferramentas para PowerShell V4	Não	

Injeção de prefixo de host

Note

Para obter ajuda na compreensão do layout das páginas de configurações ou na interpretação da tabela Support by AWS SDKs and tools a seguir, consulte [Entender as páginas de configurações deste guia](#).

A injeção de prefixo de host é um recurso em que se AWS SDKs acrescenta automaticamente um prefixo ao nome do host dos endpoints de serviço para determinadas operações de API. Esse prefixo pode ser uma string estática ou um valor dinâmico que inclui dados dos parâmetros da solicitação.

Por exemplo, ao usar o Amazon Simple Storage Service para realizar ações em objetos ou buckets do Amazon S3, o SDK substitui o nome e o Conta da AWS ID do bucket no endpoint final da API.

Embora esse comportamento seja necessário para endpoints de AWS serviço normais, ele pode causar problemas ao usar endpoints personalizados, como endpoints VPC ou ferramentas de teste locais. Nesses casos, pode ser necessário desabilitar a injeção de prefixo de host.

Configure essa funcionalidade usando o seguinte:

disable_host_prefix_injection- configuração de AWS **config** arquivo compartilhado, **AWS_DISABLE_HOST_PREFIX_INJECTION**: variável de ambiente, **aws.disableHostPrefixInjection**- Propriedade do sistema JVM: somente Java/Kotlin

Essa configuração controla se o SDK ou a ferramenta modificará o nome do host do endpoint colocando um prefixo de host no início, conforme definido no objeto ou na variável de cliente do SDK.

Valor padrão: `false`

Valores válidos:

- **true**: desabilitar injeção de prefixo de host. O SDK não modificará o nome do host do endpoint.
- **false**: habilitar injeção de prefixo de host. O SDK acrescentará o prefixo de host ao nome do host do endpoint.

Exemplo de configuração desse valor no arquivo `config`:

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

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

```
export AWS_DISABLE_HOST_PREFIX_INJECTION=true
```

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

```
setx AWS_DISABLE_HOST_PREFIX_INJECTION true
```

Exemplos de injeção de prefixo de host

A tabela de exemplos a seguir mostra como SDKs modificar o endpoint final quando a injeção de prefixo do host está ativada e desativada.

- Prefixo de host: o modelo da string de caracteres de propriedade de prefixo de host definida no objeto ou na variável de cliente do SDK no código.
- Entradas: entradas adicionais definidas no objeto ou na variável de cliente do SDK no código.
- Endpoint do cliente: o endpoint derivado do cliente.
- Valor da configuração: valor resolvido para a configuração anterior.
- Endpoint resultante: o endpoint resultante que o cliente do SDK usa para fazer a chamada de API.

Prefixo do host	Entradas	Endpoint do cliente	Valor da configuração	Endpoint resultante
"dados".	{}	"https://service.us-west-2.amazonaws.com"	false	"https://data.service.us-west-2.amazonaws.com"
"{Balde} - {AccountId}."	Caçamba: "amzn-s3-demo-bucket1",:"	"https://service.us-west-2.amazonaws.com"	false	"https://amzn-s3-demo-bucket1-123456789012.se"

Prefixo do host	Entradas	Endpoint do cliente	Valor da configuração	Endpoint resultante
	123456789012" AccountId			vice.us-west-2.amazonaws.com"
"dados".	{	"https://override.us-west-2.amazonaws.com"(como um endpoint de substituição)	true	"https://override.us-west-2.amazonaws.com"

Support by AWS SDKs and tools

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 sistema JVM são suportadas pelo AWS SDK para Java e pelo AWS SDK para Kotlin único.

SDK	Compartilhado	Notas ou mais informações
AWS CLI v2	Sim	
SDK para C++	Não	Configuração não compatível, mas pode ser configurada no código do cliente usando: enableHostPrefixInjection .
SDK para Go V2 (1.x)	Não	Pode ser desabilitado usando middleware .
SDK para Go 1.x (V1)	Não	
SDK para Java 2.x	Não	Configuração não compatível, mas pode ser configurada no código do cliente usando: SdkAdvancedClientOption.DISABLE_HOST_PREFIX_INJECTION .

SDK	Compatível	Notas ou mais informações
SDK para Java 1.x	Não	Configuração não compatível, mas pode ser configurada no código do cliente usando: withDisableHostPrefixInjection .
SDK para 3.x JavaScript	Não	Configuração não compatível, mas pode ser configurada no código do cliente usando: disableHostPrefix .
SDK para 2.x JavaScript	Não	Configuração não compatível, mas pode ser configurada no código do cliente usando: hostPrefixEnabled .
SDK para Kotlin	Não	
SDK para .NET 4.x	Não	Configuração não compatível, mas pode ser configurada no código do cliente usando: DisableHostPrefixInjection .
SDK para .NET 3.x	Não	Configuração não compatível, mas pode ser configurada no código do cliente usando: DisableHostPrefixInjection .
SDK para PHP 3.x	Não	Configuração não compatível, mas pode ser configurada no código do cliente usando: disable_host_prefix_injection .
SDK para Python (Boto3)	Sim	Pode ser configurado em código no cliente usando: inject_host_prefix .
SDK para Ruby 3.x	Não	Configuração não compatível, mas pode ser configurada no código do cliente usando: disable_host_prefix_injection .
SDK para Rust	Não	
SDK para Swift	Não	

SDK	C	Notas ou mais informações
Ferramentas para PowerShell V5	Não	A configuração não é compatível, mas pode ser incluída em cmdlets específicos usando o parâmetro <code>-ClientConfig @{DisableHostPrefixInjection = \$true}</code> .
Ferramentas para PowerShell V4	Não	A configuração não é compatível, mas pode ser incluída em cmdlets específicos usando o parâmetro <code>-ClientConfig @{DisableHostPrefixInjection = \$true}</code> .

Cliente de IMDS

Note

Para obter ajuda na compreensão do layout das páginas de configurações ou na interpretação da tabela Support by AWS SDKs and tools a seguir, consulte [Entender as páginas de configurações deste guia](#).

SDKs implemente um cliente do Instance Metadata Service versão 2 (IMDSv2) usando solicitações orientadas à sessão. Para obter mais informações sobre IMDSv2, consulte [Uso IMDSv2](#) no Guia do usuário do Amazon EC2. O cliente IMDS é configurável por meio de um objeto de configuração do cliente disponível na base de código do SDK.

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 endpoint 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: URL válido.

As opções a seguir são suportadas pela maioria SDKs. Consulte sua base de código específica do SDK 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 I/O fluxo válido, como STDOUT.

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 com o SDK.

Valores válidos: variam de acordo com o SDK. Pode ser um valor numérico ou uma chamada para uma função personalizada.

Support by AWS SDKs and tools

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 sistema JVM são suportadas pelo AWS SDK para Java e pelo AWS SDK para Kotlin único.

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

SDK	Comportamento de repetição	Notas ou mais informações
SDK para Kotlin	Não	
SDK para .NET 4.x	Sim	
SDK para .NET 3.x	Sim	
SDK para PHP 3.x	Sim	
SDK para Python (Boto3)	Sim	
SDK para Ruby 3.x	Sim	
SDK para Rust	Sim	
SDK para Swift	Sim	
Ferramentas para PowerShell V5	Sim	
Ferramentas para PowerShell V4	Sim	

Comportamento de repetição

Important

O comportamento descrito nesta página exige a aceitação até que se torne o comportamento padrão. Defina `AWS_NEW_RETRIES_2026=true` em seu ambiente. Sem essa configuração, seu SDK usa o comportamento de repetição anterior a 2026, que difere no tempo de recuo, nos custos da cota de repetição e nos padrões específicos do serviço. Para obter detalhes, consulte a [postagem do blog do anúncio](#).

Quando uma solicitação para um AWS service (Serviço da AWS) falha devido a um erro transitório ou limitação, o SDK pode repetir a solicitação automaticamente. Esta página aborda como configurar novas tentativas e como elas funcionam internamente.

- [Configurando novas tentativas](#): escolha um modo de repetição, defina o máximo de tentativas e entenda a precedência da configuração.
- [Como funcionam as novas tentativas](#): repita o fluxo, a classificação de erros, a fórmula de recuo, a mecânica da cota de repetição e o comportamento específico do serviço.

Configurando novas tentativas

Você controla qual estratégia de repetição o SDK usa e quantas vezes ele tenta novamente.

Escolhendo um modo de repetição

O modo de nova tentativa determina como o SDK se comporta quando uma solicitação falha. Três modos estão disponíveis: padrão, adaptável e legado.

	Standard	Adaptativo	Legada
Repetir a cota	Sim	Sim	Varia de acordo com o SDK
Pode atrasar a solicitação inicial	Não	Sim	Não
Error-type-specific recuar	Sim	Sim	Varia de acordo com o SDK
Padronizado em todos os SDKs	Sim	Sim	Não
Recomendação	Padrão para todas as cargas de trabalho	Single-resource, com muita limitação de pressão, tolerante à latência	Somente compatibilidade com versões anteriores

Modo padrão (padrão)

O modo padrão repete solicitações com falha usando recuo exponencial com instabilidade. Ele usa atrasos menores para erros transitórios (como tempos limite de rede) e atrasos maiores para erros de limitação (como). `ThrottlingException`

O modo padrão inclui uma cota de repetição, um token bucket que deduz os tokens para cada nova tentativa e os reabastece quando as solicitações são bem-sucedidas. Quando os tokens disponíveis se esgotam, o SDK retorna o erro sem tentar novamente, então seu aplicativo falha rapidamente em vez de esperar por novas tentativas que provavelmente não serão bem-sucedidas. Isso também ajuda a resolver as interrupções do serviço com mais rapidez, reduzindo o tráfego de novas tentativas. Durante a operação normal, a cota permanece cheia e não tem efeito. A cota de repetição nunca atrasa ou bloqueia a solicitação inicial. Somente as novas tentativas são afetadas. Para obter detalhes, consulte [Repetir a cota \(token bucket\)](#).

Use o modo padrão, a menos que você tenha um motivo específico para escolher outro modo.

Modo adaptativo

O modo adaptativo inclui tudo no modo padrão, além de um limitador de taxa do lado do cliente. O limitador de taxa rastreia as respostas de limitação e ajusta a taxa na qual o SDK envia solicitações. Diferentemente do modo padrão, o modo adaptativo pode atrasar ou bloquear a solicitação inicial, e não apenas novas tentativas, quando a limitação é detectada.

O limitador de taxa opera por instância do cliente SDK. Todas as solicitações de um cliente compartilham o mesmo limite de taxa, independentemente de qual operação de API ou recurso elas segmentam.

Quando usar o modo adaptativo:

- Seu cliente tem como alvo um único recurso (por exemplo, uma tabela do DynamoDB) e você espera respostas frequentes de limitação. Isso é comum em fluxos de trabalho automatizados, processadores em lote ou cargas de trabalho de IA que chamam uma única operação de API em alto volume.
- Você quer que o SDK fique automaticamente lento quando o serviço sinalizar limitação.

Quando não usar o modo adaptativo:

- Seu cliente envia solicitações para vários recursos ou atende a vários inquilinos. A limitação de um recurso faz com que o limitador de taxa reduza a velocidade de todas as solicitações desse cliente, incluindo solicitações para recursos não afetados.
- Você precisa de uma latência previsível na solicitação inicial.

O modo adaptativo não é recomendado como padrão geral.

Modo legado

O modo legado é o comportamento de repetição que cada SDK usou antes da introdução do modo padrão. Ela não inclui uma cota de repetição padronizada. Alguns SDKs (como Java) tinham suas próprias implementações de cota de repetição no modo legado, mas o comportamento não é consistente em todos os SDKs. Sem uma cota padronizada, o cliente continua tentando novamente com a taxa máxima durante interrupções no serviço. Isso vincula threads e conexões em solicitações que provavelmente não serão bem-sucedidas, ao mesmo tempo em que adiciona carga que pode atrasar a recuperação do serviço.

O modo legado varia entre os SDKs. A contagem de novas tentativas, o tempo de recuo, os conjuntos de erros que podem ser repetidos e o comportamento de limitação diferem entre os idiomas. O código que depende do comportamento de repetição legado pode se comportar de forma diferente quando movido entre SDKs.

Disponível em: Java, Python, Ruby, PHP, C++, CLI

Não disponível em: .NET, Go, Kotlin, Rust, Swift, JavaScript

O modo legado existe para compatibilidade com versões anteriores. Se você usa atualmente o modo legado, alterne para o modo padrão.

Tentar novamente as configurações

As configurações a seguir controlam o comportamento de repetição. Você pode defini-las por meio de [variáveis de ambiente](#), do [arquivo de configuração compartilhado](#) (~/.aws/config) ou da configuração do cliente no código.

Configuração	O que ele controla	Variável de ambiente	Chave do arquivo de configuração	Padrão
Modo de nova tentativa	Qual estratégia de repetição usar	AWS_RETRY_MODE	retry_mode	standard
Máximo de tentativas	Total de tentativas, incluindo a solicitação inicial	AWS_MAX_ATTEMPTS	max_attempts	3(ver notas)

Um valor máximo de tentativas 3 significa que o SDK faz uma solicitação inicial e até duas novas tentativas. Defina o máximo de tentativas 1 para desativar totalmente as novas tentativas.

Note

Os clientes do DynamoDB e do DynamoDB Streams usam como padrão o máximo de tentativas. 4 Esses serviços usam um atraso de recuo básico menor (25 ms em vez de 50 ms) para corresponder ao perfil de baixa latência. A tentativa adicional mantém o recuo máximo da última tentativa comparável a outros serviços. Você pode substituir isso com as mesmas configurações mostradas na tabela anterior.

Precedência de configuração

Quando você especifica a mesma configuração em vários lugares, o SDK resolve o valor usando a seguinte precedência, da maior para a menor:

1. Configuração explícita do cliente no código. Um valor definido diretamente no cliente SDK ou em seu objeto de configuração.
2. [Variável de ambiente](#). Por exemplo, AWS_RETRY_MODE ou AWS_MAX_ATTEMPTS.
3. Arquivo de [configuração compartilhado](#). A max_attempts chave retry_mode ou ~/.aws/config.
4. SDK padrão. O padrão incorporado para a configuração.

Isso segue a [precedência de configuração padrão do AWS SDK](#). Um valor definido em um nível superior sempre substitui um valor definido em um nível inferior. Por exemplo, se você definir `AWS_RETRY_MODE=adaptive` como variável de ambiente e `retry_mode=standard` em `~/.aws/config`, o SDK usa o modo adaptativo.

Language-specific configuração

As configurações entre SDKs descritas nesta página (`retry_mode` e `max_attempts`) funcionam em todos os SDKs. No entanto, a API para configurar novas tentativas no código varia de acordo com o idioma. Consulte o guia do desenvolvedor do SDK para ver as opções de configuração específicas do idioma, como estratégias de recuo personalizadas, erros adicionais que podem ser tentados novamente e ajuste de cota de repetição.

Como funcionam as novas tentativas

Esta seção descreve como AWS os SDKs lidam com solicitações com falha: quais erros acionam novas tentativas, quanto tempo o SDK espera entre as tentativas e quando ele para de tentar novamente.

O que acontece quando uma solicitação falha

Quando você faz uma chamada de API por meio de um AWS SDK, o SDK segue esta sequência:

1. Somente [no modo adaptativo](#): o SDK verifica o limitador de taxa do lado do cliente. Se a limitação for detectada, o SDK poderá atrasar ou bloquear a solicitação antes de enviá-la.
2. O SDK envia a solicitação para o AWS service (Serviço da AWS) endpoint.
3. Se o serviço retornar uma resposta bem-sucedida, o SDK retornará o resultado para seu código.
4. Se a solicitação falhar, o SDK classifica o erro como transitório, limitador ou não passível de nova tentativa. Consulte [Quais erros são repetidos](#).
5. Se o erro não puder ser repetido, o SDK retornará o erro para seu código imediatamente. Nenhuma tentativa é tentada novamente.
6. Se o erro puder ser repetido, o SDK verificará se ele atingiu o número máximo de tentativas. Nesse caso, ele retornará o erro para o seu código.
7. O SDK verifica o [Repetir a cota \(token bucket\)](#). Se o orçamento do token estiver esgotado, o SDK não tentará novamente e retornará o erro para seu código. Exceção: pois [Long-polling operações](#), o SDK ainda aplica um atraso antes de retornar o erro.
8. O SDK calcula um atraso de recuo com base no tipo de erro e no número da tentativa de repetição. Consulte [Quanto tempo o SDK espera?](#).

9. O SDK aguarda o atraso computado e, em seguida, envia a solicitação novamente a partir da etapa 2.

O SDK repete esse loop até que a solicitação seja bem-sucedida, o número máximo de tentativas seja atingido, a cota de repetição seja esgotada ou ocorra um erro que não pode ser repetido. Todo o processo é automático. Seu aplicativo vê uma resposta bem-sucedida ou um erro final.

Quais erros são repetidos

O SDK classifica cada solicitação com falha em uma das três categorias: transitória, limitadora ou não passível de nova tentativa. Essa classificação determina se o SDK repete a solicitação e quanto tempo espera antes de tentar novamente.

A classificação é baseada no código de erro e no código de status HTTP na resposta do serviço. Por exemplo, um HTTP 400 com o código de erro `RequestTimeout` é classificado como transitório e repetido. Um HTTP 400 com `ValidationException` é classificado como não passível de nova tentativa e retornado imediatamente.

Classificação de erros

Os erros transitórios são repetidos com um pequeno atraso básico (50 ms):

Código de erro

`RequestTimeout`

`RequestTimeoutException`

`InternalError`

`IDPCommunicationError`

I/O Falha (redefinição da conexão, falha na resolução do DNS, tempo limite do soquete)

(qualquer HTTP 500, 502, 503 ou 504 sem um código de erro reconhecido)

Os erros de limitação são repetidos com um atraso base maior (1.000 ms):

Código de erro

Throttling

ThrottlingException

ThrottledException

RequestThrottledException

TooManyRequestsException

ProvisionedThroughputExceededException

TransactionInProgressException

LimitExceededException

PriorRequestNotComplete

RequestThrottled

EC2ThrottledException

RequestLimitExceeded

SlowDown

BandwidthLimitExceeded

Non-retryable erros

(como `AccessDeniedException`, `ValidationException`, `ResourceNotFoundException`) são retornados ao seu código imediatamente.

Note

Um HTTP 5XX com um código de erro de limitação é classificado como um erro de limitação, não um erro transitório, embora os erros 5XX sejam normalmente transitórios. O SDK corresponde primeiro ao código de erro e depois volta ao código de status HTTP.

Erros de limitação significam que o serviço rejeitou ativamente sua solicitação devido aos limites de taxa, então o SDK espera mais tempo antes de tentar dar tempo ao serviço para recuperar a capacidade. Consulte [Quanto tempo o SDK espera?](#) os atrasos específicos.

Quanto tempo o SDK espera?

O SDK usa recuo exponencial com instabilidade total. Em média, cada nova tentativa espera mais do que a anterior, com randomização para distribuir solicitações de vários clientes.

Atrasos básicos por tipo de erro

O atraso básico depende se o erro é transitório ou limitador:

Tipo de erro	Atraso básico	Lógica
Transiente (sem limitação de pressão)	50 ms	Os erros transitórios geralmente são resolvidos em milissegundos. Um pequeno atraso básico proporciona uma recuperação rápida.
Controle de utilização	1.000 ms	O serviço limitou a taxa de solicitação. Um atraso básico maior dá tempo para recuperar a capacidade.

Fórmula de recuo

O SDK calcula cada atraso de repetição usando esta fórmula:

```
delay = random(0, 1) × min(20,000 ms, base_delay × 2^retry)
```

Em que:

- `random(0, 1)` retorna um valor uniformemente distribuído entre 0 e 1
- `base_delay` é de 50 ms para erros transitórios ou 1.000 ms para erros de limitação
- `retry` começa em 0 para a primeira tentativa (a segunda tentativa geral de solicitação)

O limite máximo de recuo é de 20 segundos. Nenhum atraso individual excede 20 segundos, independentemente de quantas tentativas tenham ocorrido.

Exemplos trabalhados

Exemplo 1: erro transitório, máximo de 3 tentativas

Etapa	O que acontece	Delay
Tentativa 1	Solicitação inicial. O serviço retorna HTTP 503.	(none)
Tentativa 2	O SDK espera aleatoriamente (0, 50 ms). A nova tentativa falha com 503.	0—50 ms (média de ~25 ms)
Tentativa 3	O SDK espera aleatoriamente (0, 100 ms). A nova tentativa foi bem-sucedida.	0—100 ms (média de ~50 ms)

A latência total adicionada é em média de cerca de 75 ms em ambas as tentativas.

Exemplo 2: erro de limitação, máximo de 3 tentativas

Etapa	O que acontece	Delay
Tentativa 1	Solicitação inicial. O serviço retorna <code>429Throttling</code> .	(none)
Tentativa 2	O SDK espera aleatoriamente (0, 1.000 ms). Tentar novamente retorna 429.	0—1.000 ms (média de ~500 ms)
Tentativa 3	O SDK espera aleatoriamente (0, 2.000 ms). A nova tentativa foi bem-sucedida.	0—2.000 ms (média de ~1.000 ms)

A latência total adicionada é em média de cerca de 1.500 ms em ambas as tentativas.

Exemplo 3: erro transitório, atingindo a tampa de recuo

Com um atraso base de 50 ms, o atraso calculado antes do limite seria:

Tentativa de novo	Atraso máximo calculado	Depois de 20 s de boné
1	50 ms	50 ms

Tentativa de novo	Atraso máximo calculado	Depois de 20 s de boné
2	100 ms	100 ms
5	800 ms	800 ms
9	12.800 ms	12.800 ms
10	25.600 ms	20.000 ms

O limite entra em vigor na 10ª tentativa (11ª tentativa) para erros transitórios. Para erros de limitação com uma base de 1.000 ms, o limite entra em vigor na 6ª tentativa.

Note

Com o padrão de 3 tentativas máximas (1 solicitação inicial + 2 tentativas), o limite de recuo nunca é atingido. Esta tabela ilustra o que acontece se você aumentar `max_attempts` muito além do padrão.

Por que a instabilidade é importante

O multiplicador aleatório é chamado de instabilidade total. Sem isso, todos os clientes que encontrassem um erro ao mesmo tempo tentariam novamente ao mesmo tempo, criando uma explosão de tráfego de repetições (o problema do “rebanho trovejante”). A instabilidade total distribui as novas tentativas uniformemente por toda a janela de recuo, para que o serviço receba um fluxo constante de solicitações em vez de picos sincronizados.

Por exemplo, suponha que 1.000 clientes recebam um 503 no mesmo momento. A instabilidade total distribui suas primeiras tentativas uniformemente em uma janela de 50 ms, em vez de ter todas as 1.000 repetições em exatamente 50 ms.

Server-directed tempo de repetição

Alguns Serviços da AWS incluem um `x-amz-retry-after` cabeçalho nas respostas de erro. O valor do cabeçalho é um atraso em milissegundos. Quando esse cabeçalho está presente, o SDK usa o atraso especificado pelo servidor, limitado ao mínimo do atraso de recuo computado e ao máximo do atraso de recuo computado mais 5.000 ms. Como o recuo computado em si é limitado a 20 segundos, o atraso máximo efetivo direcionado ao servidor é de 25 segundos. O SDK não

aplica instabilidade a esse valor, pois espera-se que o serviço o altere. Isso permite que o serviço se comunique exatamente quando espera ter capacidade disponível.

Repetir a cota (token bucket)

O SDK mantém um orçamento interno de tokens que rastreia a proporção entre solicitações bem-sucedidas e falhas. Quando as falhas são generalizadas, o orçamento se esgota e o SDK retorna os erros diretamente. Seu aplicativo falha rapidamente em vez de esperar por novas tentativas que provavelmente não serão bem-sucedidas. Isso também reduz o tráfego de novas tentativas, ajudando a resolver as interrupções do serviço com mais rapidez.

Como funciona a cota de repetição

O orçamento do token começa cheio. Cada tentativa de nova tentativa deduz tokens. Quando uma nova tentativa é bem-sucedida, o SDK restaura os tokens consumidos por essa nova tentativa. Quando uma solicitação é bem-sucedida na primeira tentativa (sem necessidade de novas tentativas), o SDK restaura 1 token. Quando o orçamento chega a zero, o SDK para de tentar novamente e retorna os erros diretamente para o seu código.

Parâmetro	Valor
Capacidade orçamentária	500 fichas
Custo por nova tentativa transitória (sem limitação)	14 fichas
Custo por nova tentativa de limitação	5 fichas
Tokens restaurados em caso de sucesso após nova tentativa	Quantidade consumida pela última tentativa (14 ou 5)
Tokens restaurados em caso de sucesso, sem nova tentativa	1 ficha

O custo mais alto das novas tentativas transitórias reflete seus diferentes padrões de falha. Erros transitórios como 500s e falhas de conexão geralmente indicam um problema em todo o serviço. Nessas situações, é improvável que a repetição contínua seja bem-sucedida. Isso adiciona latência às suas chamadas, consome recursos do cliente e pode atrasar a recuperação de todos. Erros de limitação indicam que o serviço precisa de mais tempo para que a solicitação seja bem-sucedida. O SDK espera mais tempo entre as novas tentativas para aumentar a probabilidade de sucesso.

Quando a cota bloqueia novas tentativas?

A cota de repetição rastreia os tokens o tempo todo, mas só bloqueia novas tentativas quando o orçamento está esgotado. Durante a operação normal, quase todas as solicitações são bem-sucedidas e o orçamento permanece cheio. A cota não tem efeito observável nas novas tentativas.

Uma nova tentativa bem-sucedida restaura somente seu próprio custo de token (14 ou 5 tokens), não o custo de tentativas anteriores malsucedidas na mesma solicitação. Por exemplo, se a primeira tentativa falhar e a segunda for bem-sucedida, o orçamento perderá 14 tokens líquidos. O orçamento se esgota mais rapidamente quando as novas tentativas esgotam todas as tentativas sem sucesso, mas também se esgota gradualmente quando as solicitações precisam de várias tentativas antes de serem bem-sucedidas.

Com o padrão de no máximo 3 tentativas, a cota começa a diminuir quando mais de aproximadamente 22% das solicitações resultam em falhas transitórias sustentadas ou mais de aproximadamente 32% em erros de limitação. Abaixo dessas taxas, as solicitações bem-sucedidas reabastecem o orçamento mais rapidamente do que as tentativas malsucedidas o esgotam.

O saldo inicial de 500 tokens do orçamento fornece um buffer que absorve pequenas explosões de falhas. Um breve pico de erros, mesmo que grave, não bloqueia novas tentativas, a menos que persista por tempo suficiente para drenar o buffer.

Implicações práticas

- **Baixas taxas de falha:** a cota não tem efeito. O orçamento permanece na capacidade máxima ou próxima.
- **Durante uma interrupção do serviço:** se uma alta porcentagem de suas solicitações falhar por um período prolongado, a cota se esgota e seu cliente recupera os erros imediatamente, em vez de esperar por novas tentativas. Isso reduz a latência do lado do cliente, libera threads e conexões e ajuda o serviço a se recuperar mais rapidamente.
- **Recuperação:** à medida que o serviço se recupera e as solicitações começam a ser bem-sucedidas novamente, as novas tentativas bem-sucedidas restauram o custo total do token e as tentativas bem-sucedidas restauram 1 token. O orçamento é reabastecido gradualmente e as tentativas são retomadas automaticamente.
- **Escopo:** o orçamento do token normalmente tem como escopo uma única instância do cliente SDK. O escopo exato pode variar de acordo com o SDK. Ele não é compartilhado entre processos ou hosts.

Service-specific comportamento

DynamoDB

Os clientes do DynamoDB usam padrões ajustados otimizados para o perfil de baixa latência do DynamoDB:

Configuração	Padrão geral	Padrão do DynamoDB
Atraso de base transitório (sem limitação)	50 ms	25 ms
Atraso de base de aceleração	1.000 ms	1.000 ms
Máximo de tentativas	3	4

Esses padrões se aplicam tanto ao Amazon DynamoDB quanto ao DynamoDB Streams.

Long-polling operações

Certas AWS operações usam sondagens longas. Eles podem manter uma conexão aberta aguardando a chegada do trabalho. Essas operações recebem tratamento especial de repetição:

- `SQS.ReceiveMessage`
- `SFN.GetActivityTask`
- `SWF.PollForActivityTask`
- `SWF.PollForDecisionTask`

Comportamento especial: quando a cota de repetição se esgota e as tentativas são bloqueadas (etapa [7O que acontece quando uma solicitação falha](#)), o SDK ainda aplica um atraso antes de retornar o erro ao seu código.

Isso é importante porque as operações de sondagem longa geralmente são realizadas em um circuito fechado. Seu código liga `ReceiveMessage`, processa qualquer mensagem e liga imediatamente `ReceiveMessage` novamente. Sem o recuo forçado, um orçamento de token esgotado faria com que o SDK retornasse erros sem demora. Seu ciclo de pesquisa enviaria imediatamente a próxima solicitação, aumentando o uso da CPU do cliente e gerando tráfego

adicional. O atraso forçado interrompe esse ciclo, mantendo o uso de recursos do cliente e a taxa de pesquisa gerenciáveis durante falhas.

Support by AWS SDKs e ferramentas

A tabela a seguir lista a disponibilidade do comportamento de repetição atualizado em cada SDK. Para SDK-specific obter detalhes, incluindo a versão mínima, os padrões de antes e depois e exemplos de código, consulte o problema de rastreamento. GitHub

SDK	Compatível	GitHub problema de rastreamento
SDK para Java 2.x	Sim	Problema de rastreamento
SDK para Python (Boto3)	Sim	Problema de rastreamento
SDK para .NET 4.x	Sim	Problema de rastreamento
Ferramentas para PowerShell V5	Sim	Problema de rastreamento
SDK para 3.x JavaScript	Sim	Problema de rastreamento
SDK para PHP 3.x	Sim	Problema de rastreamento
SDK para Kotlin	Sim	Problema de rastreamento
SDK para Rust	Sim	Problema de rastreamento
SDK para Swift	Veja o problema de rastreamento	Problema de rastreamento
SDK para Ruby 3.x	Veja o problema de rastreamento	Problema de rastreamento
SDK para Go V2 (1.x)	Veja o problema de rastreamento	Problema de rastreamento
SDK para C++	Veja o problema de rastreamento	Problema de rastreamento

SDK	Compatível	GitHub problema de rastreamento
AWS CLI v2	Veja o problema de rastreamento	Problema de rastreamento

Compactação de solicitações

Note

Para obter ajuda na compreensão do layout das páginas de configurações ou na interpretação da tabela Support by AWS SDKs and tools a seguir, consulte [Entender as páginas de configurações deste guia](#).

AWS SDKs e as ferramentas podem compactar cargas automaticamente ao enviar solicitações para Serviços da AWS que suportem 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, o SDK ou a ferramenta 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 oferece suporte à compactação CloudWatch é o método de [PutMetricDataAPI](#).

Configure essa funcionalidade usando o seguinte:

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

Ativa ou desativa se o SDK ou a ferramenta compactarão 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- configuração de AWS **config** arquivo compartilhado, **AWS_REQUEST_MIN_COMPRESSION_SIZE_BYTES**: variável de ambiente, **aws.requestMinCompressionSizeBytes**- Propriedade do sistema JVM: somente Java/Kotlin

Define o tamanho mínimo em bytes do corpo da solicitação que o SDK ou a ferramenta devem 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.

Support by AWS SDKs and tools

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 sistema JVM são suportadas pelo AWS SDK para Java e pelo AWS SDK para Kotlin único.

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

SDK	Compartilhado	Notas ou mais informações
SDK para 2.x JavaScript	Não	
SDK para Kotlin	Sim	
SDK para .NET 4.x	Sim	
SDK para .NET 3.x	Sim	
SDK para PHP 3.x	Sim	
SDK para Python (Boto3)	Sim	
SDK para Ruby 3.x	Sim	
SDK para Rust	Sim	
SDK para Swift	Não	
Ferramentas para PowerShell V5	Sim	
Ferramentas para PowerShell V4	Sim	

Endpoints específicos de serviço

Note

Para obter ajuda na compreensão do layout das páginas de configurações ou na interpretação da tabela Support by AWS SDKs and tools a seguir, consulte [Entender as páginas de configurações deste guia](#).

A configuração de endpoint específico de serviço oferece a opção de usar um endpoint de sua escolha para solicitações de API e para ter a persistência dessa escolha. Essas configurações oferecem flexibilidade para permitir endpoints locais, endpoints da VPC e ambientes de

desenvolvimento da AWS local de terceiros. Diferentes endpoints podem ser usados para ambientes de teste e produção. Você pode especificar um URL de endpoint para Serviços da AWS individuais.

Configure essa funcionalidade usando o seguinte:

endpoint_url- configuração de AWS **config** arquivo compartilhado, **AWS_ENDPOINT_URL**: variável de ambiente, **aws.endpointUrl**- Propriedade do sistema JVM: 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 AWS `config` arquivo compartilhado 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: um URL incluindo o esquema e o host do endpoint. Opcionalmente, o URL pode conter um componente de caminho que contenha um ou mais segmentos de caminho.

AWS_ENDPOINT_URL_<SERVICE>: variável de ambiente, **aws.endpointUrl<ServiceName>**- Propriedade do sistema JVM: 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: um URL incluindo o esquema e o host do endpoint. Opcionalmente, o URL pode conter um componente de caminho que contenha um ou mais segmentos de caminho.

ignore_configured_endpoint_urls- configuração de AWS **config** arquivo compartilhado, **AWS_IGNORE_CONFIGURED_ENDPOINT_URLS**: variável de ambiente, **aws.ignoreConfiguredEndpointUrls**- Propriedade do sistema JVM: 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, incluir o parâmetro da linha de `--endpoint-url` comando com um AWS CLI comando ou passar uma URL de endpoint para um construtor de cliente sempre terá efeito.

Valor padrão: `false`

Valores válidos:

- **true**: o SDK ou a ferramenta não lê nenhuma opção de configuração personalizada do arquivo `config` compartilhado ou das variáveis de ambiente para definir um URL de endpoint.
- **false**: o SDK ou a ferramenta usa todos os endpoints disponíveis fornecidos pelo usuário a partir do arquivo `config` compartilhado ou de variáveis de ambiente.

Configurar endpoints usando variáveis de ambiente

Para rotear solicitações de todos os serviços para um URL de endpoint personalizado, defina a variável de ambiente global `AWS_ENDPOINT_URL`.

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

Para encaminhar solicitações de um URL específico AWS service (Serviço da AWS) para um endpoint personalizado, use a variável de `AWS_ENDPOINT_URL_<SERVICE>` ambiente. Amazon DynamoDB tem um `serviceId` de [DynamoDB](#). Para esse serviço, a variável de ambiente do URL 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](#).

O AWS service (Serviço da AWS) identificador é baseado no modelo de API, 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 definição `services` para configurar um URL de endpoint específico do serviço para o Amazon S3 e um endpoint global personalizado para ser usado para todos os demais 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](#). O AWS service (Serviço da AWS) identificador é baseado no modelo de API, 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 o seu perfil tiver credenciais baseadas em perfis configurados por meio de um parâmetro `source_profile` para a funcionalidade “assumir função” do IAM, o SDK usará 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 o Amazon EC2, 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 chamadas de API 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 um SDK, 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 URL de endpoint padrão para o respectivo AWS service (Serviço da AWS) é usada por último.

Support by AWS SDKs and tools

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 sistema JVM são suportadas pelo AWS SDK para Java e pelo AWS SDK para Kotlin único.

SDK	Cl	Notas ou mais informações
AWS CLI v2	Sim	

SDK	Cl	Notas ou mais informações
SDK para C++	Sim	
SDK para Go V2 (1.x)	Sim	
SDK para Go 1.x (V1)	Não	
SDK para Java 2.x	Sim	
SDK para Java 1.x	Não	
SDK para 3.x JavaScript	Sim	
SDK para 2.x JavaScript	Não	
SDK para Kotlin	Sim	
SDK para .NET 4.x	Sim	
SDK para .NET 3.x	Sim	
SDK para PHP 3.x	Sim	
SDK para Python (Boto3)	Sim	
SDK para Ruby 3.x	Sim	
SDK para Rust	Sim	
SDK para Swift	Sim	
Ferramentas para PowerShell V5	Sim	
Ferramentas para PowerShell V4	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	AWS_ENDPOINT_URL_<SERVICE>	variável de ambiente
AccessAnalyzer	ac	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	af	AWS_ENDPOINT_URL_ALEXA_FOR_BUSINESS	
amp	ar	AWS_ENDPOINT_URL_AMP	
Amplify	ar	AWS_ENDPOINT_URL_AMPLIFY	
AmplifyBackend	ar	AWS_ENDPOINT_URL_AMPLIFYBACKEND	
AmplifyUIBuilder	ar	AWS_ENDPOINT_URL_AMPLIFYUIBUILDER	

serviceId	Cl	AWS_ENDPOINT_URL_<SERVICE>	variável de ambiente
API Gateway	ap	AWS_ENDPOINT_URL_API_GATEWAY	
ApiGatewayManagem entApi	ap	AWS_ENDPOINT_URL_APIGATEWAYMANAGEMENTAPI	
ApiGatewayV2	ap	AWS_ENDPOINT_URL_APIGATEWAYV2	
AppConfig	ap	AWS_ENDPOINT_URL_APPCONFIG	
AppConfigData	ap	AWS_ENDPOINT_URL_APPCONFIGDATA	
AppFabric	ap	AWS_ENDPOINT_URL_APPFABRIC	
Appflow	ap	AWS_ENDPOINT_URL_APPFLOW	
AppIntegrations	ap	AWS_ENDPOINT_URL_APPINTEGRATIONS	

serviceId	Cl	AWS_ENDPOINT_URL_<SERVICE>	variável de ambiente
Application Auto Scaling	a	AWS_ENDPOINT_URL_APPLICATION_AUTO_SCALING	
Application Insights	a	AWS_ENDPOINT_URL_APPLICATION_INSIGHTS	
ApplicationCostProfiler	a	AWS_ENDPOINT_URL_APPLICATIONCOSTPROFILER	
App Mesh	a	AWS_ENDPOINT_URL_APP_MESH	
AppRunner	a	AWS_ENDPOINT_URL_APPRUNNER	
AppStream	a	AWS_ENDPOINT_URL_APPSTREAM	
AppSync	a	AWS_ENDPOINT_URL_APPSYNC	
ARC Zonal Shift	a:	AWS_ENDPOINT_URL_ARC_ZONAL_SHIFT	
Artifact	a:	AWS_ENDPOINT_URL_ARTIFACT	

serviceId	Classe	Nome da variável de ambiente
	ClientConfiguration	AWS_ENDPOINT_URL_<SERVICE>
Athena	athena	AWS_ENDPOINT_URL_ATHENA
AuditManager	auditmanager	AWS_ENDPOINT_URL_AUDITMANAGER
Auto Scaling	autoscaling	AWS_ENDPOINT_URL_AUTO_SCALING
Auto Scaling Plans	autoscalingplans	AWS_ENDPOINT_URL_AUTO_SCALING_PLANS
b2bi	b2bi	AWS_ENDPOINT_URL_B2BI
Backup	backup	AWS_ENDPOINT_URL_BACKUP
Backup Gateway	backupgateway	AWS_ENDPOINT_URL_BACKUP_GATEWAY
BackupStorage	backupstorage	AWS_ENDPOINT_URL_BACKUPSTORAGE
Batch	batch	AWS_ENDPOINT_URL_BATCH
BCM Data Exports	bcmdataexports	AWS_ENDPOINT_URL_BCM_DATA_EXPORTS

serviceId	Cl	AWS_ENDPOINT_URL_<SERVICE>	variável de ambiente
Bedrock	b:	AWS_ENDPOINT_URL_BEDROCK	
Bedrock Agent	b:	AWS_ENDPOINT_URL_BEDROCK_AGENT	
Bedrock Agent Runtime	b:	AWS_ENDPOINT_URL_BEDROCK_AGENT_RUNTIME	
Bedrock Runtime	b:	AWS_ENDPOINT_URL_BEDROCK_RUNTIME	
billingconductor	b:	AWS_ENDPOINT_URL_BILLINGCONDUCTOR	
Braket	b:	AWS_ENDPOINT_URL_BRAKET	
Budgets	b:	AWS_ENDPOINT_URL_BUDGETS	
Cost Explorer	c:	AWS_ENDPOINT_URL_COST_EXPLORER	
chatbot	c:	AWS_ENDPOINT_URL_CHATBOT	
Chime	c:	AWS_ENDPOINT_URL_CHIME	

serviceId	Cl	AWS_ENDPOINT_URL_<SERVICE>	variável de ambiente
Chime SDK Identity	cl	AWS_ENDPOINT_URL_CHIME_SDK_IDENTITY	
Chime SDK Media Pipelines	cl	AWS_ENDPOINT_URL_CHIME_SDK_MEDIA_PIPELINES	
Chime SDK Meetings	cl	AWS_ENDPOINT_URL_CHIME_SDK_MEETINGS	
Chime SDK Messaging	cl	AWS_ENDPOINT_URL_CHIME_SDK_MESSAGING	
Chime SDK Voice	cl	AWS_ENDPOINT_URL_CHIME_SDK_VOICE	
CleanRooms	c:	AWS_ENDPOINT_URL_CLEANROOMS	
CleanRoomsML	c:	AWS_ENDPOINT_URL_CLEANROOMSML	
Cloud9	c:	AWS_ENDPOINT_URL_CLOUD9	

serviceId	AWS_ENDPOINT_URL_<SERVICE>	variável de ambiente
CloudControl	c: AWS_ENDPOINT_URL_CLOUDCONTROL	
CloudDirectory	c: AWS_ENDPOINT_URL_CLOUDDIRECTORY	
CloudFormation	c: AWS_ENDPOINT_URL_CLOUDFORMATION	
CloudFront	c: AWS_ENDPOINT_URL_CLOUDFRONT	
CloudFront KeyValueStore	c: AWS_ENDPOINT_URL_CLOUDFRONT_KEYVALUESTORE	
CloudHSM	c: AWS_ENDPOINT_URL_CLOUDHSM	
CloudHSM V2	c: AWS_ENDPOINT_URL_CLOUDHSM_V2	
CloudSearch	c: AWS_ENDPOINT_URL_CLOUDSEARCH	

serviceId	AWS_ENDPOINT_URL_<SERVICE>	variável de ambiente
CloudSearch Domain	c: AWS_ENDPOINT_URL_CLOUDSEARCH_DOMAIN	
CloudTrail	c: AWS_ENDPOINT_URL_CLOUDTRAIL	
CloudTrail Data	c: AWS_ENDPOINT_URL_CLOUDTRAIL_DATA	
CloudWatch	c: AWS_ENDPOINT_URL_CLOUDWATCH	
codeartifact	c: AWS_ENDPOINT_URL_CODEARTIFACT	
CodeBuild	c: AWS_ENDPOINT_URL_CODEBUILD	
CodeCatalyst	c: AWS_ENDPOINT_URL_CODECATALYST	
CodeCommit	c: AWS_ENDPOINT_URL_CODECOMMIT	

serviceId	Cl	AWS_ENDPOINT_URL_<SERVICE>	variável de ambiente
CodeDeploy	cd	AWS_ENDPOINT_URL_CODEDEPLOY	
CodeGuru Reviewer	cg	AWS_ENDPOINT_URL_CODEGURU_REVIEWER	
CodeGuru Security	cg	AWS_ENDPOINT_URL_CODEGURU_SECURITY	
CodeGuruProfiler	cg	AWS_ENDPOINT_URL_CODEGURUPROFILER	
CodePipeline	cp	AWS_ENDPOINT_URL_CODEPIPELINE	
CodeStar	cs	AWS_ENDPOINT_URL_CODESTAR	
CodeStar connections	cs	AWS_ENDPOINT_URL_CODESTAR_CONNECTIONS	
codestar notificat ions	cs	AWS_ENDPOINT_URL_CODESTAR_NOTIFICATIONS	

serviceId	Cl	AWS_ENDPOINT_URL_<SERVICE>	variável de ambiente
Cognito Identity	cl	AWS_ENDPOINT_URL_COGNITO_IDENTITY	
Cognito Identity Provider	cl	AWS_ENDPOINT_URL_COGNITO_IDENTITY_PROVIDER	
Cognito Sync	cl	AWS_ENDPOINT_URL_COGNITO_SYNC	
Comprehend	cl	AWS_ENDPOINT_URL_COMPREHEND	
ComprehendMedical	cl	AWS_ENDPOINT_URL_COMPREHENDMEDICAL	
Compute Optimizer	cl	AWS_ENDPOINT_URL_COMPUTE_OPTIMIZER	
Config Service	cl	AWS_ENDPOINT_URL_CONFIG_SERVICE	
Connect	cl	AWS_ENDPOINT_URL_CONNECT	

serviceId	Cl	AWS_ENDPOINT_URL_<SERVICE>	variável de ambiente
Connect Contact Lens	co	AWS_ENDPOINT_URL_CONNECT_CONTACT_LENS	
ConnectCampaigns	co	AWS_ENDPOINT_URL_CONNECTCAMPAIGNS	
ConnectCases	co	AWS_ENDPOINT_URL_CONNECTCASES	
ConnectParticipant	co	AWS_ENDPOINT_URL_CONNECTPARTICIPANT	
ControlTower	co	AWS_ENDPOINT_URL_CONTROLTOWER	
Cost Optimization Hub	co	AWS_ENDPOINT_URL_COST_OPTIMIZATION_HUB	
Cost and Usage Report Service	co	AWS_ENDPOINT_URL_COST_AND_USAGE_REPO	RT_SERVICE

serviceId	Cl	AWS_ENDPOINT_URL_<SERVICE>	variável de ambiente
Customer Profiles	cl	AWS_ENDPOINT_URL_CUSTOMER_PROFILES	
DataBrew	d	AWS_ENDPOINT_URL_DATABREW	
DataExchange	d	AWS_ENDPOINT_URL_DATAEXCHANGE	
Data Pipeline	d	AWS_ENDPOINT_URL_DATA_PIPELINE	
DataSync	d	AWS_ENDPOINT_URL_DATASYNC	
DataZone	d	AWS_ENDPOINT_URL_DATAZONE	
DAX	d	AWS_ENDPOINT_URL_DAX	
Detective	d	AWS_ENDPOINT_URL_DETECTIVE	
Device Farm	d	AWS_ENDPOINT_URL_DEVICE_FARM	
DevOps Guru	d	AWS_ENDPOINT_URL_DEVOPS_GURU	

serviceId	Cl	AWS_ENDPOINT_URL_<SERVICE>	variável de ambiente
Direct Connect	d:	AWS_ENDPOINT_URL_DIRECT_CONNECT	
Application Discovery Service	a:	AWS_ENDPOINT_URL_APPLICATION_DISCOVERY_SERVICE	
DLM	d:	AWS_ENDPOINT_URL_DLM	
Database Migration Service	d:	AWS_ENDPOINT_URL_DATABASE_MIGRATION_SERVICE	
DocDB	d:	AWS_ENDPOINT_URL_DOCDB	
DocDB Elastic	d:	AWS_ENDPOINT_URL_DOCDB_ELASTIC	
drs	d:	AWS_ENDPOINT_URL_DRS	
Directory Service	d:	AWS_ENDPOINT_URL_DIRECTORY_SERVICE	
DynamoDB	d:	AWS_ENDPOINT_URL_DYNAMODB	

serviceId	Cl	AWS_ENDPOINT_URL_<SERVICE>	variável de ambiente
DynamoDB Streams	dy	AWS_ENDPOINT_URL_DYNAMODB_STREAMS	
EBS	el	AWS_ENDPOINT_URL_EBS	
EC2	ec	AWS_ENDPOINT_URL_EC2	
EC2 Instance Connect	ec	AWS_ENDPOINT_URL_EC2_INSTANCE_CONNECT	
ECR	ec	AWS_ENDPOINT_URL_ECR	
ECR PUBLIC	ec	AWS_ENDPOINT_URL_ECR_PUBLIC	
ECS	ec	AWS_ENDPOINT_URL_ECS	
EFS	ef	AWS_ENDPOINT_URL_EFS	
EKS	el	AWS_ENDPOINT_URL_EKS	
EKS Auth	el	AWS_ENDPOINT_URL_EKS_AUTH	
Elastic Inference	ei	AWS_ENDPOINT_URL_ELASTIC_INFERENCE	

serviceId	Cl	AWS_ENDPOINT_URL_<SERVICE>	variável de ambiente
ElastiCache	e:	AWS_ENDPOINT_URL_ELASTICACHE	
Elastic Beanstalk	e:	AWS_ENDPOINT_URL_ELASTIC_BEANSTALK	
Elastic Transcoder	e:	AWS_ENDPOINT_URL_ELASTIC_TRANSCODER	
Elastic Load Balancing	e:	AWS_ENDPOINT_URL_ELASTIC_LOAD_BALANCING	
Elastic Load Balancing v2	e:	AWS_ENDPOINT_URL_ELASTIC_LOAD_BALANCING_V2	
EMR	er	AWS_ENDPOINT_URL_EMR	
EMR containers	er	AWS_ENDPOINT_URL_EMR_CONTAINERS	
EMR Serverless	er	AWS_ENDPOINT_URL_EMR_SERVERLESS	

serviceId	Cl	AWS_ENDPOINT_URL_<SERVICE>	variável de ambiente
EntityResolution	e:	AWS_ENDPOINT_URL_ENTITYRESOLUTION	
Elasticsearch Service	e:	AWS_ENDPOINT_URL_ELASTICSEARCH_SERVICE	
EventBridge	e:	AWS_ENDPOINT_URL_EVENTBRIDGE	
Evidently	e:	AWS_ENDPOINT_URL_EVIDENTLY	
finspace	f:	AWS_ENDPOINT_URL_FINSPACE	
finspace data	f:	AWS_ENDPOINT_URL_FINSPACE_DATA	
Firehose	f:	AWS_ENDPOINT_URL_FIREHOSE	
fis	f:	AWS_ENDPOINT_URL_FIS	
FMS	fr	AWS_ENDPOINT_URL_FMS	
forecast	fc	AWS_ENDPOINT_URL_FORECAST	

serviceId	Cl	AWS_ENDPOINT_URL_<SERVICE>	variável de ambiente
forecastquery	f:	AWS_ENDPOINT_URL_FORECASTQUERY	
FraudDetector	f:	AWS_ENDPOINT_URL_FRAUDETECTOR	
FreeTier	f:	AWS_ENDPOINT_URL_FREETIER	
FSx	f:	AWS_ENDPOINT_URL_FSX	
GameLift	g:	AWS_ENDPOINT_URL_GAMELIFT	
Glacier	g:	AWS_ENDPOINT_URL_GLACIER	
Global Accelerator	g:	AWS_ENDPOINT_URL_GLOBAL_ACCELERATOR	
Glue	g:	AWS_ENDPOINT_URL_GLUE	
grafana	g:	AWS_ENDPOINT_URL_GRAFANA	
Greengrass	g:	AWS_ENDPOINT_URL_GREENGRASS	
GreengrassV2	g:	AWS_ENDPOINT_URL_GREENGRASSV2	

serviceId	Cl	AWS_ENDPOINT_URL_<SERVICE>	variável de ambiente
GroundStation	g:	AWS_ENDPOINT_URL_GROUNDSTATION	
GuardDuty	g:	AWS_ENDPOINT_URL_GUARDDUTY	
Health	h:	AWS_ENDPOINT_URL_HEALTH	
HealthLake	h:	AWS_ENDPOINT_URL_HEALTHLAKE	
Honeycode	h:	AWS_ENDPOINT_URL_HONEYCODE	
IAM	i:	AWS_ENDPOINT_URL_IAM	
identitystore	i:	AWS_ENDPOINT_URL_IDENTITYSTORE	
imagebuilder	i:	AWS_ENDPOINT_URL_IMAGEBUILDER	
ImportExport	i:	AWS_ENDPOINT_URL_IMPORTEXPORT	
Inspector	i:	AWS_ENDPOINT_URL_INSPECTOR	

serviceId	Cl	AWS_ENDPOINT_URL_<SERVICE>	variável de ambiente
Inspector Scan	id	AWS_ENDPOINT_URL_INSPECTOR_SCAN	
Inspector2	id	AWS_ENDPOINT_URL_INSPECTOR2	
InternetMonitor	id	AWS_ENDPOINT_URL_INTERNETMONITOR	
IoT	id	AWS_ENDPOINT_URL_IOT	
IoT Data Plane	id	AWS_ENDPOINT_URL_IOT_DATA_PLANE	
IoT Jobs Data Plane	id	AWS_ENDPOINT_URL_IOT_JOBS_DATA_PLANE	
IoT 1Click Devices Service	id	AWS_ENDPOINT_URL_IOT_1CLICK_DEVICES_SERVICE	
IoT 1Click Projects	id	AWS_ENDPOINT_URL_IOT_1CLICK_PROJECTS	

serviceId	Cl	AWS_ENDPOINT_URL_<SERVICE>	variável de ambiente
	id		
	ac		
	de		
	se		
	pa		
	Al		
	co		
	ar		
	cc		
	ha		
IoTAnalytics	id	AWS_ENDPOINT_URL_IOTANALYTICS	
	id		
IotDeviceAdvisor	id	AWS_ENDPOINT_URL_IOTDEVICEADVISOR	
	ac		
IoT Events	id	AWS_ENDPOINT_URL_IOT_EVENTS	
	s		
IoT Events Data	id	AWS_ENDPOINT_URL_IOT_EVENTS_DATA	
	s_		
IoTFleetHub	id	AWS_ENDPOINT_URL_IOTFLEETHUB	
	ul		
IoTFleetWise	id	AWS_ENDPOINT_URL_IOTFLEETWISE	
	is		
IoTSecureTunneling	id	AWS_ENDPOINT_URL_IOTSECURETUNNELING	
	ti		
IoTSiteWise	id	AWS_ENDPOINT_URL_IOTSITWISE	
	se		

serviceId	Cl	AWS_ENDPOINT_URL_<SERVICE>	variável de ambiente
	id		
	ac		
	de		
	se		
	pa		
	Al		
	co		
	ar		
	cc		
	ha		
IoTThingsGraph	i	AWS_ENDPOINT_URL_IOTTHINGSGRAPH	
	g:		
IoTTwinMaker	i	AWS_ENDPOINT_URL_IOTTWINMAKER	
	k		
IoT Wireless	i	AWS_ENDPOINT_URL_IOT_WIRELESS	
	e:		
ivs	i	AWS_ENDPOINT_URL_IVS	
IVS RealTime	i	AWS_ENDPOINT_URL_IVS_REALTIME	
	ir		
ivschat	i	AWS_ENDPOINT_URL_IVSCHAT	
Kafka	k	AWS_ENDPOINT_URL_KAFKA	
KafkaConnect	k	AWS_ENDPOINT_URL_KAFKACONNECT	
	e:		
kendra	k	AWS_ENDPOINT_URL_KENDRA	
Kendra Ranking	k	AWS_ENDPOINT_URL_KENDRA_RANKING	
	nl		

serviceId	Environment Variable	Environment Variable Description
	<code>AWS_ENDPOINT_URL_<SERVICE></code>	variável de ambiente
Keyspaces	<code>k: AWS_ENDPOINT_URL_KEYSPACES</code>	
Kinesis	<code>k: AWS_ENDPOINT_URL_KINESIS</code>	
Kinesis Video Archived Media	<code>k: AWS_ENDPOINT_URL_KINESIS_VIDEO_ARCHIVED_MEDIA</code>	
Kinesis Video Media	<code>k: AWS_ENDPOINT_URL_KINESIS_VIDEO_MEDIA</code>	
Kinesis Video Signaling	<code>k: AWS_ENDPOINT_URL_KINESIS_VIDEO_SIGNALING</code>	
Kinesis Video WebRTC Storage	<code>k: AWS_ENDPOINT_URL_KINESIS_VIDEO_WEBRTC_STORAGE</code>	
Kinesis Analytics	<code>k: AWS_ENDPOINT_URL_KINESIS_ANALYTICS</code>	

serviceId	Cl	AWS_ENDPOINT_URL_<SERVICE>	variável de ambiente
Kinesis Analytics V2	k:	AWS_ENDPOINT_URL_KINESIS_ANALYTICS_V2	
Kinesis Video	k:	AWS_ENDPOINT_URL_KINESIS_VIDEO	
KMS	kr	AWS_ENDPOINT_URL_KMS	
LakeFormation	l:	AWS_ENDPOINT_URL_LAKEFORMATION	
Lambda	l:	AWS_ENDPOINT_URL_LAMBDA	
Launch Wizard	l:	AWS_ENDPOINT_URL_LAUNCH_WIZARD	
Lex Model Building Service	l:	AWS_ENDPOINT_URL_LEX_MODEL_BUILDING_SERVICE	
Lex Runtime Service	l:	AWS_ENDPOINT_URL_LEX_RUNTIME_SERVICE	

serviceId	Cl	AWS_ENDPOINT_URL_<SERVICE>	variável de ambiente
Lex Models V2	l	AWS_ENDPOINT_URL_LEX_MODELS_V2	
Lex Runtime V2	l	AWS_ENDPOINT_URL_LEX_RUNTIME_V2	
License Manager	l	AWS_ENDPOINT_URL_LICENSE_MANAGER	
License Manager Linux Subscriptions	l	AWS_ENDPOINT_URL_LICENSE_MANAGER_LINUX_SUBSCRIPTIONS	
License Manager User Subscriptions	l	AWS_ENDPOINT_URL_LICENSE_MANAGER_USER_SUBSCRIPTIONS	
Lightsail	l	AWS_ENDPOINT_URL_LIGHTSAIL	
Location	l	AWS_ENDPOINT_URL_LOCATION	
CloudWatch Logs	c	AWS_ENDPOINT_URL_CLOUDWATCH_LOGS	

serviceId	Cl	AWS_ENDPOINT_URL_<SERVICE>	variável de ambiente
LookoutEquipment	l	AWS_ENDPOINT_URL_LOOKOUTEQUIPMENT	
LookoutMetrics	l	AWS_ENDPOINT_URL_LOOKOUTMETRICS	
LookoutVision	l	AWS_ENDPOINT_URL_LOOKOUTVISION	
m2	m	AWS_ENDPOINT_URL_M2	
Machine Learning	m	AWS_ENDPOINT_URL_MACHINE_LEARNING	
Macie2	m	AWS_ENDPOINT_URL_MACIE2	
ManagedBlockchain	m	AWS_ENDPOINT_URL_MANAGEDBLOCKCHAIN	
ManagedBlockchain Query	m	AWS_ENDPOINT_URL_MANAGEDBLOCKCHAIN_QUERY	
Marketplace Agreement	m	AWS_ENDPOINT_URL_MARKETPLACE_AGREEMENT	

serviceId	Cl	AWS_ENDPOINT_URL_<SERVICE>	variável de ambiente
Marketplace Catalog	m	AWS_ENDPOINT_URL_MARKETPLACE_CATALOG	
Marketplace Deployment	m	AWS_ENDPOINT_URL_MARKETPLACE_DEPLOYMENT	
Marketplace Entitlement Service	m	AWS_ENDPOINT_URL_MARKETPLACE_ENTITLEMENT_SERVICE	
Marketplace Commerce Analytics	m	AWS_ENDPOINT_URL_MARKETPLACE_COMMERCE_ANALYTICS	
MediaConnect	m	AWS_ENDPOINT_URL_MEDIACONNECT	
MediaConvert	m	AWS_ENDPOINT_URL_MEDIACONVERT	
MediaLive	m	AWS_ENDPOINT_URL_MEDIALIVE	

serviceId	Cl	AWS_ENDPOINT_URL_<SERVICE>	variável de ambiente
MediaPackage	me	AWS_ENDPOINT_URL_MEDIAPACKAGE	
MediaPackage Vod	me	AWS_ENDPOINT_URL_MEDIAPACKAGE_VOD	
MediaPackageV2	me	AWS_ENDPOINT_URL_MEDIAPACKAGEV2	
MediaStore	me	AWS_ENDPOINT_URL_MEDIASTORE	
MediaStore Data	me	AWS_ENDPOINT_URL_MEDIASTORE_DATA	
MediaTailor	me	AWS_ENDPOINT_URL_MEDIATAILOR	
Medical Imaging	me	AWS_ENDPOINT_URL_MEDICAL_IMAGING	
MemoryDB	me	AWS_ENDPOINT_URL_MEMORYDB	
Marketplace Metering	me	AWS_ENDPOINT_URL_MARKETPLACE_METERING	

serviceId	Cl	AWS_ENDPOINT_URL_<SERVICE>	variável de ambiente
Migration Hub	id ac de se pa Al co ar cc ha	m: AWS_ENDPOINT_URL_MIGRATION_HUB	
mgn	m	AWS_ENDPOINT_URL_MGN	
Migration Hub Refactor Spaces	ct es	m: AWS_ENDPOINT_URL_MIGRATION_HUB_REFAC _I TOR_SPACES	
MigrationHub Config	ht g	m: AWS_ENDPOINT_URL_MIGRATIONHUB_CONFIG	
MigrationHubOrches trator	ht t:	m: AWS_ENDPOINT_URL_MIGRATIONHUBORCHESTRATOR	
MigrationHubStrategy	ht g)	m: AWS_ENDPOINT_URL_MIGRATIONHUBSTRATEGY	
Mobile	m	AWS_ENDPOINT_URL_MOBILE	
mq	m	AWS_ENDPOINT_URL_MQ	

serviceId	Cl	AWS_ENDPOINT_URL_<SERVICE>	variável de ambiente
	id		
	ac		
	de		
	se		
	pa		
	Al		
	co		
	ar		
	cc		
	ha		
MTurk	mt	AWS_ENDPOINT_URL_MTURK	
MWAA	mw	AWS_ENDPOINT_URL_MWAA	
Neptune	ne	AWS_ENDPOINT_URL_NEPTUNE	
Neptune Graph	ne	AWS_ENDPOINT_URL_NEPTUNE_GRAPH	
	ra		
neptunedata	ne	AWS_ENDPOINT_URL_NEPTUNEDATA	
	ta		
Network Firewall	ne	AWS_ENDPOINT_URL_NETWORK_FIREWALL	
	if		
NetworkManager	ne	AWS_ENDPOINT_URL_NETWORKMANAGER	
	nm		
NetworkMonitor	ne	AWS_ENDPOINT_URL_NETWORKMONITOR	
	nm		
nimble	nm	AWS_ENDPOINT_URL_NIMBLE	
OAM	oa	AWS_ENDPOINT_URL_OAM	
Omics	or	AWS_ENDPOINT_URL_OMICS	

serviceId	Cl	AWS_ENDPOINT_URL_<SERVICE>	variável de ambiente
OpenSearch	o:	AWS_ENDPOINT_URL_OPENSEARCH	
OpenSearchServerless	o:	AWS_ENDPOINT_URL_OPENSEARCHSERVERLESS	
OpsWorks	o:	AWS_ENDPOINT_URL_OPSWORKS	
OpsWorksCM	o:	AWS_ENDPOINT_URL_OPSWORKSCM	
Organizations	o:	AWS_ENDPOINT_URL_ORGANIZATIONS	
OSIS	o:	AWS_ENDPOINT_URL_OSIS	
Outposts	o:	AWS_ENDPOINT_URL_OUTPOSTS	
p8data	p:	AWS_ENDPOINT_URL_P8DATA	
p8data	p:	AWS_ENDPOINT_URL_P8DATA	
Panorama	p:	AWS_ENDPOINT_URL_PANORAMA	

serviceId	Cl id ac de se pa Al co ar cc ha	AWS_ENDPOINT_URL_<SERVICE> variável de ambiente
Payment Cryptography	p r h	AWS_ENDPOINT_URL_PAYMENT_CRYPTOGRAPHY
Payment Cryptography Data	p r h	AWS_ENDPOINT_URL_PAYMENT_CRYPTOGRAPHY_DATA
Pca Connector Ad	p c	AWS_ENDPOINT_URL_PCA_CONNECTOR_AD
Personalize	p z	AWS_ENDPOINT_URL_PERSONALIZE
Personalize Events	p z	AWS_ENDPOINT_URL_PERSONALIZE_EVENTS
Personalize Runtime	p z e	AWS_ENDPOINT_URL_PERSONALIZE_RUNTIME
PI	p	AWS_ENDPOINT_URL_PI
Pinpoint	p	AWS_ENDPOINT_URL_PINPOINT

serviceId	Cl	AWS_ENDPOINT_URL_<SERVICE>	variável de ambiente
Pinpoint Email	p:	AWS_ENDPOINT_URL_PINPOINT_EMAIL	
Pinpoint SMS Voice	p:	AWS_ENDPOINT_URL_PINPOINT_SMS_VOICE	
Pinpoint SMS Voice V2	p:	AWS_ENDPOINT_URL_PINPOINT_SMS_VOICE_V2	
Pipes	p:	AWS_ENDPOINT_URL_PIPES	
Polly	p:	AWS_ENDPOINT_URL_POLLY	
Pricing	p:	AWS_ENDPOINT_URL_PRICING	
PrivateNetworks	p:	AWS_ENDPOINT_URL_PRIVATENETWORKS	
Proton	p:	AWS_ENDPOINT_URL_PROTON	
QBusiness	q:	AWS_ENDPOINT_URL_QBUSINESS	
QConnect	q:	AWS_ENDPOINT_URL_QCONNECT	

serviceId	Cl	AWS_ENDPOINT_URL_<SERVICE>	variável de ambiente
QLDB	q	AWS_ENDPOINT_URL_QLDB	
QLDB Session	q	AWS_ENDPOINT_URL_QLDB_SESSION	
QuickSight	q	AWS_ENDPOINT_URL_QUICKSIGHT	
RAM	r	AWS_ENDPOINT_URL_RAM	
rbin	r	AWS_ENDPOINT_URL_RBIN	
RDS	r	AWS_ENDPOINT_URL_RDS	
RDS Data	r	AWS_ENDPOINT_URL_RDS_DATA	
Redshift	r	AWS_ENDPOINT_URL_REDSHIFT	
Redshift Data	r	AWS_ENDPOINT_URL_REDSHIFT_DATA	
Redshift Serverless	r	AWS_ENDPOINT_URL_REDSHIFT_SERVERLESS	
Rekognition	r	AWS_ENDPOINT_URL_REKOGNITION	

serviceId	Cl id ac de se pa Al co ar cc ha	AWS_ENDPOINT_URL_<SERVICE> variável de ambiente
repostspace	r(AWS_ENDPOINT_URL_REPOSTSPACE
resiliencehub	r(AWS_ENDPOINT_URL_RESILIENCEHUB
Resource Explorer 2	r(AWS_ENDPOINT_URL_RESOURCE_EXPLORER_2
Resource Groups	r(AWS_ENDPOINT_URL_RESOURCE_GROUPS
Resource Groups Tagging API	r(AWS_ENDPOINT_URL_RESOURCE_GROUPS_TAG g: GING_API g(
RoboMaker	r(AWS_ENDPOINT_URL_ROBOMAKER
RolesAnywhere	r(AWS_ENDPOINT_URL_ROLESEANYWHERE h(
Route 53	r(AWS_ENDPOINT_URL_ROUTE_53

serviceId	Cl	AWS_ENDPOINT_URL_<SERVICE>	variável de ambiente
Route53 Recovery Cluster	recovery-cluster	AWS_ENDPOINT_URL_ROUTE53_RECOVERY_CLUSTER	
Route53 Recovery Control Config	recovery-control-config	AWS_ENDPOINT_URL_ROUTE53_RECOVERY_CONTROL_CONFIG	
Route53 Recovery Readiness	recovery-readiness	AWS_ENDPOINT_URL_ROUTE53_RECOVERY_READINESS	
Route 53 Domains	domains	AWS_ENDPOINT_URL_ROUTE_53_DOMAINS	
Route53Resolver	resolver	AWS_ENDPOINT_URL_ROUTE53RESOLVER	
RUM	rum	AWS_ENDPOINT_URL_RUM	
S3	s3	AWS_ENDPOINT_URL_S3	
S3 Control	s3-control	AWS_ENDPOINT_URL_S3_CONTROL	

serviceId	Cl	AWS_ENDPOINT_URL_<SERVICE>	variável de ambiente
S3Outposts	s:	AWS_ENDPOINT_URL_S3OUTPOSTS	
SageMaker	s:	AWS_ENDPOINT_URL_SAGEMAKER	
SageMaker A2I Runtime	s:	AWS_ENDPOINT_URL_SAGEMAKER_A2I_RUNTIME	
Sagemaker Edge	s:	AWS_ENDPOINT_URL_SAGEMAKER_EDGE	
SageMaker FeatureStore Runtime	s:	AWS_ENDPOINT_URL_SAGEMAKER_FEATURESTORE_RUNTIME	
SageMaker Geospatial	s:	AWS_ENDPOINT_URL_SAGEMAKER_GEOSPATIAL	
SageMaker Metrics	s:	AWS_ENDPOINT_URL_SAGEMAKER_METRICS	

serviceId	Cl	AWS_ENDPOINT_URL_<SERVICE>	variável de ambiente
SageMaker Runtime	s:	AWS_ENDPOINT_URL_SAGEMAKER_RUNTIME	
savingsplans	s:	AWS_ENDPOINT_URL_SAVINGSPLANS	
Scheduler	s:	AWS_ENDPOINT_URL_SCHEDULER	
schemas	s:	AWS_ENDPOINT_URL_SCHEMAS	
SimpleDB	s:	AWS_ENDPOINT_URL_SIMPLEDB	
Secrets Manager	s:	AWS_ENDPOINT_URL_SECRETS_MANAGER	
SecurityHub	s:	AWS_ENDPOINT_URL_SECURITYHUB	
SecurityLake	s:	AWS_ENDPOINT_URL_SECURITYLAKE	
ServerlessApplicationRepository	s:	AWS_ENDPOINT_URL_SERVERLESSAPPLICATIONREPOSITORY	

serviceId	Cl id ac de se pa Al co ar cc ha	AWS_ENDPOINT_URL_<SERVICE> variável de ambiente
Service Quotas	s u	AWS_ENDPOINT_URL_SERVICE_QUOTAS
Service Catalog	s a	AWS_ENDPOINT_URL_SERVICE_CATALOG
Service Catalog AppRegistry	s a p	AWS_ENDPOINT_URL_SERVICE_CATALOG_APP REGISTRY
ServiceDiscovery	s s	AWS_ENDPOINT_URL_SERVICEDISCOVERY
SES	s	AWS_ENDPOINT_URL_SES
SESV2	s	AWS_ENDPOINT_URL_SESV2
Shield	s	AWS_ENDPOINT_URL_SHIELD
signer	s	AWS_ENDPOINT_URL_SIGNER
SimSpaceWeaver	s e	AWS_ENDPOINT_URL_SIMSPACEWEAVER
SMS	s	AWS_ENDPOINT_URL_SMS

serviceId	Cl	AWS_ENDPOINT_URL_<SERVICE>	variável de ambiente
Snow Device Management	si	AWS_ENDPOINT_URL_SNOW_DEVICE_MANAGEMENT	
Snowball	si	AWS_ENDPOINT_URL_SNOWBALL	
SNS	si	AWS_ENDPOINT_URL_SNS	
SQS	si	AWS_ENDPOINT_URL_SQS	
SSM	si	AWS_ENDPOINT_URL_SSM	
SSM Contacts	si	AWS_ENDPOINT_URL_SSM_CONTACTS	
SSM Incidents	si	AWS_ENDPOINT_URL_SSM_INCIDENTS	
Ssm Sap	si	AWS_ENDPOINT_URL_SSM_SAP	
SSO	si	AWS_ENDPOINT_URL_SSO	
SSO Admin	si	AWS_ENDPOINT_URL_SSO_ADMIN	
SSO OIDC	si	AWS_ENDPOINT_URL_SSO_OIDC	

serviceId	Ci id ac de se pa Al co ar cc ha	AWS_ENDPOINT_URL_<SERVICE>	variável de ambiente
SFN	sf	AWS_ENDPOINT_URL_SFN	
Storage Gateway	st at	AWS_ENDPOINT_URL_STORAGE_GATEWAY	
STS	st	AWS_ENDPOINT_URL_STS	
SupplyChain	sc ir	AWS_ENDPOINT_URL_SUPPLYCHAIN	
Support	su	AWS_ENDPOINT_URL_SUPPORT	
Support App	su pi	AWS_ENDPOINT_URL_SUPPORT_APP	
SWF	sw	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	

serviceId	Cl id ac de se pa Al co ar cc ha	AWS_ENDPOINT_URL_<SERVICE> variável de ambiente
Timestream Query	t:	AWS_ENDPOINT_URL_TIMESTREAM_QUERY
Timestream Write	t:	AWS_ENDPOINT_URL_TIMESTREAM_WRITE
tnb	tr	AWS_ENDPOINT_URL_TNB
Transcribe	t:	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	vi:	AWS_ENDPOINT_URL_VOICE_ID
VPC Lattice	vl: c:	AWS_ENDPOINT_URL_VPC_LATTICE

serviceId	Cl	AWS_ENDPOINT_URL_<SERVICE>	variável de ambiente
WAF	wi	AWS_ENDPOINT_URL_WAF	
WAF Regional	wi	AWS_ENDPOINT_URL_WAF_REGIONAL	
WAFV2	wi	AWS_ENDPOINT_URL_WAFV2	
WellArchitected	wi	AWS_ENDPOINT_URL_WELLARCHITECTED	
Wisdom	wi	AWS_ENDPOINT_URL_WISDOM	
WorkDocs	wi	AWS_ENDPOINT_URL_WORKDOCS	
WorkLink	wi	AWS_ENDPOINT_URL_WORKLINK	
WorkMail	wi	AWS_ENDPOINT_URL_WORKMAIL	
WorkMailMessageFlow	wi	AWS_ENDPOINT_URL_WORKMAILMESSAGEFLOW	
WorkSpaces	wi	AWS_ENDPOINT_URL_WORKSPACES	

serviceId	C	AWS_ENDPOINT_URL_<SERVICE>	variável de ambiente
WorkSpaces Thin Client	w	AWS_ENDPOINT_URL_WORKSPACES_THIN_CLIENT	
WorkSpaces Web	w	AWS_ENDPOINT_URL_WORKSPACES_WEB	
XRay	x:	AWS_ENDPOINT_URL_XRAY	

Padrões de configuração inteligente

Note

Para obter ajuda na compreensão do layout das páginas de configurações ou na interpretação da tabela Support by AWS SDKs and tools a seguir, consulte [Entender as páginas de configurações deste guia](#).

Com o recurso de padrões de configuração inteligente, AWS SDKs pode fornecer valores padrão predefinidos e otimizados para outras configurações.

Configure essa funcionalidade usando o seguinte:

defaults_mode- configuração de AWS **config** arquivo compartilhado, **AWS_DEFAULTS_MODE**: variável de ambiente, **aws.defaultsMode**- Propriedade do sistema JVM: 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 uma configuração do AWS SDK tiver um valor definido explicitamente, esse valor sempre terá precedência. Se uma configuração do AWS SDK não tiver um valor definido explicitamente e não `defaults_mode` for igual ao legado, esse recurso poderá fornecer valores padrão diferentes para várias configurações otimizadas para seu aplicativo. As configurações podem incluir o seguinte: configurações de comunicação HTTP, comportamento de repetição, configurações de endpoint regional do serviço e, potencialmente, qualquer configuração relacionada ao SDK. Os clientes que usam esse atributo podem obter novos padrões de configuração personalizados para cenários de uso comuns. Se seu `defaults_mode` não for igual a `legacy`, recomendamos realizar testes de seu aplicativo ao atualizar o SDK, 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 de acordo com o 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 ligam Serviços da AWS de dentro do mesmo Região da AWS.
- `cross-region`— Baseia-se no modo padrão e inclui otimização personalizada para aplicativos que fazem chamadas 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. O SDK tenta descobrir o ambiente de runtime 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 o SDK tenta novas tentativas. Consulte [Comportamento de repetição](#).
- `stsRegionalEndpoints`— Especifica como o SDK determina o AWS service (Serviço da AWS) endpoint que ele usa para se comunicar com o AWS Security Token Service ().AWS STS Consulte [AWS STS Endpoints regionais](#).
- `s3UsEast1RegionalEndpoints`— Especifica como o SDK determina o endpoint AWS 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 handshake TLS pode levar desde o momento em que a mensagem CLIENT HELLO 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
<code>retryMode</code>	<code>standard</code>	<code>standard</code>	<code>standard</code>	<code>standard</code>
<code>stsRegionalEndpoints</code>	<code>regional</code>	<code>regional</code>	<code>regional</code>	<code>regional</code>
<code>s3UsEast1RegionalEndpoints</code>	<code>regional</code>	<code>regional</code>	<code>regional</code>	<code>regional</code>

Parâmetro	Modo standard	Modo in-region	Modo cross-region	Modo mobile
<code>connectTimeoutInMillis</code>	3100	1100	3100	30000
<code>tlsNegotiationTimeoutInMillis</code>	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).

Support by AWS SDKs and tools

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 sistema JVM são suportadas pelo AWS SDK para Java e pelo AWS SDK para Kotlin único.

SDK	Compatível	Notas ou mais informações
AWS CLI v2	Não	
SDK para C++	Sim	Parâmetros não otimizado <code>s:stsRegionalEndpoints</code> , <code>s3UsEast1RegionalEndpoints</code> , <code>tlsNegotiationTimeoutInMillis</code> .
SDK para Go V2 (1.x)	Sim	Parâmetros não otimizado <code>s:retryMode</code> , <code>stsRegionalEndpoints</code> ,

SDK	Compatível	Notas ou mais informações
		<code>s3UsEast1RegionalEndpoints</code> .
SDK para Go 1.x (V1)	Não	
SDK para Java 2.x	Sim	Parâmetros não otimizados: <code>stsRegionalEndpoints</code> .
SDK para Java 1.x	Não	
SDK para 3.x JavaScript	Sim	Parâmetros não otimizados: <code>stsRegionalEndpoints</code> , <code>s3UsEast1RegionalEndpoints</code> , <code>tlsNegotiationTimeoutInMillis</code> . <code>connectTimeoutInMillis</code> é chamado <code>connectionTimeout</code> .
SDK para 2.x JavaScript	Não	
SDK para Kotlin	Não	
SDK para .NET 4.x	Sim	Parâmetros não otimizados: <code>connectTimeoutInMillis</code> , <code>tlsNegotiationTimeoutInMillis</code> .
SDK para .NET 3.x	Sim	Parâmetros não otimizados: <code>connectTimeoutInMillis</code> , <code>tlsNegotiationTimeoutInMillis</code> .

SDK	Compatível	Notas ou mais informações
SDK para PHP 3.x	Sim	Parâmetros não otimizado s: <code>tlsNegotiationTimeoutInMillis</code> .
SDK para Python (Boto3)	Sim	Parâmetros não otimizado s: <code>tlsNegotiationTimeoutInMillis</code> .
SDK para Ruby 3.x	Sim	
SDK para Rust	Não	
SDK para Swift	Não	
Ferramentas para PowerShell V5	Sim	Parâmetros não otimizado s: <code>connectTimeoutInMillis</code> , <code>tlsNegotiationTimeoutInMillis</code> .
Ferramentas para PowerShell V4	Sim	Parâmetros não otimizado s: <code>connectTimeoutInMillis</code> , <code>tlsNegotiationTimeoutInMillis</code> .

AWS Configurações de segurança de SDKs e ferramentas

Esta seção aborda algumas configurações de segurança comuns a todos os AWS SDKs. No entanto, para obter informações abrangentes sobre segurança para AWS SDKs individuais, consulte o capítulo dedicado à segurança no Guia do desenvolvedor do AWS SDK relevante:

- [AWS Capítulo sobre segurança do SDK for C++](#)
- [AWS Capítulo sobre segurança do SDK for Go](#)
- [AWS Capítulo sobre segurança do SDK for Java](#)
- [AWS Capítulo sobre SDK for JavaScript Security](#)
- [AWS Capítulo sobre segurança do SDK para Kotlin](#)
- [AWS Capítulo sobre segurança do SDK for .NET](#)
- [AWS Capítulo de segurança do SDK for PHP](#)
- [AWS Capítulo sobre segurança do SDK para Python](#)
- [AWS Capítulo sobre segurança do SDK for Ruby](#)
- [AWS Capítulo SDK para Rust Security](#)
- [AWS Capítulo SDK para Swift Security](#)

Tópicos

- [Habilitando o TLS híbrido pós-quântico](#)

Habilitando o TLS híbrido pós-quântico

AWS Os SDKs e as ferramentas têm recursos criptográficos e configurações que diferem de acordo com a linguagem e o tempo de execução. Atualmente, há três maneiras pelas quais um AWS SDK ou ferramenta fornece suporte ao PQ TLS:

Tópicos

- [SDKs com PQ TLS habilitado por padrão](#)
- [Opt-in Suporte para PQ TLS](#)
- [SDKs que dependem do System OpenSSL](#)
- [AWS SDKs e ferramentas que não planejam oferecer suporte ao PQ TLS](#)

SDKs com PQ TLS habilitado por padrão

Note

A partir de 6Nov-2025, o AWS SDK e suas bibliotecas CRT subjacentes para macOS e Windows usam bibliotecas do sistema para TLS, portanto, os recursos do PQ TLS nessas plataformas geralmente são determinados pelo suporte em nível de sistema.

AWS SDK para Go

O AWS SDK for Go usa a própria implementação de TLS da Golang fornecida por sua biblioteca padrão. O Golang suporta e prefere o PQ TLS a partir da versão 1.24, então os usuários do SDK for Go podem habilitar o PQ TLS AWS simplesmente atualizando o Golang para a versão 1.24

AWS SDK para JavaScript (navegador)

O AWS SDK para JavaScript (navegador) usa a pilha TLS do navegador, então o SDK negociará o PQ TLS se o tempo de execução do navegador o suportar e preferir. O Firefox lançou o suporte para PQ TLS na v132.0. O Chrome anunciou suporte para PQ TLS na v131. O Edge oferece suporte ao PQ TLS opcional na versão 120 para desktop e 140 para Android.

AWS SDK para Node.js

A partir das Node.js v22.20 (LTS) e v24.9.0, vincula e agrupa estaticamente Node.js o OpenSSL 3.5. Isso significa que o PQ TLS está habilitado e preferido por padrão para essas e versões subsequentes.

AWS SDK para Kotlin

O Kotlin SDK suporta e prefere o PQ TLS no Linux a partir da versão 1.5.78. Como o CRT-based cliente do AWS SDK for Kotlin depende de bibliotecas de sistema para TLS no macOS e no Windows, o suporte ao PQ TLS dependerá dessas bibliotecas subjacentes do sistema.

AWS SDK para Rust

O AWS SDK para Rust distribui pacotes distintos (conhecidos como “caixas” no ecossistema Rust) para cada cliente de serviço. Tudo isso é gerenciado em um GitHub repositório consolidado, mas cada cliente de serviço segue sua própria versão e cadência de lançamento. O SDK consolidado

lançou a preferência PQ TLS em 8/29 /25, portanto, qualquer versão individual do cliente de serviço lançada após essa data oferecerá suporte e preferirá PQ TLS por padrão.

Você pode determinar a versão mínima compatível com PQ TLS para um cliente de serviço específico navegando até o URL da versão relevante do crates.io (por exemplo, Crédito promocional da AWS está [aqui](#)) e encontrando a primeira versão publicada após 29- Aug-25 Qualquer versão do cliente de serviço publicada após 29- Aug-25 terá o PQ TLS ativado e preferido por padrão.

Opt-in Suporte para PQ TLS

AWS SDK para C++

Por padrão, o SDK para C++ usa clientes nativos da plataforma, como libcurl e WinHttp O Libcurl geralmente depende do OpenSSL do sistema para TLS, então o PQ TLS só é habilitado por padrão se o OpenSSL do sistema for \geq v3.5. Você pode substituir esse padrão no SDK para C++ v1.11.673 ou posterior e optar pelo `AwsCrHttpClient` que suporta e ativa o PQ TLS por padrão.

[Notas sobre a criação do Opt-In PQ TLS](#) Você pode buscar as dependências CRT do SDK com esse [script](#). A criação do SDK a partir do código-fonte é descrita [aqui](#) e [aqui](#), mas observe que talvez você precise de alguns sinalizadores adicionais do CMake:

```
-DUSE_CRT_HTTP_CLIENT=ON \  
-DUSE_TLS_V1_2=OFF \  
-DUSE_TLS_V1_3=ON \  
-DUSE_OPENSSL=OFF \  

```

AWS SDK para Java

A partir da v2, o AWS SDK for Java fornece AWS um cliente HTTP Common Runtime AWS (CRT) que pode ser configurado para executar PQ TLS. A partir da v2.35.11, ele `AwsCrHttpClient` ativa e prefere o PQ TLS por padrão, onde quer que seja usado.

SDKs que dependem do System OpenSSL

Vários AWS SDKs e ferramentas dependem da `libcrypto/libssl` biblioteca do sistema para TLS. A biblioteca do sistema mais usada é o OpenSSL. O OpenSSL habilitou o suporte ao PQ TLS na

versão 3.5, então a maneira mais fácil de configurar esses SDKs e ferramentas para o PQ TLS é usá-los em uma distribuição do sistema operacional que tenha pelo menos o OpenSSL 3.5 instalado.

Você também pode configurar um contêiner do Docker para usar o OpenSSL 3.5 para habilitar o PQ TLS em qualquer sistema que ofereça suporte ao Docker. Consulte Post-quantum TLS em Python para ver um exemplo de como configurar isso para Python.

AWS CLI

A partir da versão 2.34.54, o [AWS instalador de CLI para Linux](#) inclui o OpenSSL 3.5.6, portanto, o PQ TLS é habilitado e preferido por padrão para essa e versões subsequentes no Linux. AWS Os usuários da CLI no Linux podem habilitar o PQ TLS fazendo o upgrade para a AWS CLI v2.34.54 ou posterior.

Para macOS, instale a AWS CLI via [Homebrew](#) e certifique-se de que seu OpenSSL Homebrew-vended seja atualizado para a versão 3.5+. Você pode fazer isso com “brew install openssl @3.6” e validar com “brew list | grep openssl”.

Se você não conseguir atualizar para a AWS CLI v2.34.54 ou posterior no Ubuntu ou Debian Linux, certifique-se de que a distribuição Linux que você está usando tenha o OpenSSL 3.5+ instalado como sistema OpenSSL. [Em seguida, instale a AWS CLI usando apt ou PyPI](#). Com esses pré-requisitos, a AWS CLI vendida pelo apt ou PyPI será configurada para negociar. PQ-TLS [Para obter instruções passo a passo para validar a instalação, consulte o repositório github e a postagem do blog que a acompanha](#).

AWS SDK para PHP

O AWS SDK for PHP depende do sistema. libssl/libcrypto Para usar o PQ TLS, use esse SDK em uma distribuição de sistema operacional que tenha pelo menos o OpenSSL 3.5 instalado.

AWS SDK para Python (Boto3)

O AWS SDK para Python (Boto3) depende do sistema. libssl/libcrypto Para usar o PQ TLS, use esse SDK em uma distribuição de sistema operacional que tenha pelo menos o OpenSSL 3.5 instalado.

AWS SDK para Ruby

O AWS SDK para Ruby depende do sistema. libssl/libcrypto Para usar o PQ TLS, use esse SDK em uma distribuição de sistema operacional que tenha pelo menos o OpenSSL 3.5 instalado.

AWS SDK para .NET

No Linux, o AWS SDK for .NET depende do sistema. libssl/libcrypto Para usar o PQ TLS, use esse SDK em uma distribuição de sistema operacional que tenha pelo menos o OpenSSL 3.5 instalado. [No Windows e no macOS, o PQ TLS está disponível a partir do.NET 10 e no Windows 11. No macOS, o suporte ao TLS 1.3 \(um pré-requisito para o PQ TLS\) pode ser ativado ao optar pelo da Apple, conforme descrito aqui. Network.framework](#) Supondo uma versão mínima do.NET de 10, o PQ TLS deve então ser habilitado.

AWS SDKs e ferramentas que não planejam oferecer suporte ao PQ TLS

No momento, não há planos para oferecer suporte aos seguintes SDKs e ferramentas de linguagem:

- AWS SDK para SAP
- AWS SDK para Swift
- AWS Ferramentas para Windows PowerShell

AWS Bibliotecas do Common Runtime (CRT)

As bibliotecas do AWS Common Runtime (CRT) são uma biblioteca base do SDKs. O CRT é 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 (, SHA256 HMAC) MD5 SHA256, signatários, AES
- [awslabs/aws-c-common](#): estruturas de dados básicas, tipos threading/synchronization primitivos, 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 (RPC) em fluxos de eventos
- [awslabs/aws-c-http](#): implementação de C99 das especificações do HTTP/1.1 e do HTTP/2
- [awslabs/aws-c-io](#): soquetes (TCP, UDP), DNS, canais, circuitos de eventos, canais, SSL/TLS
- [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 BoringSSL e do projeto OpenSSL
- [awslabs/s2n](#): implementação C99 dos protocolos TLS/SSL, projetados para serem pequenos e rápidos, com a segurança como prioridade

O CRT está disponível em todos, SDKs exceto Go e Rust.

Adicionar dependências

As bibliotecas CRT formam uma rede complexa de relacionamentos e dependências. Conhecer essas relações é útil se você precisar criar o CRT diretamente da fonte. No entanto, a maioria dos usuários acessa a funcionalidade CRT por meio de seu SDK de linguagem (como AWS SDK para C++ ou SDK AWS para Java) ou do SDK de dispositivo de IoT de sua linguagem (como SDK de IoT AWS para C++ ou SDK de IoT para Java). AWS No diagrama a seguir, a caixa Associações de CRT do idioma se refere ao pacote que envolve as bibliotecas CRT para o SDK de um idioma específico. Essa é uma coleção de pacotes do formulário `aws-crt-*`, em que `*` é um idioma do SDK (como [aws-crt-cpp](#) ou [aws-crt-java](#)).

A seguir está uma ilustração das dependências hierárquicas das bibliotecas CRT.

Diagrama de dependência do CRT mostrando como as bibliotecas CRT individuais se relacionam entre si.

AWS SDKs e política de manutenção de ferramentas

Visão geral do

Este documento descreve a política de manutenção de kits (SDKs) e ferramentas de desenvolvimento de AWS software, incluindo dispositivos móveis e SDKs IoT, e suas dependências subjacentes. AWS fornece regularmente às Ferramentas AWS SDKs e às Ferramentas atualizações que podem conter suporte para recursos novos ou atualizados AWS APIs, novos recursos, aprimoramentos, correções de erros, 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 no. 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 continuado 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 pela AWS.

Ciclo de vida da versão principal do SDK

O ciclo de vida das versões principais SDKs e de ferramentas consiste em 5 fases, descritas abaixo.

- Developer Preview (Fase 0) - Durante essa fase, não SDKs são suportadas, não devem ser usadas em ambientes de produção e são destinadas 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 esta fase, 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, a 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.
- End-of-Support (Fase 4) - Quando um SDK atingir o 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.

A ilustração a seguir é uma ilustração visual do ciclo de vida da versão principal do SDK. Observe que os cronogramas mostrados abaixo são ilustrativos e não vinculativos.

Cronogramas da política de manutenção

Ciclo de vida de dependência

A maioria 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.
- 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 do AWS SDKs and Tools e onde elas estão em seu ciclo de vida de manutenção, consulte. [Ciclo de vida da versão](#)

AWS SDKs e ciclo de vida da versão Tools

A tabela abaixo mostra a lista das principais versões disponíveis do AWS 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 SDKs and Tools e suas dependências subjacentes, consulte. [Política de manutenção](#)

SDK	Versão principal	Fase atual	Data da disponibilidade geral	Observações
AWS CLI	1.x	Anúncio de manutenção	02/09/2013	Veja o anúncio para obter detalhes e datas
AWS CLI	2.x	Disponibilidade geral	2/10/2020	
SDK para C++	1.x	Disponibilidade geral	02/09/2015	
SDK para Go V2	V2 1.x	Disponibilidade geral	19/01/2021	
SDK para Go	1.x	Fim do suporte	19/11/2015	
SDK para Java	1.x	Fim do suporte	25/03/2010	
SDK para Java	2.x	Disponibilidade geral	20/11/2018	
SDK para JavaScript	1.x	Fim do suporte	6/5/2013	
SDK para JavaScript	2.x	Fim do suporte	19/06/2014	

SDK	Versão principal	Fase atual	Data da disponibilidade geral	Observações
SDK para JavaScript	3.x	Disponibilidade geral	15/12/2020	
SDK para Kotlin	1.x	Disponibilidade geral	27/11/2023	
SDK para .NET	1.x	Fim do suporte	11/2009	
SDK para .NET	2.x	Fim do suporte	08/11/2013	
SDK para .NET	3.x	Disponibilidade geral	28/07/2015	
SDK para .NET	4.x	Disponibilidade geral	28/04/2025	
SDK para PHP	2.x	Fim do suporte	11/2/2012	
SDK para PHP	3.x	Disponibilidade geral	27/05/2015	
SDK para Python (Boto2)	1.x	Fim do suporte	13/07/2011	
SDK para Python (Boto3)	1.x	Disponibilidade geral	22/06/2015	
SDK para Python (Botocore)	1.x	Disponibilidade geral	22/06/2015	
SDK para Ruby	1.x	Fim do suporte	14/07/2011	
SDK para Ruby	2.x	Fim do suporte	15/02/2015	

SDK	Versão principal	Fase atual	Data da disponibilidade geral	Observações
SDK para Ruby	3.x	Disponibilidade geral	29/08/2017	
SDK para Rust	1.x	Disponibilidade geral	27/11/2023	
SDK para 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	
Ferramentas para PowerShell	5.x	Disponibilidade geral	23/06/2025	

Procurando um SDK ou ferramenta não mencionada? Criptografia SDKs, dispositivo SDKs de IoT e dispositivos móveis SDKs, por exemplo, não estão incluídos neste guia. Para encontrar a documentação sobre essas outras ferramentas, consulte [Tools to Build on AWS](#).

Histórico de documentos AWS SDKs e guia de referência de ferramentas

A tabela a seguir descreve adições e atualizações importantes no Guia de referência de ferramentas AWS SDKs e ferramentas. Para receber notificações sobre atualizações dessa documentação, você pode se inscrever em o feed RSS.

Alteração	Descrição	Data
Adição de nova configuração do S3 Express One Zone	Adição de nova configuração do S3 Express One Zone para desabilitar autenticação de sessão.	13 de outubro de 2025
Adição de nova árvore decisória de autenticação	Adição de nova árvore decisória para auxiliar em decisões de autenticação entre opções.	23 de setembro de 2025
Adição de novo atributo de esquema de autenticação	Adição de novo atributo de esquema de autenticação. Atualizações nos endpoints AWS STS regionais.	18 de agosto de 2025
Adicionando uma nova versão do Tools for PowerShell	Adicionando a versão mais recente do Tools for PowerShell support a todas as referências de configuração Compatibilidade com AWS SDKs tabelas. Adicionado atributo de injeção de prefixo de host.	23 de junho de 2025
Atualizações de títulos de páginas	Mais títulos, títulos de tabelas, resumos e atualizações de SEO.	05 de março de 2025

Atualizações de títulos de páginas	Atualização de conteúdo para usar títulos mais descritivos.	24 de fevereiro de 2025
Adição de SDK do Swift à Referência de configurações	Adicionando suporte ao Swift SDK a todas as referências de configuração Compatibilidade com AWS SDKs tabelas.	17 de setembro de 2024
Propriedades de sistema do SDK para Java 1.x	Adicione detalhes sobre as configurações do sistema JVM suportadas pela versão 1.x. AWS SDK para Java	30 de maio de 2024
Atualizações de configurações	Adicionadas configurações de sistema de JVM.	27 de março de 2024
Atualizações da tabela de compatibilidade	Atualizações para compatibilidade com SDKs, atualizações de procedimentos do IAM Identity Center.	20 de fevereiro de 2024
Atualização da credencial do contêiner. Atualização do IMDS.	Adicionando suporte para o Amazon EKS. 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 para SDK e recursos de ferramentas atualizados para incluir SDK para Kotlin, SDK para Rust e Ferramentas da AWS para PowerShell.	10 de dezembro de 2023

Atualizações de autenticação	Atualizações dos métodos de autenticação SDKs e ferramentas compatíveis.	1º de julho de 2023
Atualizações de práticas recomendadas do IAM	Guia atualizado para alinhamento com as práticas recomendadas do IAM. Para saber mais, consulte Práticas recomendadas de segurança no IAM .	27 de fevereiro de 2023
Atualizações em SSO	Atualizações nas credenciais de SSO para a nova configuração do token SSO.	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 maior clareza do cliente IMDS e das credenciais do IMDS. 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 que o AWS SSO agora é chamado de Centro de Identidade do AWS IAM.	26 de julho de 2022
Atualização de configurações	Pequenas atualizações nos detalhes do arquivo de configuração e nas configurações 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

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.