Guía de referencia

AWS SDKsy herramientas

AWS SDKsy herramientas: Guía de referencia

Table of Contents

AWS SDKsGuía de referencia de herramientas y herramientas

Muchas SDKs herramientas comparten alguna funcionalidad común, ya sea a través de especificaciones de diseño compartidas o de una biblioteca compartida.

Esta guía incluye información sobre:

- Configuración— Cómo utilizar los `credentials` archivos `config` y variables de entorno compartidos para configurar sus AWS SDKs propias herramientas.
- Autenticación y acceso— Establece cómo se autentica tu código o herramienta AWS cuando desarrollas con Servicios de AWS ellos.
- Referencia de configuración: referencia para todos los ajustes estandarizados disponibles para la autenticación y la configuración.
- Bibliotecas de Common Runtime (CRT) AWS— Descripción general de las bibliotecas compartidas de AWS Common Runtime (CRT) que están disponibles para casi todosSDKs.
- AWS Política de mantenimiento de SDK y herramientascubre la política de mantenimiento y el control de versiones de los kits y herramientas de desarrollo de AWS software (SDKs), incluidos los dispositivos móviles y el Internet de las cosas (IoT)SDKs, y sus dependencias subyacentes.

Esta guía de referencia AWS SDKs y las herramientas pretenden ser una base de información aplicable a múltiples SDKs herramientas. La guía específica de la herramienta SDK o herramienta que esté utilizando debe utilizarse además de cualquier información que se presente aquí. Las siguientes son las herramientas SDK y herramientas que incluyen secciones de material relevantes en esta guía:

Si utiliza:	Las secciones relevantes de esta guía para usted son:
• ¿Alguna herramienta SDK o herramienta	AWS Política de mantenimiento de SDK y herramientas
• AWS Cloud9 • AWS CDK • AWS Toolkit for Azure DevOps	Configuración Autenticación y acceso

Si utiliza:	Las secciones relevantes de esta guía para usted son:
• 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	AWS Política de mantenimiento de SDK y herramientas
• AWS CLI • AWS SDK for C++ • AWS SDK for Go • AWS SDK for Java • AWS SDK for JavaScript • AWS SDK para Kotlin • AWS SDK for .NET • AWS SDK for PHP • AWS SDK for Python (Boto3) • AWS SDK for Ruby • AWS SDK para Rust • AWS SDK para Swift • AWS Tools for Windows PowerShell	Configuración Autenticación y acceso Referencia de configuración Bibliotecas de Common Runtime (CRT) AWS AWS Política de mantenimiento de SDK y herramientas AWS SDKsy compatibilidad con las versiones de Tools

Recursos para desarrolladores

Para obtener una descripción general de las herramientas que pueden ayudarle a desarrollar aplicaciones AWS, consulte Herramientas sobre las que construir AWS. Para obtener más información sobre el soporte, consulte el Centro de conocimiento de AWS.

Amazon Q Developer es un asistente conversacional generativo basado en inteligencia artificial que puede ayudarlo a comprender, crear, ampliar y operar aplicaciones. AWS Para acelerar su desarrollo AWS, el modelo que impulsa Amazon Q se complementa con AWS contenido de alta calidad para producir respuestas más completas, procesables y referenciadas. Para obtener más información, consulta ¿Qué es Amazon Q Developer? en la Guía del usuario para desarrolladores de Amazon Q.

Notificación de telemetría del kit de herramientas

AWS Los kits de herramientas del entorno de desarrollo integrado (IDE) son complementos y extensiones que permiten el acceso a AWS los servicios desde su. IDE Para obtener información detallada sobre cada uno de los IDE kits de herramientas, consulte las guías de usuario del kit de herramientas de la tabla anterior.

AWS IDELos kits de herramientas pueden recopilar y almacenar datos de telemetría del lado del cliente para informar las decisiones sobre futuras versiones de Toolkit. AWS Los datos recopilados cuantifican su uso del kit de herramientas. AWS

Para obtener más información sobre los datos de telemetría recopilados en todos los AWS IDE kits de herramientas, consulta el commonDefinitionsdocumento.json en el repositorio de Github. `aws-toolkit-common`

Para obtener información detallada sobre los datos de telemetría recopilados por cada uno de los AWS IDE kits de herramientas, consulta los documentos de recursos de los siguientes repositorios de Github: AWS

- AWS Toolkit for Visual Studio
- AWS Toolkit for Visual Studio Code
- AWS Toolkit for JetBrains

Algunos AWS servicios a los que se puede acceder en los AWS kits de herramientas pueden recopilar datos de telemetría adicionales del lado del cliente. Para obtener información detallada

sobre el tipo de datos que recopila cada AWS servicio individual, consulte el tema de la AWS documentación correspondiente al servicio específico que le interese.

Configuración

Con AWS los SDK y otras herramientas para AWS desarrolladores, como AWS Command Line Interface (AWS CLI), puedes interactuar con las API de AWS servicio. Sin embargo, antes de intentarlo, debes configurar el SDK o la herramienta con la información necesaria para realizar la operación solicitada.

La información incluye los siguientes elementos:

- Información de credenciales que identifica quién llama a la API. Las credenciales se utilizan para cifrar la solicitud a los AWS servidores. Con esta información, AWS confirma su identidad y puede recuperar las políticas de permisos asociadas a la misma. Luego, puede determinar qué acciones puedes realizar.
- Otros detalles de configuración que se utilizan para indicar al SDK AWS CLI o al software cómo procesar la solicitud, dónde enviarla (a qué punto final del AWS servicio) y cómo interpretar o mostrar la respuesta.

Cada SDK o herramienta admite varias fuentes que puede utilizar para proporcionar las credenciales y la información de configuración necesarias. Algunas fuentes son exclusivas del SDK o la herramienta, y debes consultar la documentación de esa herramienta o SDK para obtener más información sobre cómo usar ese método.

Sin embargo, la mayoría de AWS los SDK y las herramientas admiten configuraciones comunes procedentes de dos fuentes principales (además del propio código):

- Archivos de AWS configuración y credenciales compartidos: los `credentials` archivos `config` AND compartidos son la forma más común de especificar la autenticación y la configuración de un AWS SDK o una herramienta. Usa estos archivos para almacenar la configuración que pueden usar tus herramientas y aplicaciones. Los ajustes de los archivos compartidos `config` y `credentials` están asociados a un perfil específico. Con varios perfiles, puede crear diferentes opciones de configuración para aplicarlas en diferentes escenarios. Cuando utilizas una AWS herramienta para invocar un comando o un SDK para invocar una AWS API, puedes especificar qué perfil y, por lo tanto, qué ajustes de configuración quieres usar para esa acción. Uno de los perfiles se denomina perfil `default` y se utiliza automáticamente cuando no especifica explícitamente un perfil que se va a utilizar. La configuración que puede almacenar en estos archivos se documenta en esta guía de referencia.

- Variables de entorno: algunas de las configuraciones también se pueden almacenar en las variables de entorno del sistema operativo. Aunque solo puede tener un conjunto de variables de entorno en vigor a la vez, se modifican fácilmente de forma dinámica a medida que se ejecuta el programa y cambian sus requisitos.

Temas adicionales en esta sección

- Archivos config y credentials compartidos
- Ubicación de los archivos config y credentials compartidos
- Compatibilidad con variables de entorno
- Soporte de propiedades del sistema JVM

Archivos **`config`** y **`credentials`** compartidos

Lo compartido AWS `config`y `credentials` los archivos contienen un conjunto de perfiles. Un perfil es un conjunto de valores de configuración, en pares clave-valor, que utiliza el AWS Command Line Interface (AWS CLI), el AWS SDKs, y otras herramientas. Los valores de configuración se adjuntan a un perfil para configurar algún aspecto de la SDK /tool cuando se utiliza ese perfil. Estos archivos se «comparten», ya que los valores se aplican a cualquier aplicación o proceso o SDKs al entorno local del usuario.

Tanto los archivos compartidos `config` como `credentials` los archivos son archivos de texto sin formato que contienen solo ASCII caracteres (codificados con el UTF código -8). Adoptan la forma de lo que generalmente se denomina INIarchivos.

Perfiles

Los ajustes de los archivos compartidos `config` y `credentials` están asociados a un perfil específico. Se pueden definir varios perfiles en el archivo para crear diferentes configuraciones de configuración que se puedan aplicar en diferentes entornos de desarrollo.

El `[default]` perfil contiene los valores que utiliza una operación de una SDK herramienta si no se especifica un perfil con nombre específico. También puede crear perfiles independientes a los que pueda hacer referencia de forma explícita por su nombre. Cada perfil puede usar diferentes configuraciones y valores según lo necesite la aplicación y el escenario.

> **Note**
>
> `[default]` es simplemente un perfil sin nombre. Este perfil recibe su nombre `default` porque es el perfil predeterminado que utiliza SDK si el usuario no especifica ningún perfil. No proporciona valores predeterminados heredados a otros perfiles. Si establece algo en el `[default]` perfil y no lo establece en un perfil con nombre, el valor no se establece cuando usa el perfil con nombre.

Establece un perfil con nombre

El `[default]` perfil y varios perfiles con nombre pueden existir en el mismo archivo. Usa la siguiente configuración para seleccionar qué configuración de perfil usará tu herramienta SDK o tu herramienta al ejecutar el código. Los perfiles también se pueden seleccionar dentro del código o mediante un comando cuando se trabaja con AWS CLI.

Configure esta funcionalidad mediante una de las siguientes opciones:

`AWS_PROFILE`- variable de entorno

Cuando esta variable de entorno se establece en un perfil con nombre o «predeterminado», todo el SDK código y AWS CLI los comandos utilizan la configuración de ese perfil.

Ejemplo de configuración de variables de entorno en Linux/macOS mediante la línea de comandos:

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

Ejemplo de configuración de variables de entorno en Windows mediante la línea de comandos:

```
setx AWS_PROFILE "my_default_profile_name"
```

`aws.profile`- propiedad JVM del sistema

SDKPara Kotlin on the JVM y SDK para Java 2.x, puede establecer la propiedad del `aws.profile` sistema. Cuando SDK crea un cliente de servicio, utiliza la configuración del perfil indicado, a menos que la configuración se anule en el código. La versión 1.x SDK para Java no admite esta propiedad del sistema.

Formato del archivo de configuración

El archivo `config` está organizado en secciones. Una sección es una colección con nombre de configuraciones y continúa hasta que se encuentra otra línea de definición de sección.

El archivo `config` es un archivo de texto sin formato que utiliza el formato siguiente:

- Todas las entradas de una sección adoptan el formato general de `setting-name=value`.
- Las líneas se pueden comentar si se inician con un carácter de almohadilla (#).

Tipo de sección

La definición de una sección es una línea que aplica un nombre a un conjunto de ajustes. Las líneas de definición de sección comienzan y terminan con corchetes (`[ ]`). Dentro de los corchetes, hay un identificador de tipo de sección y un nombre personalizado para la sección. Puede utilizar letras, números, guiones (`-`) y guiones bajos (_), pero no espacios.

Tipo de sección: **`default`**

Ejemplo de línea de definición de sección: `[default]`

`[default]`es el único perfil que no requiere el identificador de `profile` sección.

En el siguiente ejemplo, se muestra un archivo `config` con un perfil `[default]`. Establece la configuración `region`. Todos los ajustes que siguen esta línea, hasta que se encuentre otra definición de sección, forman parte de este perfil.

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

Tipo de sección: **`profile`**

Ejemplo de línea de definición de sección: `[profile dev]`

La línea de definición de la `profile` sección es una agrupación de configuraciones con nombre que se puede aplicar a distintos escenarios de desarrollo. Para conocer mejor los perfiles con nombre, consulte la sección anterior sobre Perfiles.

El siguiente ejemplo muestra un `config` archivo con una línea de definición de `profile` sección y un perfil con nombre denominado`foo`. Todos los ajustes que siguen esta línea, hasta que se encuentre otra definición de sección, forman parte de este perfil con nombre.

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

Algunas configuraciones tienen su propio grupo anidado de subconfiguraciones, como la configuración `s3` y las subconfiguraciones del siguiente ejemplo. Para asociar los subajustes al grupo, indéntelos con uno o más espacios.

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

Tipo de sección: **`sso-session`**

Ejemplo de línea de definición de sección: `[sso-session my-sso]`

La línea de definición de la `sso-session` sección nombra un grupo de ajustes que se utilizan para configurar un perfil para resolverlos AWS credenciales mediante AWS IAM Identity Center. Para obtener más información sobre la configuración de la autenticación de inicio de sesión único, consulte. Autenticación del Centro de identidades de IAM Un perfil está vinculado a una sección `sso-session` mediante un par clave-valor en el que `sso-session` es la clave y el nombre de la sección `sso-session` es el valor, como `sso-session = <name-of-sso-session-section>`.

El siguiente ejemplo configura un perfil que será de corta duración AWS credenciales para el IAM rol «SampleRole» en la cuenta «111122223333" utilizando un token de «my-sso». La sección “my-sso” `sso-session` se menciona en la sección `profile` por su nombre mediante la clave `sso-session`.

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

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

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

Tipo de sección: **`services`**

Ejemplo de línea de definición de sección: `[services dev]`

> **Note**
>
> La `services` sección admite personalizaciones de terminales específicas del servicio y solo está disponible en las herramientas que incluyen esta función. SDKs Para ver si esta función está disponible para sus puntos de conexión específicos del servicioSDK, consulte los puntos de conexión específicos del Compatibilidad con AWS SDKs servicio.

La línea de definición de la `services` sección indica un grupo de ajustes que configuran puntos de conexión personalizados para Servicio de AWS solicitudes. Un perfil está vinculado a una sección `services` mediante un par clave-valor en el que `services` es la clave y el nombre de la sección `services` es el valor, como `services = <name-of-services-section>`.

La `services` sección se divide a su vez en subsecciones por `<SERVICE> =` líneas, donde `<SERVICE>` está la Servicio de AWS clave de identificación. La Servicio de AWS el identificador se basa en el API modelo sustituyendo todos los espacios `serviceId` por guiones bajos y minúsculas todas las letras. Para obtener una lista de todas las claves de identificación de servicio que se van a utilizar en la sección de `services`, consulte Identificadores de punto de conexión específicos del servicio. La clave del identificador del servicio va seguida de configuraciones anidadas, cada una en su propia línea y marcada con dos espacios.

En el siguiente ejemplo, se utiliza una `services` definición para configurar el punto final que se utilizará únicamente en las solicitudes realizadas al Amazon DynamoDB servicio. La sección `"local-dynamodb"` de `services` se menciona en la sección `profile` por su nombre mediante la clave `services`. La Servicio de AWS la clave identificadora es`dynamodb`. La Amazon DynamoDB la subsección de servicio comienza en la línea`dynamodb =` . Todas las líneas inmediatamente siguientes que estén sangradas se incluyen en esa subsección y se aplican a ese servicio.

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

[services local-dynamodb]
dynamodb =
```

```
  endpoint_url = http://localhost:8000
```

Para obtener más información sobre la configuración de punto de conexión, consulte Puntos de conexión específicos del servicio.

Formato del archivo de credenciales

Por lo general, las reglas del archivo `credentials` son idénticas a las del archivo `config`, con la salvedad de que las secciones del perfil no comienzan por la palabra "`profile`". Utilice únicamente el propio nombre del perfil entre corchetes. En el siguiente ejemplo se muestra un `credentials` archivo con una sección de perfil con nombre denominada`foo`.

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

En el `credentials` archivo solo se pueden almacenar las siguientes configuraciones que se consideran «secretas» o confidenciales: `aws_access_key_id`, `aws_secret_access_key`, y`aws_session_token`. Si bien estas configuraciones también se pueden colocar en el `config` archivo compartido, le recomendamos que mantenga estos valores confidenciales en un `credentials` archivo independiente. De esta forma, puede proporcionar permisos independientes para cada archivo, si es necesario.

En el siguiente ejemplo, se muestra un archivo `credentials` con un perfil `[default]`. Establece la configuración global `aws_access_key_id`, `aws_secret_access_key`, y la configuración `aws_session_token` global.

```
[default]
aws_access_key_id=AKIAIOSFODNN7EXAMPLE
aws_secret_access_key=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
aws_session_token=IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZVERYLONGSTRI
```

Independientemente de si utiliza un perfil con nombre o «`default`» en el `credentials` archivo, cualquier configuración que aparezca aquí se combinará con la configuración del `config` archivo que utilice el mismo nombre de perfil. Si hay credenciales en ambos archivos para un perfil que comparte el mismo nombre, las claves del archivo de credenciales tienen prioridad.

Ubicación de los archivos **config** y **credentials** compartidos

Los `credentials` archivos AWS `config` y compartidos son archivos de texto sin formato que contienen información de configuración de los AWS SDK y las herramientas. Los archivos residen localmente en su entorno y el código del SDK o los AWS CLI comandos que ejecuta en ese entorno los utilizan automáticamente. Por ejemplo, en tu propio ordenador o al desarrollar en una instancia de Amazon Elastic Compute Cloud.

Cuando se ejecuta el SDK o la herramienta, comprueba estos archivos y carga todos los ajustes de configuración disponibles. Si los archivos aún no existen, el SDK o la herramienta crean automáticamente un archivo básico.

De forma predeterminada, los archivos se encuentran en una carpeta con el nombre `.aws` que se encuentra en su carpeta home o en la de usuario.

Sistema operativo	Ubicación y nombre predeterminados de los archivos
Linux y macOS	`~/.aws/config` `~/.aws/credentials`
Windows	`%USERPROFILE%\.aws\config` `%USERPROFILE%\.aws\credentials`

Resolución del directorio principal

~solo se utiliza para la resolución del directorio principal cuando:

- Inicia la ruta
- Va seguido inmediatamente por un separador específico de la plataforma `/` o por uno específico. En Windows, `~/` `~\` ambos se resuelven en el directorio principal.

Al determinar el directorio principal, se comprueban las siguientes variables:

- (Todas las plataformas) La variable de entorno `HOME`
- (Plataformas Windows) La variable de entorno `USERPROFILE`

- (Plataformas Windows) La concatenación de las variables de `HOMEDRIVE` `HOMEPATH` entorno () `$HOMEDRIVE$HOMEPATH`
- (Opcional según el SDK o la herramienta) Una función o variable de resolución de la ruta de inicio específica del SDK o de la herramienta

Cuando sea posible, si el directorio principal de un usuario se especifica al principio de la ruta (por ejemplo, `~username/`), se resuelve en el directorio principal del nombre de usuario solicitado (por ejemplo, `/home/username/.aws/config`).

Cambie la ubicación predeterminada de estos archivos

Puedes usar cualquiera de las siguientes opciones para anular el lugar desde el que el SDK o la herramienta cargan estos archivos.

Utilización de variables de entorno

Se pueden configurar las siguientes variables de entorno para cambiar la ubicación o el nombre de estos archivos del valor predeterminado a un valor personalizado:

- Variable de entorno de archivo `config`: **`AWS_CONFIG_FILE`**
- Variable de entorno de archivo `credentials`: **`AWS_SHARED_CREDENTIALS_FILE`**

Linux/macOS

Puede especificar una ubicación alternativa ejecutando los siguientes comandos de export en Linux o 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

Puede especificar una ubicación alternativa ejecutando los siguientes comandos de setx en 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 obtener más información sobre la configuración del sistema mediante variables de entorno, consulteCompatibilidad con variables de entorno.

Utilice las propiedades del sistema JVM

Para el SDK para Kotlin que se ejecuta en la JVM y para el SDK para Java 2.x, puedes configurar las siguientes propiedades del sistema JVM para cambiar la ubicación o el nombre de estos archivos del valor predeterminado a uno personalizado:

- `config`propiedad del sistema JVM del archivo: **`aws.configFile`**
- Variable de entorno de archivo `credentials`: **`aws.sharedCredentialsFile`**

Para obtener instrucciones sobre cómo configurar las propiedades del sistema JVM, consulte. the section called “¿Cómo configurar las propiedades del sistema JVM” El SDK for Java 1.x no admite estas propiedades del sistema.

Compatibilidad con variables de entorno

Las variables de entorno constituyen otro mecanismo para especificar opciones de configuración y credenciales, y pueden ser útiles para crear scripts o configurar temporalmente un perfil con nombre como la opción predeterminada. Para ver la lista de variables de entorno compatibles con la mayoríaSDKs, consulteLista de variables de entorno.

Prioridad de las opciones

- Si especifica una configuración mediante su variable de entorno, anulará cualquier valor cargado desde un perfil en el entorno compartido AWS `config`y `credentials` archivos.
- Si especifica un ajuste mediante un parámetro del AWS CLI en la línea de comandos, anula cualquier valor de la variable de entorno correspondiente o de un perfil del archivo de configuración.

Cómo configurar las variables de entorno

En los siguientes ejemplos se muestra cómo se pueden configurar las variables de entorno para el usuario predeterminado.

Linux, macOS, or Unix

```
$ export AWS_ACCESS_KEY_ID=AKIAIOSFODNN7EXAMPLE
$ export AWS_SECRET_ACCESS_KEY=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
$ export
 AWS_SESSION_TOKEN=AQoEXAMPLEH4aoAH0gNCAPy...truncated...zrkuWJOgQs8IZZaIv2BXIa2R4Olgk
$ export AWS_REGION=us-west-2
```

La configuración de la variable de entorno cambia el valor usado hasta el final de su sesión del intérprete de comandos o hasta que otorgue a la variable un valor diferente. Puede hacer que las variables persistan en sesiones futuras configurándolas en el script de startup del intérprete de comandos.

Windows Command Prompt

```
C:\> setx AWS_ACCESS_KEY_ID AKIAIOSFODNN7EXAMPLE
C:\> setx AWS_SECRET_ACCESS_KEY wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
C:\> setx
 AWS_SESSION_TOKEN AQoEXAMPLEH4aoAH0gNCAPy...truncated...zrkuWJOgQs8IZZaIv2BXIa2R4Olgk
C:\> setx AWS_REGION us-west-2
```

El uso de `set` para configurar una variable de entorno cambia el valor usado hasta que finalice la sesión de símbolo del sistema actual o hasta que otorgue a la variable un valor diferente. El uso de `setx` para establecer una variable de entorno cambia el valor usado en la sesión de símbolo del sistema actual y en todas las sesiones de símbolo del sistema que cree después de ejecutar el comando. La operación no afecta a otros comandos del shell que ya se están ejecutando en el momento de ejecutar el comando.

PowerShell

```
PS C:\> $Env:AWS_ACCESS_KEY_ID="AKIAIOSFODNN7EXAMPLE"
PS C:\> $Env:AWS_SECRET_ACCESS_KEY="wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY"
PS C:
\> $Env:AWS_SESSION_TOKEN="AQoEXAMPLEH4aoAH0gNCAPy...truncated...zrkuWJOgQs8IZZaIv2BXIa2R4Ol
PS C:\> $Env:AWS_REGION="us-west-2"
```

Si establece una variable de entorno en la PowerShell línea de comandos, como se muestra en los ejemplos anteriores, guardará el valor únicamente durante la sesión actual. Para que la configuración de la variable de entorno sea persistente en todas las sesiones PowerShell y en las de Command Prompt, guárdela mediante la aplicación Sistema del Panel de control. Como

alternativa, puede configurar la variable para todas las PowerShell sesiones futuras añadiéndola a su PowerShell perfil. Consulte la PowerShell documentación para obtener más información sobre cómo almacenar variables de entorno o cómo conservarlas en todas las sesiones.

Configuración de variables de entorno sin servidor

Si utiliza una arquitectura sin servidor para el desarrollo, tiene otras opciones para configurar las variables de entorno. En función del contenedor, puede usar diferentes estrategias para que el código que se ejecute en esos contenedores pueda ver las variables de entorno y acceder a ellas, de forma similar a lo que ocurre en los entornos que no son de nube.

Por ejemplo, con AWS Lambda, puede configurar directamente las variables de entorno. Para obtener más información, consulte Uso AWS Lambda variables de entorno en AWS Lambda Guía para desarrolladores.

En Serverless Framework, a menudo puede configurar variables de SDK entorno en el `serverless.yml` archivo con la clave del proveedor en la configuración del entorno. Para obtener información sobre el archivo `serverless.yml`, consulte la configuración general de las funciones en la documentación de Serverless Framework.

Independientemente del mecanismo que utilice para establecer las variables de entorno del contenedor, hay algunas que están reservadas por el contenedor, como las documentadas para Lambda en las variables de entorno de tiempo de ejecución definidas. Consulte siempre la documentación oficial del contenedor que utilice para determinar cómo se tratan las variables de entorno y si hay alguna restricción.

Soporte de propiedades del sistema JVM

Las propiedades del sistema JVM proporcionan otra forma de especificar las opciones de configuración y las credenciales de los SDK que se ejecutan en la JVM, como el y el. AWS SDK for Java AWS SDK para KotlinPara obtener una lista de las propiedades del sistema JVM compatibles con los SDK, consulte la referencia de configuración.

Prioridad de las opciones

- Si especifica una configuración mediante su propiedad de sistema JVM, anulará cualquier valor que se encuentre en las variables de entorno o que se cargue desde un perfil en los archivos `config` y `credentials` AWS compartidos.

- Si especifica una configuración mediante su variable de entorno, anulará cualquier valor cargado desde un perfil en los `credentials` archivos `config` y AWS compartidos.

¿Cómo configurar las propiedades del sistema JVM

Puede configurar las propiedades del sistema JVM de varias maneras.

En la línea de comandos

Establezca las propiedades del sistema JVM en la línea de comandos al invocar el `java` comando mediante el conmutador. `-D` El siguiente comando lo configura Región de AWS globalmente para todos los clientes del servicio, a menos que se anule explícitamente el valor del código.

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

Si necesita configurar varias propiedades del sistema JVM, especifique el `-D` conmutador varias veces.

Con una variable de entorno

Si no puede acceder a la línea de comandos para invocar la JVM y ejecutar la aplicación, puede usar la variable de `JAVA_TOOL_OPTIONS` entorno para configurar las opciones de la línea de comandos. Este enfoque resulta útil en situaciones como la ejecución de una AWS Lambda función en el entorno de ejecución de Java o la ejecución de código en una JVM integrada.

En el siguiente ejemplo, se configura Región de AWS globalmente para todos los clientes del servicio, a menos que se anule explícitamente el valor del código.

Linux, macOS, or Unix

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

La configuración de la variable de entorno cambia el valor usado hasta el final de su sesión del intérprete de comandos o hasta que otorgue a la variable un valor diferente. Puede hacer que las variables persistan en sesiones futuras configurándolas en el script de startup del intérprete de comandos.

Windows Command Prompt

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

El uso de `set` para configurar una variable de entorno cambia el valor usado hasta que finalice la sesión de Símbolo del Sistema actual o hasta que otorgue a la variable un valor diferente. El uso de `setx` para establecer una variable de entorno cambia el valor usado en la sesión de Símbolo del Sistema actual y en todas las sesiones de Símbolo del Sistema que cree después de ejecutar el comando. La operación no afecta a otros comandos del shell que ya se están ejecutando en el momento de ejecutar el comando.

En tiempo de ejecución

También puede establecer las propiedades del sistema JVM en tiempo de ejecución en el código mediante el `System.setProperty` método que se muestra en el siguiente ejemplo.

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

> **Important**
>
> Establezca las propiedades del sistema JVM antes de inicializar los clientes del servicio del SDK; de lo contrario, los clientes del servicio podrían utilizar otros valores.

Autenticación y acceso

Debe establecer cómo se autentica su código con AWS cuando se desarrolla con Servicios de AWS. Puede configurar el acceso programático a AWS recursos de diferentes maneras, según el entorno y la AWS acceso disponible para usted.

Opciones de autenticación para el código que se ejecuta localmente (no en AWS)

- Autenticación del Centro de identidades de IAM— Como práctica recomendada de seguridad, recomendamos utilizar AWS Organizations con IAM Identity Center para gestionar el acceso en todos sus Cuentas de AWS. Puede crear usuarios en AWS IAM Identity Center, utilice Microsoft Active Directory, utilice un proveedor de identidades (IdP) SAML 2.0 o federe individualmente su IdP para Cuentas de AWS. Para comprobar si su región es compatible con IAM Identity Center, consulte AWS IAM Identity Center puntos finales y cuotas en el Referencia general de Amazon Web Services.
- Funciones de IAM en cualquier lugar— Puede utilizar IAM Roles Anywhere para obtener credenciales de seguridad temporales IAM para cargas de trabajo, como servidores, contenedores y aplicaciones que se ejecutan fuera de AWS. Para usar IAM Roles Anywhere, sus cargas de trabajo deben usar certificados X.509.
- Asumir un rol— Puedes asumir un IAM rol para acceder temporalmente AWS recursos a los que de otro modo no tendría acceso.
- AWS claves de acceso— Otras opciones que podrían resultar menos prácticas o que podrían aumentar el riesgo de seguridad para su AWS recursos.

Opciones de autenticación para el código que se ejecuta en un AWS entorno

Si su código se ejecuta en AWS, las credenciales se pueden poner automáticamente a disposición de su aplicación. Por ejemplo, si su aplicación está alojada en Amazon Elastic Compute Cloud y hay un IAM rol asociado a ese recurso, las credenciales estarán disponibles automáticamente para su aplicación. Del mismo modo, si utilizas Amazon ECS o EKS contenedores de Amazon, el código que se ejecuta dentro del contenedor a través de la cadena de proveedores de credenciales del IAM rol puede obtener automáticamente las SDK credenciales configuradas para el rol.

- Uso de IAM roles para EC2 instancias de Amazon— Usa IAM roles para ejecutar tu aplicación de forma segura en una EC2 instancia de Amazon.

- Puede interactuar mediante programación con AWS utilizando IAM Identity Center de las siguientes maneras:
  - Utilizar AWS CloudShellpara ejecutar AWS CLI comandos desde la consola.
  - Si quieres probar un espacio de colaboración basado en la nube para equipos de desarrollo de software, considera usar Amazon CodeCatalyst.

Autenticación a través de un proveedor de identidades basado en web, aplicaciones web móviles o basadas en cliente

Si está creando aplicaciones móviles o aplicaciones web basadas en clientes que requieren acceso a AWS, cree su aplicación de modo que solicite datos temporales AWS credenciales de seguridad de forma dinámica mediante la federación de identidades web.

Con la federación de identidades web no necesita crear código de inicio de sesión personalizado ni administrar sus propias identidades de usuario. En su lugar, los usuarios de la aplicación pueden iniciar sesión con un proveedor de identidad externo (IdP) conocido, como Login with Amazon, Facebook, Google o cualquier otro IdP compatible con OpenID Connect ()OIDC. Pueden recibir un token de autenticación y, a continuación, cambiarlo por credenciales de seguridad temporales en AWS que se asignan a un IAM rol con permisos para usar los recursos de su Cuenta de AWS.

Para obtener información sobre cómo configurar esto para su herramienta SDK o herramienta, consulteFederar con identidad web u OpenID Connect.

Para aplicaciones móviles, le recomendamos que utilice Amazon Cognito. Amazon Cognito actúa como agente de identidades y realiza gran parte del trabajo de federación por usted. Para obtener más información, consulte Uso de Amazon Cognito para aplicaciones móviles en la Guía del IAMusuario.

Más información sobre la administración de acceso

La guía del IAM usuario contiene la siguiente información sobre cómo controlar de forma segura el acceso a AWS recursos:

- IAMIdentidades (usuarios, grupos de usuarios y roles): comprenda los conceptos básicos de las identidades en AWS.
- Prácticas recomendadas de seguridad en IAM: recomendaciones de seguridad que se deben seguir a la hora de desarrollar AWS aplicaciones según el modelo de responsabilidad compartida.

Con la Referencia general de Amazon Web Servicestiene los conceptos básicos básicos sobre lo siguiente:

- Comprender y obtener su AWS credenciales: acceda a las principales opciones y prácticas de administración tanto para el acceso mediante consola como mediante programación.

ID de creador de AWS

Sus ID de creador de AWS complementa cualquier Cuentas de AWS puede que ya tengas o quieras crear. Mientras que un Cuenta de AWS actúa como contenedor para AWS los recursos que cree y proporciona un límite de seguridad para esos recursos, su ID de creador de AWS lo representa como individuo. Puedes iniciar sesión con tu ID de creador de AWS para acceder a herramientas y servicios para desarrolladores como Amazon CodeWhisperer y Amazon CodeCatalyst.

- Inicia sesión con ID de creador de AWS en la AWS Sign-In Guía del usuario: aprenda a crear y usar un ID de creador de AWS y aprenda lo que proporciona el Builder ID.
- Autenticarse con y CodeWhisperer AWS Toolkit - Builder ID en la guía CodeWhisperer del usuario: aprenda a usar CodeWhisperer un ID de creador de AWS.
- CodeCatalyst conceptos - ID de creador de AWSen la Guía del CodeCatalyst usuario de Amazon: descubre cómo se CodeCatalyst usa un ID de creador de AWS.

Autenticación del Centro de identidades de IAM

AWS IAM Identity Center es el método recomendado para proporcionar AWS credenciales cuando se desarrolla en un servicio que no es AWS informático. Por ejemplo, sería algo así como su entorno de desarrollo local. Si estás desarrollando en un AWS recurso, como Amazon Elastic Compute Cloud (Amazon EC2) AWS Cloud9 o, te recomendamos que obtengas las credenciales de ese servicio.

En este tutorial, establecerá el acceso al centro de identidad de IAM y lo configurará para su SDK o herramienta mediante el portal de AWS acceso y el. AWS CLI

- El portal de AWS acceso es la ubicación web en la que se inicia sesión manualmente en el Centro de identidades de IAM. El formato de la URL es `d-xxxxxxxxxx.awsapps.com/start` o *`your_subdomain`*`.awsapps.com/start`. Al iniciar sesión en el portal de AWS acceso, puede ver Cuentas de AWS los roles que se han configurado para ese usuario. Este procedimiento utiliza el portal de AWS acceso para obtener los valores de configuración que necesita para el proceso de autenticación del SDK o la herramienta.

- AWS CLI Se utiliza para configurar el SDK o la herramienta para que utilice la autenticación del Centro de Identidad de IAM para las llamadas a la API realizadas por el código. Este proceso único actualiza el AWS `config` archivo compartido, que luego es utilizado por el SDK o la herramienta al ejecutar el código.

Configuración del acceso mediante programación mediante el Centro de identidades de IAM

Paso 1: Establecer el acceso y seleccionar el conjunto de permisos adecuado

Si aún no has activado el Centro de Identidad de IAM, consulta Cómo activar el Centro de Identidad de IAM en la Guía del AWS IAM Identity Center usuario.

Elija uno de los siguientes métodos para acceder a sus credenciales. AWS

No he establecido el acceso a través del Centro de identidades de IAM

1. Añada un usuario y añada permisos administrativos siguiendo el procedimiento de configuración del acceso de los usuarios con el directorio predeterminado del IAM Identity Center de la Guía del AWS IAM Identity Center usuario.
2. El conjunto de `AdministratorAccess` permisos no debe utilizarse para un desarrollo normal. En su lugar, le recomendamos que utilice el conjunto de `PowerUserAccess` permisos predefinido, a menos que su empresa haya creado un conjunto de permisos personalizado para este fin.

   Siga el mismo procedimiento para configurar el acceso de los usuarios con el directorio predeterminado del Centro de Identidad de IAM, pero esta vez:

   - En lugar de crear el *`Admin team`* grupo, cree un *`Dev team`* grupo y sustitúyalo por éste a continuación en las instrucciones.
   - Puede usar el usuario existente, pero debe agregarlo al nuevo *`Dev team`* grupo.
   - En lugar de crear el conjunto de *`AdministratorAccess`* permisos, cree un conjunto de *`PowerUserAccess`* permisos y sustitúyalo posteriormente en las instrucciones.

   Cuando haya terminado, debería disponer de lo siguiente:

   - Un `Dev team` grupo.

- Un conjunto de `PowerUserAccess` permisos adjunto al `Dev team` grupo.
- El usuario se ha añadido al `Dev team` grupo.

3. Salga del portal e inicie sesión de nuevo para ver sus opciones Cuentas de AWS y para `Administrator` o`PowerUserAccess`. Seleccione esta opción `PowerUserAccess` cuando trabaje con su herramienta o SDK.

Ya tengo acceso a AWS través de un proveedor de identidad federado administrado por mi empresa (como Microsoft Entra u Okta)

Inicia sesión a AWS través del portal de tu proveedor de identidad. Si el administrador de la nube te ha concedido permisos `PowerUserAccess` (de desarrollador), verás aquellos a los Cuentas de AWS que tienes acceso y tu conjunto de permisos. Junto al nombre de su conjunto de permisos, verá las opciones para acceder a las cuentas de forma manual o programática mediante ese conjunto de permisos.

Las implementaciones personalizadas pueden dar lugar a experiencias diferentes, como distintos nombres de conjuntos de permisos. Si no está seguro de qué configuración de permisos debe utilizar, contacte con su equipo de TI para obtener ayuda.

Ya tengo acceso a él a AWS través del portal de AWS acceso gestionado por mi empresa

Inicie sesión a AWS través del portal de AWS acceso. Si el administrador de la nube te ha concedido permisos `PowerUserAccess` (de desarrollador), verás los permisos a los Cuentas de AWS que tienes acceso y tu conjunto de permisos. Junto al nombre de su conjunto de permisos, verá las opciones para acceder a las cuentas de forma manual o programática mediante ese conjunto de permisos.

Ya tengo acceso a AWS través de un proveedor de identidad personalizado federado administrado por mi empleador

Contacte con su equipo de TI para obtener ayuda.

Paso 2: Configure los SDK y las Herramientas para usar el IAM Identity Center

1. En su máquina de desarrollo, instale la versión más reciente AWS CLI.

   a. Consulte Instalación o actualización de la versión más reciente de la AWS CLI en la Guía del usuario de AWS Command Line Interface .

b. (Opcional) Para comprobar que funciona, abra una línea de comandos y ejecute el `aws --version` comando. AWS CLI

2. Inicie sesión en el portal de AWS acceso. Es posible que su empresa le facilite esta URL o que la reciba en un correo electrónico tras el paso 1: establecer el acceso. Si no es así, busque la URL de su portal de AWS acceso en el panel de control de https://console.aws.amazon.com/singlesignon/.

   a. En el portal de AWS acceso, en la pestaña Cuentas, seleccione la cuenta individual que desee administrar. Se muestran las funciones de su usuario. Elija las claves de acceso para obtener las credenciales de acceso mediante línea de comandos o mediante programación para el conjunto de permisos correspondiente. Utilice el conjunto de permisos `PowerUserAccess` predefinido o el conjunto de permisos que usted o su empleador hayan creado para aplicar permisos de privilegio mínimo para el desarrollo.

   b. En el cuadro de diálogo Obtener credenciales, elija MacOS y Linux o Windows, en función del sistema operativo.

   c. Elija el método de Credenciales del IAM Identity Center para obtener los valores `SSO Start URL` y `SSO Region` que necesita para el próximo paso.

3. En la AWS CLI línea de comandos, ejecute el `aws configure sso` comando. Cuando se le solicite, introduzca los valores de configuración que recopiló en el paso anterior. Para obtener más información sobre este AWS CLI comando, consulte Configurar su perfil con el `aws configure sso` asistente.

   - Para el Nombre del perfil CLI, le recomendamos que introduzca el *`valor predeterminado`* al empezar. Para obtener información sobre cómo configurar perfiles no predeterminados (con nombre) y su variable de entorno asociada, consulte Perfiles.

4. (Opcional) En la AWS CLI línea de comandos, confirme la identidad de la sesión activa ejecutando el `aws sts get-caller-identity` comando. La respuesta debería mostrar el conjunto de permisos del IAM Identity Center que configuró.

5. Si utiliza un AWS SDK, cree una aplicación para su SDK en su entorno de desarrollo.

   a. En el caso de algunos SDK, es necesario añadir paquetes adicionales como `SSO` y `SSOOIDC`, a la aplicación antes de poder utilizar la autenticación del IAM Identity Center. Para obtener más detalles, consulte su SDK específica.

   b. Si anteriormente configuraste el acceso a AWS, revisa tu AWS `credentials` archivo compartido para ver si hay algunoAWS claves de acceso. Debe eliminar todas las

credenciales estáticas antes de que el SDK o la herramienta utilicen las credenciales del IAM Identity Center debido a la precedencia Cadena de proveedores de credenciales.

Para obtener información detallada sobre cómo los SDK y las herramientas utilizan y actualizan las credenciales con esta configuración, consulte Comprender la autenticación del Centro de identidades de IAM.

En función de la duración de las sesiones configuradas, el acceso eventualmente caducará y los SDK o las Herramientas detectarán un error de autenticación. Para volver a actualizar la sesión del portal de acceso cuando sea necesario, utilice el comando AWS CLI para ejecutar el `aws sso login` comando.

Puede ampliar tanto la duración de la sesión del portal de acceso al IAM Identity Center como la duración de la sesión del conjunto de permisos. Esto prolonga el tiempo que puede ejecutar el código antes de tener que volver a iniciar sesión manualmente con el AWS CLI. Para obtener más información, consulte los siguientes temas en la Guía del usuario de AWS IAM Identity Center :

- Duración de la sesión del IAM Identity Center: configure la duración de las sesiones del portal de AWS acceso de sus usuarios
- Duración de la sesión establecida por permisos: Establecer la duración de la sesión

Para obtener más información sobre la configuración del proveedor del Centro de Identidades de IAM para los SDK y las herramientas, consulte IAMProveedor de credenciales de Identity Center en esta guía.

Comprender la autenticación del Centro de identidades de IAM

Términos relevantes del Centro de identidades de IAM

Los siguientes términos le ayudan a entender el proceso y la configuración subyacentes en AWS IAM Identity Center. La documentación de las API del SDK de AWS utiliza nombres diferentes a los del Centro de identidades de IAM para algunos de estos conceptos de autenticación. Resulta útil conocer ambos nombres.

En la siguiente tabla, se muestra cómo se relacionan entre sí los nombres alternativos.

Nombre del Centro de identidades de IAM	Nombre de la API del SDK	Descripción
Centro de identidades	`sso`	Aunque se haya cambiado el nombre de inicio de sesión único de AWS, los espacios de nombres de las API de `sso` mantendrá n su nombre original por motivos de compatibilidad con versiones anteriores. Para más información, consulte Cambiar el nombre del Centro de identidades de IAM en la Guía del usuario de AWS IAM Identity Center.
Consola del Centro de identidades de IAM Consola administrativa		La consola que se utiliza para configurar el inicio de sesión único.
URL del portal de acceso a AWS		Una URL exclusiva de su cuenta del Centro de identidades de IAM, como `https://xxx.awsapps.com/start` . Inicie sesión en este portal con sus credenciales de inicio de sesión del Centro de identidades de IAM.
Sesión del Portal de Acceso del Centro de identidades de IAM	Sesión de autenticación	Proporciona un token de acceso al portador al intermediario.
Sesión del conjunto de permisos		La sesión de IAM que el SDK usa internamente para realizar

Nombre del Centro de identidades de IAM	Nombre de la API del SDK	Descripción
		las llamadas de Servicio de AWS. En las discusiones informales, es posible que vea que esto se denomina incorrectamente “sesión de roles”.
Credenciales de configuración de permisos	Credenciales de AWS Single Sign-On credenciales de sigv4	Las credenciales que el SDK utiliza realmente para la mayoría de las llamadas de Servicio de AWS (específi camente, todas las llamadas sigv4 de Servicio de AWS). En las discusiones informales, es posible que veas que esto se denomina incorrectamente “credenciales de roles”.
Proveedor de credenciales del IAM Identity Center	Proveedor de credenciales SSO	Cómo se obtienen las credenciales, como la clase o el módulo que proporciona la funcionalidad.

Comprenda la resolución de credenciales del SDK para Servicios de AWS

La API del IAM Identity Center intercambia credenciales de token de portador por credenciales sigv4. La mayoría son API sigv4 de Servicios de AWS, con algunas excepciones, como Amazon CodeWhisperer y Amazon CodeCatalyst. A continuación, se describe el proceso de resolución de credenciales para admitir la mayoría de las llamadas de Servicio de AWS mediante el código de la aplicación AWS IAM Identity Center.

Iniciar una sesión en el portal de acceso a AWS

- Inicie el proceso iniciando sesión con sus credenciales.

  - Use el comando de `aws sso login` en el AWS Command Line Interface (AWS CLI). Esto inicia una nueva sesión en el IAM Identity Center si aún no tiene una sesión activa.
- Al iniciar una nueva sesión, recibirá un token de actualización y un token de acceso del IAM Identity Center. El AWS CLI también actualiza un archivo JSON de la caché del SSO con un nuevo token de acceso y un token de actualización, y lo pone a disposición de los SDK para que lo utilicen.
- Si ya tiene una sesión activa, el AWS CLI comando reutiliza la sesión existente y caducará cuando caduque la sesión existente. Para obtener información sobre cómo establecer la duración de una sesión del IAM Identity Center, consulte Configurar la duración de las sesiones del portal de acceso de los usuarios de AWS en la AWS IAM Identity CenterGuía del usuario.
  - La duración máxima de la sesión se ha ampliado a 90 días para reducir la necesidad de iniciar sesión con frecuencia.

Cómo obtiene el SDK las credenciales para las llamadas a Servicio de AWS

Los SDK proporcionan acceso a Servicios de AWS cuando se crea una instancia de un objeto de cliente por servicio. Cuando el perfil seleccionado del archivo compartido de AWS `config` está configurado para la resolución de credenciales del IAM Identity Center, el IAM Identity Center se utiliza para resolver las credenciales de su aplicación.

- El proceso de resolución de credenciales se completa durante el tiempo de ejecución cuando se crea un cliente.

Para recuperar las credenciales de las API sigv4 mediante el inicio de sesión único del IAM Identity Center, el SDK utiliza el token de acceso al IAM Identity Center para obtener una sesión de IAM. Esta sesión de IAM se denomina sesión de conjunto de permisos y proporciona a AWS acceso al SDK al asumir un rol de IAM.

- La duración de la sesión del conjunto de permisos se establece independientemente de la duración de la sesión del IAM Identity Center.
  - Para obtener información sobre cómo configurar la duración de la sesión del conjunto de permisos, consulte Establecer la duración de la sesión en la Guía del usuario de AWS IAM Identity Center.
- Ten en cuenta que las credenciales del conjunto de permisos también se denominan credenciales de AWS y credenciales sigv4 en la mayoría de la documentación de la API del SDK de AWS.

Las credenciales del conjunto de permisos se devuelven de una llamada a getRoleCredentials de la API del Centro de identidades de IAM al SDK. El objeto cliente del SDK utiliza ese supuesto rol de IAM para realizar llamadas al Servicio de AWS, por ejemplo, pedir a Amazon S3 que incluya los buckets en su cuenta. El objeto de cliente puede seguir funcionando con esas credenciales del conjunto de permisos hasta que caduque la sesión del conjunto de permisos.

Caducidad y actualización de la sesión

Al utilizar el SSOconfiguración del proveedor de tokens, el token de acceso por hora obtenido del Centro de identidades de IAM se actualiza automáticamente mediante el token de actualización.

- Si el token de acceso ha caducado cuando el SDK intenta usarlo, el SDK utiliza el token de actualización para intentar obtener un nuevo token de acceso. El Centro de identidades de IAM compara el token de actualización con la duración de la sesión del portal de acceso al Centro de identidades de IAM. Si el token de actualización no ha caducado, el Centro de identidades de IAM responde con otro token de acceso.
- Este token de acceso se puede utilizar para actualizar la sesión del conjunto de permisos de los clientes existentes o para resolver las credenciales de los nuevos clientes.

Sin embargo, si la sesión del portal de acceso del Centro de identidades de IAM ha caducado, no se concede ningún token de acceso nuevo. Por lo tanto, la duración del conjunto de permisos no se puede renovar. Caducará (y se perderá el acceso) cuando se agote el tiempo de espera de la sesión del conjunto de permisos almacenado en caché para los clientes existentes.

Cualquier código que cree un nuevo cliente no se autenticará en cuanto caduque la sesión del Centro de identidades de IAM. Esto se debe a que las credenciales del conjunto de permisos no se almacenan en caché. Su código no podrá crear un nuevo cliente ni completar el proceso de resolución de credenciales hasta que tenga un token de acceso válido.

En resumen, cuando el SDK necesita nuevas credenciales de conjunto de permisos, primero compruebe si hay credenciales válidas y existentes y si las utiliza. Esto se aplica tanto si las credenciales son para un cliente nuevo como para un cliente existente con credenciales caducadas. Si no se encuentran las credenciales o no son válidas, el SDK llama a la API del Centro de identidades de IAM para obtener nuevas credenciales. Para llamar a la API, necesita el token de acceso. Si el token de acceso ha caducado, el SDK utiliza el token de actualización para intentar obtener un nuevo token de acceso del servicio del Centro de identidades de IAM. Este token se concede si la sesión del portal de acceso al IAM Identity Center no ha caducado.

Funciones de IAM en cualquier lugar

Puede utilizar Funciones de IAM en cualquier lugar para obtener credenciales de seguridad temporales en IAM para cargas de trabajo, como servidores, contenedores y aplicaciones que se ejecutan fuera de AWS. Para utilizar Funciones de IAM en cualquier lugar, sus cargas de trabajo deben utilizar certificados X.509. El administrador de la nube debe proporcionar el certificado y la clave privada necesarios para configurar Funciones de IAM en cualquier lugar como su proveedor de credenciales.

Paso 1: Configurar las Funciones de IAM en cualquier lugar

Las funciones de IAM en cualquier lugar proporcionan una forma de obtener credenciales temporales para una carga de trabajo o un proceso que se ejecuta fuera de AWS. Se establece un anclaje de confianza con la autoridad de certificación para obtener credenciales temporales para el rol de IAM asociado. El rol establece los permisos que tendrá su carga de trabajo cuando su código se autentique con las Funciones de IAM en cualquier lugar.

Para ver los pasos necesarios para configurar el anclaje de confianza, el rol de IAM y el perfil de Funciones de IAM en cualquier lugar, consulte Creación de un anclaje de confianza y un perfil en Funciones de AWS Identity and Access Management en cualquier lugar en la Guía del usuario de Funciones de IAM en cualquier lugar.

> **Note**
>
> Un perfil en la Guía de usuario de Funciones de IAM en cualquier lugar hace referencia a un concepto exclusivo del servicio de Funciones de IAM en cualquier lugar. No está relacionado con los perfiles del archivo compartido AWS `config`.

Paso 2: Utilice las funciones de IAM en cualquier lugar

Para obtener credenciales de seguridad temporales de Funciones de IAM en cualquier lugar, utilice la herramienta ayudante de credenciales proporcionada por Funciones de IAM en cualquier lugar. La herramienta de credenciales implementa el proceso de firma de Funciones de IAM en cualquier lugar.

Para obtener instrucciones sobre cómo descargar la herramienta del ayudante de credenciales, consulte Obtener credenciales de seguridad temporales de Funciones de AWS Identity and Access Management en cualquier lugar en la Guía del usuario de Funciones de IAM en cualquier lugar.

Para utilizar credenciales de seguridad temporales de Funciones de IAM en cualquier lugar con los SDK de AWS y el AWS CLI, puede configurar `credential_process` los ajustes del archivo compartido AWS `config`. Los SDK y AWS CLI son compatibles con un proveedor de credenciales de proceso que se utiliza `credential_process` para autenticarse. A continuación se muestra la estructura general para establecer `credential_process`.

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

El comando `credential-process` de la herramienta auxiliar devuelve las credenciales temporales en un formato JSON estándar que es compatible con la configuración `credential_process`. Tenga en cuenta que el nombre del comando contiene un guión, pero el nombre de la configuración contiene un guión bajo. El comando requiere los parámetros siguientes:

- `private-key` – La ruta a la clave privada que firmó la solicitud.
- `certificate` – La ruta al certificado.
- `role-arn` – El ARN del rol para el que se van a obtener las credenciales temporales.
- `profile-arn` – El ARN del perfil que proporciona una asignación para el rol especificado.
- `trust-anchor-arn` – El ARN del anclaje de confianza usado para autenticar.

Su administrador de la nube debe proporcionarle el certificado y la clave privada. Los tres valores del ARN se pueden copiar de AWS Management Console. El siguiente ejemplo muestra un archivo compartido `config` que configura la recuperación de credenciales temporales de la herramienta 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 ver los parámetros opcionales y los detalles adicionales de las herramientas auxiliares, consulte el Ayudante de credenciales de las Funciones de IAM en cualquier lugar en GitHub.

Para obtener más información sobre el ajuste de configuración del SDK en sí y el proveedor de credenciales del proceso, consulte Proveedor de credenciales de proceso en esta guía.

Asumir un rol

Para asumir un rol, se utiliza un conjunto de credenciales de seguridad temporales para acceder a los recursos de AWS a los que de otro modo usted no tendría acceso. Las credenciales temporales incluyen un ID de clave de acceso, una clave de acceso secreta y un token de seguridad. Para obtener más información sobre las solicitudes de la API de AWS Security Token Service (AWS STS), consulte Acciones en la Referencia de la API de AWS Security Token Service.

Para configurar el SDK o la herramienta para que asuma un rol, primero debe crear o identificar el rol específico que desee asumir. Los roles de IAM se identifican de forma exclusiva mediante un nombre de recurso de Amazon (ARN) del rol. Los roles establecen relaciones de confianza con otra entidad. La entidad de confianza que usa el rol puede ser un Servicio de AWS, otro Cuenta de AWS, un proveedor de identidad web o una federación OIDC o SAML. Para más información acerca de los roles de IAM, consulte Roles de IAM en la Guía del usuario de IAM.

Una vez identificado el rol de IAM, si esa función confía en usted, puede configurar el SDK o la herramienta para que utilice los permisos que otorga la función. Para ello, puede elegir entre Asumir un rol de IAM. o Federar con identidad web u OpenID Connect.

Asumir un rol de IAM.

Al asumir un rol, AWS STS devuelve un conjunto de credenciales de seguridad temporales. Estas credenciales provienen de otro perfil o de la instancia o contenedor en el que se ejecuta el código. Otros ejemplos de cómo asumir un rol incluyen la administración de múltiples Cuentas de AWS desde Amazon EC2, el uso de AWS CodeCommit en las Cuentas de AWS o el acceso a otra cuenta desde AWS CodeBuild.

Paso 1: Configurar un rol de IAM

Para configurar el SDK o la herramienta para que asuma un rol, primero debe crear o identificar el rol específico que desee asumir. Los roles de IAM se identifican de forma exclusiva mediante un ARN de rol. Los roles establecen relaciones de confianza con otra entidad, normalmente dentro de su cuenta o para el acceso entre cuentas. Para obtener más información, consulte Creación de roles de IAM en la Guía del usuario de IAM.

Paso 2: Configurar el SDK o la herramienta

Configure el SDK o la herramienta para obtener las credenciales de `credential_source` o `source_profile`.

Se utiliza `credential_source` para obtener credenciales de un contenedor de Amazon ECS, de una instancia de Amazon EC2 o de variables de entorno.

Se utiliza `source_profile` para obtener credenciales de otro perfil. `source_profile`también admite el encadenamiento de roles, que consiste en jerarquías de perfiles en las que se utiliza un rol asumido para asumir otro rol.

Al especificar esto en un perfil, la herramienta o SDK realiza automáticamente la llamada a la API de AWS STS `AssumeRole` correspondiente. Para recuperar y usar credenciales temporales asumiendo un rol, especifique los siguientes valores de configuración en el archivo compartido `config` de AWS. Para obtener más información sobre esta configuración, consulte la sección Asumir la configuración del proveedor de credenciales de rol.

- `role_arn` - Del rol de IAM que creó en el paso 1
- Configure una de las siguientes opciones: `source_profile` o `credential_source`
- (Opcional) `duration_seconds`
- (Opcional) `external_id`
- (Opcional) `mfa_serial`
- (Opcional) `role_session_name`

Los siguientes ejemplos muestran la configuración de ambas opciones de asumir roles en un archivo `config` compartido:

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

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

Para obtener más información sobre la configuración del proveedor de credenciales de rol, consulte Asumir el rol de proveedor de credenciales en esta guía.

Federar con identidad web u OpenID Connect

Al crear aplicaciones móviles o aplicaciones web basadas en clientes que requieren acceso a AWS, AWS STS devuelve un conjunto de credenciales de seguridad temporales para los usuarios

federados que se autentican a través de un proveedor de identidades (IdP) público. Entre los ejemplos de proveedores de identidad públicos se incluyen Login with Amazon, Facebook, Google o cualquier proveedor de identidad compatible con OpenID Connect (OIDC). Con este método, los usuarios no necesitan su propia identidad AWS ni la de IAM.

Si utiliza Amazon Elastic Kubernetes Service, esta característica permite especificar diferentes roles de IAM para cada uno de sus contenedores. Kubernetes ofrece la posibilidad de distribuir los tokens de OIDC a sus contenedores, que este proveedor de credenciales utiliza para obtener credenciales temporales. Para obtener más información sobre esta configuración de Amazon EKS, consulte Roles de IAM para cuentas de servicio en la Guía del usuario de Amazon EKS. Sin embargo, para simplificar el proceso, le recomendamos que utilice Amazon EKS Pod Identities si su SDK es compatible.

Paso 1: Configurar un proveedor de identidades y un rol de IAM

Si desea configurar la federación con un IdP externo, utilice un proveedor de identidades de IAM para informar a AWS sobre el IdP externo y su configuración. Esto establece una relación de confianza entre su Cuenta de AWS y el IdP (proveedor de identidades) externo. Antes de configurar el SDK para usar el token de identidad web para la autenticación, primero debe configurar el proveedor de identidad (IdP) y el rol de IAM que se usa para acceder a él. Para configurarlos, consulte Creación de un rol para identidades web u OpenID Connect Federation (consola) en la Guía del usuario de IAM.

Paso 2: Configurar el SDK o la herramienta

Configure el SDK o la herramienta para usar un token de identidad web de AWS STS para la autenticación.

Al especificar esto en un perfil, la herramienta o SDK realiza automáticamente la llamada a la API de AWS STS `AssumeRoleWithWebIdentity` correspondiente. Para recuperar y utilizar credenciales temporales utilizando federación de identidades web, puede especificar los siguientes valores de configuración en el archivo `config` compartido de AWS. Para obtener más información sobre esta configuración, consulte la sección Asumir la configuración del proveedor de credenciales de rol.

- `role_arn` - Del rol de IAM que creó en el paso 1
- `web_identity_token_file` - Desde el IdP externo
- (Opcional) `duration_seconds`
- (Opcional) `role_session_name`

El siguiente es un ejemplo de una configuración de archivos compartidos de `config` para asumir un rol con identidad web:

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

> **Note**
>
> Para aplicaciones móviles, le recomendamos que utilice Amazon Cognito. Amazon Cognito actúa como agente de identidades y realiza gran parte del trabajo de federación por usted. Sin embargo, el proveedor de identidades de Amazon Cognito no está incluido en las bibliotecas principales de SDK y herramientas como otros proveedores de identidades. Para acceder a la API de Amazon Cognito, incluya el cliente del servicio Amazon Cognito en la compilación o las bibliotecas de su SDK o herramienta. Para su uso con los SDK AWS, consulte los ejemplos de código en la Guía para desarrolladores de Amazon Cognito.

Para obtener más información sobre la configuración del proveedor de credenciales de rol, consulte Asumir el rol de proveedor de credenciales en esta guía.

AWS claves de acceso

Use credenciales a corto plazo.

Recomendamos configurar su SDK o herramienta para utilizar Autenticación del Centro de identidades de IAM para usar opciones de duración de sesión ampliada.

Sin embargo, para configurar directamente las credenciales temporales del SDK o de la herramienta, consulte Autenticación mediante credenciales a corto plazo.

Use credenciales a largo plazo.

> **Warning**
>
> Para evitar riesgos de seguridad, no utilice a los usuarios de IAM para la autenticación cuando desarrolle software especialmente diseñado o trabaje con datos reales. En cambio, utilice la federación con un proveedor de identidades como AWS IAM Identity Center.

Gestione el acceso en todos Cuentas de AWS

Como práctica recomendada de seguridad, te recomendamos que utilices AWS Organizations IAM Identity Center para gestionar el acceso en todas tus Cuentas de AWSinstalaciones. Para más información, consulte Prácticas recomendadas de seguridad en IAM en la Guía del usuario de IAM.

Puede crear usuarios en el Centro de identidades de IAM, usar Microsoft Active Directory, usar un proveedor de identidades (IdP) SAML 2.0 o federar individualmente su IdP a. Cuentas de AWSCon una de estas opciones podrá ofrecer a sus usuarios una experiencia de inicio de sesión único. También puede aplicar la autenticación multifactor (MFA) y utilizar credenciales Cuenta de AWS temporales para el acceso. El caso de un usuario de IAM es diferente, ya que utiliza una credencial de larga duración que se puede compartir y que podría aumentar el riesgo de seguridad de sus recursos de AWS .

Creación de usuarios de IAM únicamente para entornos aislados

Si es la primera vez que lo usa AWS, puede crear un usuario de IAM de prueba y luego usarlo para ejecutar tutoriales y explorar lo que AWS ofrece. Está bien usar este tipo de credenciales cuando se está aprendiendo, pero no recomendamos usarlas fuera de un entorno aislado.

Para los siguientes casos de uso, podría ser conveniente empezar con los usuarios de IAM en: AWS

- Cómo empezar a utilizar el AWS SDK o la herramienta y explorarlos Servicios de AWS en un entorno aislado.
- Como parte de su aprendizaje, ejecute scripts, trabajos y otros procesos automatizados programados que no admitan un proceso de inicio de sesión asistido por una persona.

Si utilizas usuarios de IAM fuera de estos casos de uso, cámbiate al Centro de Identidad de IAM o federa tu proveedor de identidades Cuentas de AWS lo antes posible. Para obtener más información, consulte Federación de identidades en AWS.

Seguridad para las claves de acceso de los usuarios de IAM

Debe rotar las claves de acceso de los usuarios de IAM regularmente. Siga las instrucciones de Rotación de las claves de acceso disponibles en la Guía del usuario de IAM. Si considera que puede haber compartido accidentalmente sus claves de acceso de usuario de IAM, cámbielas.

Las claves de acceso de los usuarios de IAM deben almacenarse en el AWS `credentials` archivo compartido de la máquina local. No guarde las claves de acceso de los usuarios de IAM en su

código. No incluya archivos de configuración que contengan sus claves de acceso de usuario de IAM en ningún software de administración de código fuente. Las herramientas externas, como el proyecto de código abierto git-secrets, pueden ayudarle a no enviar información confidencial accidentalmente a un repositorio de Git. Para obtener más información acerca de los usuarios de de IAM, consulte Identidades IAM (usuarios, grupos y funciones) en la Guía de usuario de IAM.

Para configurar un usuario de IAM para empezar, consulte Autenticar mediante credenciales a largo plazo.

Autenticación mediante credenciales a corto plazo

Le recomendamos configurar su herramienta SDK o herramienta para utilizarla Autenticación del Centro de identidades de IAM con opciones de duración de sesión prolongada. Sin embargo, puede copiar y usar las credenciales temporales que están disponibles en AWS portal de acceso. Las credenciales nuevas deberán copiarse cuando caduquen. Puede utilizar las credenciales temporales en un perfil o como valores para las propiedades del sistema y las variables de entorno.

Práctica recomendada: en lugar de gestionar manualmente las claves de acceso y un token del archivo de credenciales, recomendamos que la aplicación utilice credenciales temporales enviadas desde:

- Un registro AWS servicio de cómputo, como ejecutar la aplicación en Amazon Elastic Compute Cloud o en AWS Lambda.
- Otra opción en la cadena de proveedores de credenciales, comoAutenticación del Centro de identidades de IAM.
- O utilícela Proveedor de credenciales de proceso para recuperar credenciales temporales.

Configure un archivo de credenciales con las credenciales de corta duración recuperadas de AWS portal de acceso

1. Creación de archivos de credenciales compartidas.
2. En el archivo de credenciales, pegue el siguiente texto de marcador de posición hasta que pegue las credenciales temporales que funcionen.

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

3. Guarde el archivo. El archivo `~/.aws/credentials` debería existir ahora en su sistema de desarrollo local. Este archivo contiene el perfil [predeterminado] que utiliza la herramienta SDK o herramienta si no se especifica un perfil con nombre específico.
4. Inicie sesión en AWS acceder al portal.
5. Siga estas instrucciones para actualizar manualmente las credenciales de las credenciales a fin de copiar las credenciales de los IAM roles del AWS portal de acceso.

   a. En el paso 4 de las instrucciones enlazadas, elija el nombre del IAM rol que concede el acceso para sus necesidades de desarrollo. Este rol suele tener un nombre similar al PowerUserAccessde Desarrollador.

   b. Para el paso 7 de las instrucciones enlazadas, selecciona Añadir manualmente un perfil a tu AWS opción de fichero de credenciales y copia el contenido.
6. Pegue las credenciales copiadas en el archivo local `credentials`. No se necesita el nombre del perfil generado si está utilizando el perfil `default`. Su archivo debería parecerse a lo siguiente.

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

7. Guarde el archivo `credentials`.

Cuando SDK cree un cliente de servicio, accederá a estas credenciales temporales y las utilizará para cada solicitud. La configuración del IAM rol elegido en el paso 5a determina durante cuánto tiempo son válidas las credenciales temporales. La duración máxima es de doce horas.

Cuando las credenciales temporales caduquen, repita los pasos del 4 al 7.

Autenticar mediante credenciales a largo plazo

> **Warning**
>
> Para evitar riesgos de seguridad, no utilice a IAM los usuarios para autenticarse cuando desarrolle software diseñado específicamente o trabaje con datos reales. En su lugar, utilice la federación con un proveedor de identidad como AWS IAM Identity Center.

Si utilizas un IAM usuario para ejecutar el código, la herramienta SDK o la herramienta de tu entorno de desarrollo se autentican mediante credenciales de IAM usuario de larga duración en el entorno compartido AWS `credentials`archivo. Revise las prácticas recomendadas de seguridad del IAM tema y haga la transición a IAM Identity Center u otras credenciales temporales lo antes posible.

Advertencias y directrices importantes para las credenciales

Advertencias para las credenciales

- NOTUtilice las credenciales raíz de su cuenta para acceder AWS recursos. Estas credenciales proporcionan acceso ilimitado a la cuenta y son difíciles de revocar.
- Incluya NOT literalmente las claves de acceso o la información de credenciales en los archivos de su solicitud. Si lo hace, puede crear un riesgo de exposición accidental de sus credenciales si, por ejemplo, carga el proyecto en un repositorio público.
- NOTIncluya archivos que contengan credenciales en el área de su proyecto.
- Tenga en cuenta que cualquier credencial almacenada en el espacio compartido AWS `credentials`los archivos se almacenan en texto plano.

Guía adicional para administrar las credenciales de forma segura

Para una discusión general sobre cómo administrar de forma segura AWS las credenciales, consulte Prácticas recomendadas para la administración AWS claves de acceso en el Referencia general de AWS. Además de esa discusión, considere lo siguiente:

- Utilice IAMroles para las tareas de Amazon Elastic Container Service (AmazonECS).
- Usa IAMroles para las aplicaciones que se ejecutan en EC2 instancias de Amazon.

Requisitos previos: crear un AWS cuenta

Para utilizar un IAM usuario para acceder AWS servicios, necesita un AWS cuenta y AWS credenciales.

1. Cree una cuenta.

   Para crear un AWS cuenta, consulta Cómo empezar: ¿es la primera vez AWS usuario? en el AWS Account Management Guía de referencia.
2. Crear un usuario administrativo.

Evite usar la cuenta de usuario raíz (la cuenta inicial que cree) para acceder a la consola y los servicios de administración. En su lugar, cree una cuenta de usuario administrativo, tal y como se explica en la sección Creación de un usuario administrativo de la Guía del IAM usuario.

Después de crear la cuenta de usuario administrativo y registrar los detalles de inicio de sesión, asegúrese de desconectar la cuenta de usuario raíz y vuelva a iniciar sesión con la cuenta administrativa.

Ninguna de estas cuentas es adecuada para desarrollar en AWS o para ejecutar aplicaciones en AWS. Como práctica recomendada, debe crear usuarios, conjuntos de permisos o funciones de servicio que sean adecuados para estas tareas. Para obtener más información, consulte Aplicar permisos con privilegios mínimos en la Guía del IAM usuario.

Paso 1: Crea tu usuario IAM

- Cree su IAM usuario siguiendo el procedimiento de creación de IAM usuarios (consola) de la Guía del IAM usuario. Al crear el IAM usuario:
  - Le recomendamos que seleccione Proporcionar acceso de usuario a AWS Management Console. Esto te permite ver Servicios de AWS relacionado con el código que se está ejecutando en un entorno visual, como la comprobación AWS CloudTrail registros de diagnóstico o carga de archivos a Amazon Simple Storage Service, lo que resulta útil a la hora de depurar el código.
  - En Configurar permisos: opciones de permisos, selecciona Adjuntar políticas directamente para ver cómo quieres asignar los permisos a este usuario.
    - La mayoría de SDK los tutoriales de introducción utilizan el servicio Amazon S3 como ejemplo. Para proporcionar a su aplicación acceso completo a Amazon S3, seleccione la política `AmazonS3FullAccess` que desea asociar a este usuario.
  - Puede ignorar los pasos opcionales de ese procedimiento relacionados con la configuración de los límites o las etiquetas de los permisos.

Paso 2: Obtener las claves de acceso

1. En el panel de navegación de la IAM consola, seleccione Usuarios y, a continuación, seleccione el **`User name`** usuario que creó anteriormente.

2. En la página del usuario, selecciona la página Credenciales de seguridad. A continuación, en Claves de acceso, seleccione Crear clave de acceso.
3. En el paso 1 de Crear la clave de acceso, elija Interfaz de línea de comandos (CLI) o Código local. Ambas opciones generan el mismo tipo de clave para utilizarla con los dos AWS CLI y elSDKs.
4. En el paso 2 de Crear clave de acceso, introduzca una etiqueta opcional y seleccione Siguiente.
5. En el paso 3 de Crear la clave de acceso, selecciona Descargar el archivo.csv para guardar un `.csv` archivo con la clave de acceso y la clave de acceso secreta de tu IAM usuario. Necesitará esta información más tarde.

   > **Warning**
   >
   > Utilice las medidas de seguridad adecuadas para mantener estas credenciales seguras.

6. Seleccione Done (Listo).

Paso 3: Actualice el archivo compartido **`credentials`**

1. Crea o abre el archivo compartido AWS `credentials`archivo. Este archivo es `~/.aws/credentials` en sistemas Linux y macOS y `%USERPROFILE%\.aws\credentials` en Windows. Para obtener más información, consulte la ubicación de los archivos de credenciales.
2. Agregue el siguiente texto al archivo `credentials` compartido. Sustituya el valor de ID y el valor clave de ejemplo por los valores del archivo `.csv` que descargó anteriormente.

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

3. Guarde el archivo.

El archivo compartido `credentials` es la forma más común de almacenar las credenciales. También se pueden configurar como variables de entorno; consulte los nombres de las variables de entorno AWS claves de acceso. Esta es una forma de empezar, pero le recomendamos que haga la transición a IAM Identity Center o a otras credenciales temporales lo antes posible. Cuando deje de usar credenciales de larga duración, recuerde eliminarlas del archivo compartido `credentials`.

Uso de IAM roles para EC2 instancias de Amazon

En este ejemplo, se describe la configuración de un AWS Identity and Access Management rol con acceso a Amazon S3 para usarlo en la aplicación implementada en una EC2 instancia de Amazon.

Para ejecutar tu AWS SDK aplicación en una instancia de Amazon Elastic Compute Cloud, crea un IAM rol y luego dale a tu EC2 instancia de Amazon acceso a ese rol. Para obtener más información, consulta IAMRoles para Amazon EC2 en la Guía del EC2 usuario de Amazon.

Crear un rol de IAM

Es probable que la AWS SDK aplicación que desarrolle acceda al menos a una Servicio de AWS para realizar acciones. Crea un IAM rol que conceda los permisos necesarios para que la aplicación se ejecute.

Este procedimiento crea un rol que concede acceso de solo lectura a Amazon S3, por ejemplo. Muchas de las AWS SDK guías incluyen tutoriales de introducción que se leen en Amazon S3.

1. Inicie sesión en AWS Management Console y abra la IAM consola en https://console.aws.amazon.com/iam/.
2. En el panel de navegación, seleccione Roles y después Crear rol.
3. Para Seleccionar entidad de confianza, en Tipo de entidad de confianza, elija Servicio de AWS.
4. En Caso de uso, selecciona Amazon EC2 y, a continuación, selecciona Siguiente.
5. En Añadir permisos, seleccione la casilla de verificación Acceso de solo lectura a Amazon S3 en la lista de políticas y, a continuación, seleccione Siguiente.
6. Ingrese un nombre para el rol y, a continuación, seleccione Crear rol. Recuerda este nombre porque lo necesitarás cuando crees tu EC2 instancia de Amazon.

Lanza una EC2 instancia de Amazon y especifica tu IAM rol

Puedes crear y lanzar una EC2 instancia de Amazon con tu IAM rol de la siguiente manera:

- Siga Lanzar rápidamente una instancia en la Guía del EC2 usuario de Amazon. Sin embargo, antes del paso final de envío, haga también lo siguiente:
  - En Detalles avanzados, en Perfil de IAM instancia, elija el rol que creó en el paso anterior.

Con esta EC2 configuración IAM y con Amazon, puede implementar su aplicación en la EC2 instancia de Amazon y su aplicación tendrá acceso de lectura al servicio Amazon S3.

Conectarse a la EC2 instancia

Conéctate a la EC2 instancia de Amazon para poder transferirle la aplicación y, a continuación, ejecutarla. Necesitará el archivo que contiene la parte privada del par de claves que utilizó en Par de claves (inicio de sesión) cuando creó la instancia; es decir, el PEM archivo.

Para ello, sigue las instrucciones correspondientes a tu tipo de instancia: Connect to your Linux or Connect to your Windows instance. Cuando se conecte, hágalo de forma que pueda transferir archivos desde el equipo de desarrollo a la instancia.

> **Note**
>
> En un terminal Linux o macOS, puede utilizar el comando secure copy para copiar la aplicación. Para usar `scp` con un key pair, puede usar el siguiente comando:`scp -i path/to/key file/to/copy ec2-user@ec2-xx-xx-xxx-xxx.compute.amazonaws.com:~`.
> Para obtener más información sobre Windows, consulte Transferir archivos a instancias de Windows.

Si utilizas un AWS kit de herramientas, a menudo también puedes conectarte a la instancia mediante el kit de herramientas. Para más información, consulte la guía de usuario específica del kit de herramientas que utilice.

Ejecuta la aplicación en la instancia EC2

1. Copia los archivos de tu aplicación desde tu unidad local a tu EC2 instancia de Amazon.
2. Inicie la aplicación y compruebe que se ejecuta con los mismos resultados que en el equipo de desarrollo.
3. (Opcional) Compruebe que la aplicación utilice las credenciales proporcionadas por el IAM rol.
    a. Inicia sesión en la EC2 consola de Amazon AWS Management Console y ábrela en https://console.aws.amazon.com/ec2/.
    b. Seleccione la instancia.
    c. Selecciona Acciones, Seguridad y, a continuación, selecciona Modificar IAM rol.

d. Para IAMel rol, separe el IAM rol seleccionando Sin IAM rol.

e. Seleccione Actualizar IAM rol.

f. Vuelva a ejecutar la aplicación y confirme que devuelve un error de autorización.

Referencia de configuración

SDKsproporcionar un idioma específico para APIs Servicios de AWS. Se encargan de algunas de las tareas pesadas necesarias para realizar correctamente las API llamadas, como la autenticación, el comportamiento de reintento y mucho más. Para ello, SDKs disponen de estrategias flexibles para obtener credenciales para utilizarlas en sus solicitudes, mantener la configuración que se utilizará con cada servicio y obtener valores para utilizarlos en la configuración global.

Puede encontrar información detallada sobre los ajustes de configuración en las siguientes secciones:

- AWS SDKsy Herramientas: proveedores de credenciales estandarizadas— Los proveedores de credenciales más comunes están estandarizados en variosSDKs.
- AWS SDKsy funciones estandarizadas de Tools— Características comunes estandarizadas en variosSDKs.

Cómo crear clientes de servicio

Para acceder mediante programación Servicios de AWS, SDKs utilice una clase/objeto de cliente para cada Servicio de AWS. Por ejemplo, si tu aplicación necesita acceder a AmazonEC2, crea un objeto EC2 cliente de Amazon para interactuar con ese servicio. A continuación, utiliza el cliente del servicio para realizar solicitudes a ese servicio Servicio de AWS. En la mayoría de los casosSDKs, un objeto de cliente de servicio es inmutable, por lo que debe crear un cliente nuevo para cada servicio al que realice solicitudes y para realizar solicitudes al mismo servicio con una configuración diferente.

Prioridad de los ajustes

La configuración global configura las funciones, los proveedores de credenciales y otras funcionalidades compatibles con la mayoría SDKs y que tienen un amplio impacto en todo el mundo Servicios de AWS. Todos SDKs tienen una serie de lugares (o fuentes) que comprueban para encontrar un valor para la configuración global. La siguiente es la configuración de la prioridad de búsqueda:

1. Cualquier configuración explícita establecida en el código o en el propio cliente de servicio tiene prioridad sobre cualquier otra.

- Algunos ajustes se pueden establecer por operación y se pueden cambiar según sea necesario para cada operación que se invoque. Para el registro AWS CLI o AWS Tools for PowerShell, adoptan la forma de parámetros por operación que se introducen en la línea de comandos. En el caso de unaSDK, las asignaciones explícitas pueden adoptar la forma de un parámetro que se establece al crear una instancia Servicio de AWS cliente o objeto de configuración o, a veces, cuando llamas a una persona. API

2. Solo en Java/Kotlin: la propiedad del JVM sistema correspondiente a la configuración está marcada. Si se ha establecido, se usa ese valor para configurar el cliente.
3. Se comprueba la variable de entorno . Si se ha establecido, se usa ese valor para configurar el cliente.
4. SDKComprueba la configuración en el `credentials` archivo compartido. Si está configurado, el cliente lo usa.
5. El `config` archivo compartido de la configuración. Si la configuración está presente, la SDK usa.
   - La variable de `AWS_PROFILE` entorno o la propiedad `aws.profile` JVM del sistema se pueden utilizar para especificar qué perfil se va a SDK cargar.
6. Los valores por defecto proporcionados por el propio código SDK fuente se utilizan en último lugar.

> **Note**
>
> Es posible que algunas SDKs herramientas se comprueben en un orden diferente. Además, algunas SDKs herramientas admiten otros métodos de almacenamiento y recuperación de parámetros. Por ejemplo, el AWS SDK for .NET admite una fuente adicional llamada SDKStore. Para obtener más información sobre los proveedores exclusivos de una herramienta SDK o herramienta, consulte la guía específica de la herramienta SDK o herramienta que esté utilizando.

El orden determina qué métodos tienen prioridad y sustituyen a los demás. Por ejemplo, si configuras un perfil en el `config` archivo compartido, solo se encuentra y se usa después de que la herramienta SDK o compruebe primero los demás lugares. Esto significa que si colocas una configuración en el archivo `credentials`, se utilizará en lugar de la que se encuentra en el archivo `config`. Si configura una variable de entorno con una configuración y un valor, anulará esa configuración en los archivos `credentials` y `config`. Y, por último, una configuración para

la operación individual (AWS CLI parámetro de línea de comandos (o API parámetro) o en código anularía todos los demás valores de ese comando.

Páginas de configuración

Las páginas de la sección de referencia de configuración de esta guía detallan las configuraciones disponibles que se pueden configurar mediante varios mecanismos. En las tablas siguientes se enumeran los ajustes de los archivos de configuración y credenciales, las variables de entorno y (en el caso de Java y KotlinSDKs) los JVM ajustes que se pueden utilizar fuera del código para configurar la función. Cada tema vinculado de cada lista te lleva a la página de configuración correspondiente.

- Lista de ajustes de archivos `Config`
- Lista de ajustes de archivos `Credentials`
- Lista de variables de entorno
- JVMlista de propiedades del sistema

Cada función o proveedor de credenciales tiene una página en la que se enumeran los ajustes que se utilizan para configurar esa funcionalidad. Para cada configuración, normalmente puedes establecer el valor añadiendo la configuración a un archivo de configuración, estableciendo una variable de entorno o (solo para Java y Kotlin) configurando una propiedad JVM del sistema. Cada configuración muestra todos los métodos admitidos para establecer el valor en un bloque situado encima de los detalles de la descripción. Aunque la prioridad varía, la funcionalidad resultante es la misma independientemente de cómo se establezca.

La descripción incluirá el valor predeterminado, si lo hay, que surtirá efecto si no se hace nada. También define qué valor es válido para esa configuración.

Por ejemplo, veamos una configuración de la página de Compresión de solicitudes características.

La información de la configuración de `disable_request_compression` ejemplo comunica lo siguiente:

- Hay tres formas equivalentes de controlar la compresión de las solicitudes fuera del código base. Puede:
  - Configúralo en tu archivo de configuración usando `disable_request_compression`

- Configúralo como una variable de entorno usando `AWS_DISABLE_REQUEST_COMPRESSION`
- O bien, si está utilizando Java o KotlinSDK, configúrelo como una propiedad JVM del sistema mediante `aws.disableRequestCompression`

> **Note**
>
> También puede haber una forma de configurar la misma funcionalidad directamente en el código, pero esta referencia no cubre esta cuestión, ya que es única para cada uno SDK de ellos. Si quieres establecer tu configuración en el propio código, consulta tu SDK guía o API referencia específica.

- Si no hace nada, el valor predeterminado será`false`.
- Los únicos valores válidos para esta configuración booleana son `true` y. `false`

En la parte inferior de cada página de características hay una sección sobre la compatibilidad con AWS SDKstabla.

En esta tabla se muestra si SDK admite la configuración que aparece en la página. La `Supported` columna indica el nivel de soporte con los siguientes valores:

- `Yes`— La configuración es totalmente compatible con SDK lo que está escrito.
- `Partial`— Algunos de los ajustes son compatibles o el comportamiento se aparta de la descripción. `Partial`En efecto, una nota adicional indica la desviación.
- No— No se admite ninguno de los ajustes. Esto no indica si se podría lograr la misma funcionalidad en el código; solo indica que los ajustes de configuración externos enumerados no son compatibles.

Lista de ajustes de archivos **`Config`**

Los ajustes que se enumeran en la siguiente tabla se pueden asignar en el archivo compartido AWS `config`archivo. Son globales y afectan a todos Servicios de AWS. SDKsy las herramientas también pueden admitir configuraciones y variables de entorno únicas. Para ver los ajustes y las variables de entorno que solo admite una persona SDK o una herramienta, consulta esa guía específica SDK o de herramientas.

Nombre del conjunto	Detalles
account_id_endpoint_mode	Puntos finales basados en cuentas
api_versions	Ajustes de configuración general
aws_access_key_id	AWS claves de acceso
aws_account_id	puntos finales basados en cuentas
aws_secret_access_key	AWS claves de acceso
aws_session_token	AWS claves de acceso
ca_bundle	Ajustes de configuración general
credential_process	Proveedor de credenciales de proceso
credential_source	Asumir el rol de proveedor de credenciales
defaults_mode	Valores predeterminados de configuración inteligente
disable_request_compression	Compresión de solicitudes
duration_seconds	Asumir el rol de proveedor de credenciales

Nombre del conjunto	Detalles
`ec2_metadata_service_endpoint`	IMDSproveedor de credenciales
`ec2_metadata_service_endpoint_mode`	IMDSproveedor de credenciales
`ec2_metadata_v1_disabled`	IMDSproveedor de credenciales
`endpoint_discovery_enabled`	Detección de puntos de conexión
`endpoint_url`	Puntos de conexión específicos del servicio
`external_id`	Asumir el rol de proveedor de credenciales
`ignore_configured_endpoint_urls`	Puntos de conexión específicos del servicio
`max_attempts`	Comportamiento de los reintentos
`metadata_service_num_attempts`	Metadatos de EC2 instancias de Amazon
`metadata_service_timeout`	Metadatos de EC2 instancias de Amazon
`mfa_serial`	Asumir el rol de proveedor de credenciales

Nombre del conjunto	Detalles
`output`	Ajustes de configuración general
`parameter_validation`	Ajustes de configuración general
`region`	Región de AWS
`request_min_compression_size_bytes`	Compresión de solicitudes
`retry_mode`	Comportamiento de los reintentos
`role_arn`	Asumir el rol de proveedor de credenciales
`role_session_name`	Asumir el rol de proveedor de credenciales
`s3_disable_multiregion_access_points`	Puntos de acceso multirregión de Amazon S3
`s3_use_arn_region`	Puntos de acceso de Amazon S3
`sdk_ua_app_id`	Application ID
`source_profile`	Asumir el rol de proveedor de credenciales
`sso_account_id`	IAMProveedor de credenciales de Identity Center
`sso_region`	IAMProveedor de credenciales de Identity Center

Nombre del conjunto	Detalles
`sso_regis tration_scopes`	IAMProveedor de credenciales de Identity Center
`sso_role_name`	IAMProveedor de credenciales de Identity Center
`sso_start_url`	IAMProveedor de credenciales de Identity Center
`sts_regio nal_endpoints`	AWS STS Puntos de conexión regionales
`use_duals tack_endpoint`	Pila doble y puntos finales FIPS
`use_fips_ endpoint`	Pila doble y puntos finales FIPS
`web_ident ity_token_file`	Asumir el rol de proveedor de credenciales

Lista de ajustes de archivos **`Credentials`**

Los ajustes que se enumeran en la siguiente tabla se pueden asignar en el archivo compartido AWS `credentials`archivo. Son globales y afectan a todos Servicios de AWS. SDKsy las herramientas también pueden admitir configuraciones y variables de entorno únicas. Para ver los ajustes y las variables de entorno que solo admite una persona SDK o una herramienta, consulta esa guía específica SDK o de herramientas.

Nombre del conjunto	Detalles
`aws_acces s_key_id`	AWS teclas de acceso
`aws_secre t_access_key`	AWS claves de acceso

Nombre del conjunto	Detalles	
`aws_sessi on_token`	AWS claves de acceso	

Lista de variables de entorno

Las variables de entorno compatibles con la mayoría SDKs se muestran en la siguiente tabla. Son globales y afectan a todos Servicios de AWS. SDKsy las herramientas también pueden admitir configuraciones y variables de entorno únicas. Para ver los ajustes y las variables de entorno que solo admite una persona SDK o una herramienta, consulta esa guía específica SDK o de herramientas.

Nombre del conjunto	Detalles	
`AWS_ACCES S_KEY_ID`	AWS teclas de acceso	
`AWS_ACCOUNT_ID`	puntos finales basados en cuentas	
`AWS_ACCOU NT_ID_END POINT_MODE`	Puntos de enlace basados en cuentas	
`AWS_CA_BUNDLE`	Ajustes de configuración general	
`AWS_CONFIG_FILE`	Ubicación de los archivos compartidos `config` y `credentials`	
`AWS_CONTA INER_AUTH ORIZATION _TOKEN`	Proveedor de credenciales de contenedor	
`AWS_CONTA INER_AUTH`	Proveedor de credenciales de contenedor	

Nombre del conjunto	Detalles
ORIZATION _TOKEN_FILE	
AWS_CONTA INER_CRED ENTIALS_F ULL_URI	Proveedor de credenciales de contenedor
AWS_CONTA INER_CRED ENTIALS_R ELATIVE_URI	Proveedor de credenciales de contenedor
AWS_DEFAU LTS_MODE	Valores predeterminados de configuración inteligente
AWS_DISAB LE_REQUES T_COMPRESSION	Compresión de solicitudes
AWS_EC2_M ETADATA_D ISABLED	IMDSproveedor de credenciales
AWS_EC2_M ETADATA_S ERVICE_EN DPOINT	IMDSproveedor de credenciales
AWS_EC2_M ETADATA_S ERVICE_EN DPOINT_MODE	IMDSproveedor de credenciales
AWS_EC2_M ETADATA_V 1_DISABLED	IMDSproveedor de credenciales

Nombre del conjunto	Detalles
AWS_ENABLE_ENDPOINT_DISCOVERY	Detección de puntos de conexión
AWS_ENDPOINT_URL	Puntos de conexión específicos del servicio
AWS_ENDPOINT_URL_<SERVICE>	Puntos de conexión específicos del servicio
AWS_IAM_ROLE_ARN	Asumir el rol de proveedor de credenciales
AWS_IAM_ROLE_SESSION_NAME	Asumir el rol de proveedor de credenciales
AWS_IGNORE_CONFIGURED_ENDPOINT_URLS	Puntos de conexión específicos del servicio
AWS_MAX_ATTEMPTS	Comportamiento de los reintentos
AWS_METADATA_SERVICE_NUM_ATTEMPTS	Metadatos de EC2 instancias de Amazon
AWS_METADATA_SERVICE_TIMEOUT	Metadatos de EC2 instancias de Amazon
AWS_PROFILE	Archivos compartidos `config` y `credentials`

Nombre del conjunto	Detalles
AWS_REGION	Región de AWS
AWS_REQUEST_MIN_COMPRESSION_SIZE_BYTES	Compresión de solicitudes
AWS_RETRY_MODE	Comportamiento de los reintentos
AWS_S3_DISABLE_MULTIREGION_ACCESS_POINTS	Puntos de acceso multirregión de Amazon S3
AWS_S3_USE_ARN_REGION	Puntos de acceso de Amazon S3
AWS_SDK_UA_APP_ID	Application ID
AWS_SECRET_ACCESS_KEY	AWS claves de acceso
AWS_SESSION_TOKEN	AWS claves de acceso
AWS_SHARED_CREDENTIALS_FILE	Ubicación de los archivos compartidos `config` y `credentials`
AWS_STS_REGIONAL_ENDPOINTS	AWS STS Puntos de conexión regionales
AWS_USE_DUALSTACK_ENDPOINT	Pila doble y puntos finales FIPS

Nombre del conjunto	Detalles
`AWS_USE_FIPS_ENDPOINT`	Pila doble y puntos finales FIPS
`AWS_WEB_IDENTITY_TOKEN_FILE`	Asumir el rol de proveedor de credenciales

JVMlista de propiedades del sistema

Puede utilizar las siguientes propiedades JVM del sistema para AWS SDK for Java y el AWS SDK para Kotlin (dirigido a losJVM). Consulte the section called “¿Cómo configurar las propiedades del sistema JVM” para obtener instrucciones sobre cómo configurar las propiedades JVM del sistema.

Nombre del conjunto	Detalles
`aws.accessKeyId`	AWS claves de acceso
`aws.accountId`	puntos finales basados en cuentas
`aws.accountIdEndpointMode`	Puntos de enlace basados en cuentas
`aws.configFile`	Ubicación de los archivos compartidos `config` y `credentials`
`aws.defaultsMode`	Valores predeterminados de configuración inteligente
`aws.disableEc2MetadataV1`	IMDSproveedor de credenciales
`aws.disableRequestCompression`	Compresión de solicitudes

Nombre del conjunto	Detalles	
`aws.ec2MetadataServiceEndpoint`	IMDSproveedor de credenciales	
`aws.ec2MetadataServiceEndpointMode`	IMDSproveedor de credenciales	
`aws.endpointDiscoveryEnabled`	Detección de puntos de conexión	
`aws.endpointUrl`	Puntos de conexión específicos del servicio	
`aws.endpointUrl<ServiceName>`	Puntos de conexión específicos del servicio	
`aws.ignoreConfiguredEndpointUrls`	Puntos de conexión específicos del servicio	
`aws.maxAttempts`	Comportamiento de los reintentos	
`aws.profile`	Archivos compartidos `config` y `credentials`	
`aws.region`	Región de AWS	
`aws.requestMinCompressionSizeBytes`	Compresión de solicitudes	
`aws.retryMode`	Comportamiento de los reintentos	

Nombre del conjunto	Detalles
`aws.roleArn`	Asumir el rol de proveedor de credenciales
`aws.roleSessionName`	Asumir el rol de proveedor de credenciales
`aws.s3DisableMultiRegionAccessPoints`	Puntos de acceso multirregión de Amazon S3
`aws.s3UseArnRegion`	Puntos de acceso de Amazon S3
`aws.secretAccessKey`	AWS claves de acceso
`aws.sessionToken`	AWS claves de acceso
`aws.sharedCredentialsFile`	Ubicación de los archivos compartidos `config` y `credentials`
`aws.useDualstackEndpoint`	Pila doble y puntos finales FIPS
`aws.useFipsEndpoint`	Pila doble y puntos finales FIPS
`aws.userAgentAppId`	Application ID
`aws.webIdentityTokenFile`	Asumir el rol de proveedor de credenciales

AWS SDKsy Herramientas: proveedores de credenciales estandarizadas

Muchos proveedores de credenciales se han estandarizado para mantener valores predeterminados consistentes y para que funcionen de la misma manera en muchos de ellos. SDKs Esta coherencia aumenta la productividad y la claridad a la hora de codificar en varios. SDKs Todos los ajustes se pueden anular en el código. Para obtener más información, consulta tus especificaciones SDKAPI.

> **Important**
>
> No todos SDKs apoyan a todos los proveedores, ni siquiera a todos los aspectos de un proveedor.

Temas

Cadena de proveedores de credenciales

Todos SDKs tienen una serie de sitios (o fuentes) que consultan para encontrar credenciales válidas que puedan utilizar para realizar una solicitud a un Servicio de AWS. Una vez encontradas las credenciales válidas, se detiene la búsqueda. Esta búsqueda sistemática se denomina cadena de proveedores de credenciales.

Cuando se utiliza uno de los proveedores de credenciales estandarizados, el AWS SDKsintente siempre renovar las credenciales automáticamente cuando caduquen. La cadena de proveedores de credenciales integrada permite a la aplicación actualizar las credenciales independientemente del proveedor de la cadena que utilice. Para ello, no se necesita ningún código adicional. SDK

Si bien la cadena distinta utilizada por cada uno SDK varía, la mayoría de las veces incluyen fuentes como las siguientes:

Proveedor de credenciales	Descripción
AWS claves de acceso	AWS claves de acceso para un IAM usuario (como`AWS_ACCESS_KEY_ID` , y`AWS_SECRET_ACCESS_KEY`).
Federar con identidad web u OpenID Connect: asumir el rol de proveedor de credenciales	Inicie sesión con un proveedor de identidad externo (IdP) conocido, como Login with Amazon, Facebook, Google o cualquier otro IdP compatible con OpenID Connect ()OIDC. Asuma los permisos de un IAM rol mediante un token de identidad web de AWS Security Token Service (AWS STS).
IAMProveedor de credenciales de Identity Center	Obtenga las credenciales de AWS IAM Identity Center.
Asumir el rol de proveedor de credenciales	Obtenga acceso a otros recursos asumiendo los permisos de un IAM rol. (Recupere las credenciales temporales para un rol y, a continuación, utilícelas).
Proveedor de credenciales de contenedor	Credenciales de Amazon Elastic Container Service (AmazonECS) y Amazon Elastic Kubernetes Service (Amazon). EKS El proveedor de credenciales del contenedor obtiene las credenciales de la aplicación contenerizada del cliente.
Proveedor de credenciales de proceso	Proveedor de credenciales personalizadas. Obtenga sus credenciales de una fuente o proceso externo, incluido IAM Roles Anywhere.
IMDSproveedor de credenciales	Credenciales de perfil de instancia de Amazon Elastic Compute Cloud (AmazonEC2). Asocie un IAM rol a cada una de sus EC2 instancias. Las credenciales temporales de ese rol estarán disponibles para el código que se

Proveedor de credenciales	Descripción
	ejecute en la instancia. Las credenciales se entregan a través del servicio de EC2 metadatos de Amazon.

Para cada paso de la cadena, hay varias formas de asignar valores de configuración. Los valores de configuración que se especifican en el código siempre tienen prioridad. Sin embargo, también los hay Variables de entorno y los Archivos `config` y `credentials` compartidos. Para obtener más información, consulte Prioridad de los ajustes.

AWS claves de acceso

> **Warning**
>
> Para evitar riesgos de seguridad, no utilice a IAM los usuarios para autenticarse cuando desarrolle software diseñado específicamente o trabaje con datos reales. En su lugar, utilice la federación con un proveedor de identidad como AWS IAM Identity Center.

AWS las claves de acceso IAM de un usuario se pueden utilizar como AWS credenciales. La AWS SDKutiliza automáticamente estas AWS credenciales para firmar API las solicitudes dirigidas a AWS, para que sus cargas de trabajo puedan acceder a su AWS recursos y datos de forma segura y cómoda. Se recomienda utilizarlas siempre para `aws_session_token` que las credenciales sean temporales y dejen de ser válidas una vez caducadas. No se recomienda utilizar credenciales de larga duración.

> **Note**
>
> Si AWS no puede actualizar estas credenciales temporales, AWS puede extender la validez de las credenciales para que sus cargas de trabajo no se vean afectadas.

El compartido AWS `credentials`el archivo es la ubicación recomendada para almacenar la información sobre las credenciales, ya que se encuentra de forma segura fuera de los directorios de origen de la aplicación y separado SDK de la configuración específica del `config` archivo compartido.

Para obtener más información AWS credenciales y uso de claves de acceso, consulte AWS las credenciales de seguridad y la administración de las claves de acceso de IAM los usuarios en la Guía del IAM usuario.

Configure esta funcionalidad mediante lo siguiente:

`aws_access_key_id`- compartido AWS **`config`**configuración de archivos, **`aws_access_key_id`**- compartido AWS **`credentials`**configuración de archivos (método recomendado), **`AWS_ACCESS_KEY_ID`**: variable de entorno, **`aws.accessKeyId`**- propiedad JVM del sistema: solo Java/Kotlin

Especifica el AWS clave de acceso utilizada como parte de las credenciales para autenticar al usuario.

`aws_secret_access_key`- compartido AWS **`config`**configuración de archivos, **`aws_secret_access_key`**- compartido AWS **`credentials`**configuración de archivos (método recomendado), **`AWS_SECRET_ACCESS_KEY`**: variable de entorno, **`aws.secretAccessKey`**- propiedad JVM del sistema: solo Java/Kotlin

Especifica el AWS clave secreta utilizada como parte de las credenciales para autenticar al usuario.

`aws_session_token`- compartido AWS **`config`**configuración de archivos, **`aws_session_token`**- compartido AWS **`credentials`**configuración de archivos (método recomendado), **`AWS_SESSION_TOKEN`**: variable de entorno, **`aws.sessionToken`**- propiedad JVM del sistema: solo Java/Kotlin

Especifica un AWS un token de sesión utilizado como parte de las credenciales para autenticar al usuario. Este valor se recibe como parte de las credenciales temporales devueltas por las solicitudes aprobadas para asumir un rol. Un token de sesión solo es necesario si especifica manualmente credenciales de seguridad temporales. Sin embargo, le recomendamos que utilice siempre credenciales de seguridad temporales en lugar de credenciales. Para obtener recomendaciones de seguridad, consulte las prácticas recomendadas de seguridad en IAM.

Para obtener instrucciones acerca de cómo obtener estos valores, consulte Autenticación mediante credenciales a corto plazo.

Ejemplo de configuración de este valor en el archivo `config` o `credentials`:

```
[default]
```

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

Ejemplo de configuración de variables de entorno en Linux/macOS mediante la línea de comandos:

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

Ejemplo de configuración de variables de entorno en Windows mediante la línea de comandos:

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

Compatibilidad con AWS SDKs

Las siguientes opciones SDKs son compatibles con las funciones y configuraciones descritas en este tema. Se anotan todas las excepciones parciales. Cualquier configuración de propiedades del JVM sistema es compatible con la AWS SDK for Java y el AWS SDK para Kotlin únicamente.

SDK	Ce	Notas o más información
AWS CLI v2	Sí	
SDKpara C++	Sí	No se admite el archivo compartido `config`.
SDKpara Go V2 (1.x)	Sí	
SDKpara Go 1.x (V1)	Sí	Para usar la configuración de archivos compartidos `config`, debe activar la carga desde el archivo de configuración; consulte Sesiones.
SDKpara Java 2.x	Sí	
SDKpara Java 1.x	Sí	

SDK	Ce	Notas o más información
SDKpara JavaScript 3.x	Sí	
SDKpara JavaScript 2.x	Sí	
SDKpara Kotlin	Sí	
SDKpara. NET3.x	Sí	No se admiten variables de entorno.
SDKpara PHP 3.x	Sí	
SDKpara Python (Boto3)	Sí	
SDKpara Ruby 3.x	Sí	
SDKpara Rust	Sí	
SDKpara Swift	Sí	
Herramientas para PowerShell	Sí	No se admiten variables de entorno.

Asumir el rol de proveedor de credenciales

Asumir un rol implica usar un conjunto de credenciales de seguridad temporales para acceder AWS recursos a los que de otro modo no tendría acceso. Las credenciales temporales incluyen un ID de clave de acceso, una clave de acceso secreta y un token de seguridad.

Para configurar su herramienta SDK o su herramienta para que asuma un rol, primero debe crear o identificar el rol específico que desee asumir. IAMlos roles se identifican de forma única mediante un rol: Amazon Resource Name (ARN). Los roles establecen relaciones de confianza con otra entidad. La entidad de confianza que usa el rol puede ser una Servicio de AWS, otro Cuenta de AWS, un proveedor de identidad web o OIDC una SAML federación.

Una vez identificado el IAM rol, si ese rol confía en usted, puede configurar su SDK herramienta para que utilice los permisos que otorga el rol. Para ello, utilice los siguientes comandos.

Para comenzar a utilizar esta configuración, consulte Asumir un rol en esta guía.

Asumir la configuración del proveedor de credenciales de rol

Configure esta funcionalidad mediante lo siguiente:

`credential_source`- compartido AWS **`config`**configuración de archivos

Se usa en EC2 las instancias de Amazon o en los contenedores de Amazon Elastic Container Service para especificar dónde puede encontrar la herramienta las credenciales que tienen permiso para asumir la función que especificas con el `role_arn` parámetro. SDK

Valor predeterminado: ninguno.

Valores válidos:

- Entorno: especifica que la herramienta SDK o herramienta debe recuperar las credenciales de origen de las variables de entorno `AWS_ACCESS_KEY_ID`y `AWS_SECRET_ACCESS_KEY`.
- Ec2 InstanceMetadata: especifica que la herramienta SDK o herramienta debe utilizar el IAMrol adjunto al perfil de la EC2 instancia para obtener las credenciales de origen.
- EcsContainer— Especifica que la herramienta SDK o herramienta debe utilizar la IAMfunción asociada al ECS contenedor para obtener las credenciales de origen.

No puede especificar `credential_source` y `source_profile` en el mismo perfil.

Ejemplo de configuración de esto en un `config` archivo para indicar que las credenciales deben proceder de AmazonEC2:

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

`duration_seconds`- compartido AWS **`config`**configuración de archivos

Especifica la duración máxima de la sesión de rol, en segundos.

Esta configuración solo se aplica cuando el perfil especifica que se asume un rol.

Valor predeterminado: 3600 segundos (una hora)

Valores válidos: Este valor puede oscilar entre 900 segundos (15 minutos) y el valor de la duración máxima de la sesión para el rol (que puede ser 43 200 segundos como máximo, o 12 horas). Para obtener más información, consulte Ver la configuración de duración máxima de sesión para un rol en la Guía del IAM usuario.

Ejemplo de esta configuración en un archivo `config`:

```
duration_seconds = 43200
```

`external_id`- compartido AWS **`config`**configuración de archivos

Especifica un identificador único utilizado por terceros para adoptar un rol en las cuentas de los clientes.

Esta configuración solo se aplica cuando el perfil especifica asumir un rol y la política de confianza del rol requiere un valor para `ExternalId`. El valor se asigna al parámetro `ExternalId` que se pasa a la operación `AssumeRole` cuando el perfil especifica un rol.

Valor predeterminado: ninguno.

Valores válidos: consulte Cómo utilizar un identificador externo al conceder acceso a su AWS Recursos para terceros en la Guía del IAM usuario.

Ejemplo de esta configuración en un archivo `config`:

```
external_id = unique_value_assigned_by_3rd_party
```

`mfa_serial`- compartido AWS **`config`**configuración de archivos

Especifica la identificación o el número de serie de un dispositivo de autenticación multifactorial (MFA) que el usuario debe usar al asumir una función.

Se requiere cuando se asume un rol en el que la política de confianza para ese rol incluye una condición que requiere la MFA autenticación.

Valor predeterminado: ninguno.

Valores válidos: el valor puede ser un número de serie para un dispositivo de hardware (por ejemploGAHT12345678) o un nombre de recurso de Amazon (ARN) para un MFA dispositivo virtual. Para obtener más información al respectoMFA, consulte Configuración del API acceso MFA protegido en la Guía del IAM usuario.

Ejemplo de esta configuración en un archivo `config`:

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

`role_arn`- compartido AWS **`config`**configuración de archivos, **`AWS_IAM_ROLE_ARN`**: variable de entorno, **`aws.roleArn`**- propiedad JVM del sistema: solo en Java/Kotlin

Especifica el nombre de recurso de Amazon (ARN) de un IAM rol que desea usar para realizar las operaciones solicitadas con este perfil.

Valor predeterminado: ninguno.

Valores válidos: el valor debe ser el ARN de un IAM rol y tener el siguiente formato: `arn:aws:iam::`*`account-id`*`:role/`*`role-name`*

Además, también debe especificar una de las siguientes configuraciones:

- `source_profile`: para identificar otro perfil y usarlo para buscar las credenciales que tengan permiso para asumir el rol en este perfil.
- `credential_source`— Utilizar las credenciales identificadas por las variables de entorno actuales o las credenciales adjuntas a un perfil de EC2 instancia de Amazon o a una instancia de ECS contenedor de Amazon.
- `web_identity_token_file`— Utilizar proveedores de identidad públicos o cualquier proveedor de identidad compatible con OpenID Connect (OIDC) para los usuarios que se hayan autenticado en una aplicación móvil o web.

`role_session_name`- compartido AWS **`config`**configuración de archivos, **`AWS_IAM_ROLE_SESSION_NAME`**: variable de entorno, **`aws.roleSessionName`**- propiedad JVM del sistema: solo en Java/Kotlin

Especifica el nombre que se va a asociar a la sesión de rol. Este nombre aparece en AWS CloudTrail registra las entradas asociadas a esta sesión, lo que puede resultar útil a la hora de auditar.

Valor predeterminado: un parámetro opcional. Si no proporciona este valor, se genera automáticamente un nombre de sesión en caso de que el perfil asuma un rol.

Valores válidos: se proporcionan al `RoleSessionName` parámetro cuando AWS CLI o AWS APIllama a la `AssumeRole` operación (o a operaciones como la `AssumeRoleWithWebIdentity` operación) en su nombre. El valor pasa a formar parte del usuario de rol asumido Amazon Resource Name (ARN) que puede consultar y aparece como parte de las entradas de CloudTrail registro de las operaciones invocadas por este perfil.

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

Ejemplo de esta configuración en un archivo `config`:

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

`source_profile`- compartido AWS **`config`**configuración de archivos

Especifica otro perfil cuyas credenciales se utilizan para asumir la función especificada en la configuración `role_arn` del perfil original. Para entender cómo se utilizan los perfiles en el entorno compartido AWS `config`y `credentials` archivos, consulteArchivos compartidos `config` y `credentials`.

Si especifica un perfil que también sea un perfil de asunción de roles, cada rol se asumirá en orden secuencial para resolver completamente las credenciales. Esta cadena se detiene cuando SDK encuentra un perfil con credenciales. El encadenamiento de roles limita tu AWS CLI o AWS APIla sesión de rol a un máximo de una hora y no se puede aumentar. Para obtener más información, consulte los términos y conceptos de las funciones en la Guía del IAM usuario.

Valor predeterminado: ninguno.

Valores válidos: una cadena de texto que consiste en el nombre de un perfil definido en los archivos `config` y `credentials`. También debe especificar un valor para `role_arn` en el perfil actual.

No puede especificar `credential_source` y `source_profile` en el mismo perfil.

Ejemplo de esta configuración en un archivo de configuración:

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

[profile B]
aws_access_key_id=AKIAIOSFODNN7EXAMPLE
aws_secret_access_key=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
aws_session_token=IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZVERYLONGS
```

En el ejemplo anterior, el perfil A utiliza las credenciales en el perfil B. Al especificar que la herramienta SDK o debe usar el A perfil, la herramienta SDK o busca automáticamente las credenciales del B perfil vinculado y las usa para solicitar credenciales temporales para el IAM rol

especificado. Para ello, la herramienta SDK or utiliza la AssumeRole operación sts: en segundo plano. Luego, su código utiliza esas credenciales temporales para acceder AWS recursos. El rol especificado debe tener políticas de IAM permisos adjuntas que permitan ejecutar el código solicitado, como el comando, Servicio de AWS, o API método.

`web_identity_token_file`- compartido AWS **`config`**configuración de archivos, **`AWS_WEB_IDENTITY_TOKEN_FILE`**: variable de entorno, **`aws.webIdentityTokenFile`**- propiedad JVM del sistema: solo en Java/Kotlin

Especifica la ruta a un archivo que contiene un token de acceso de un proveedor OAuth 2.0 compatible o de un proveedor de identidad de OpenID Connect ID.

Esta configuración permite la autenticación mediante proveedores de federaciones de identidades web, como Google, Facebook y Amazon, entre muchos otros. La SDK herramienta para desarrolladores carga el contenido de este archivo y lo pasa como `WebIdentityToken` argumento cuando llama a la `AssumeRoleWithWebIdentity` operación en tu nombre.

Valor predeterminado: ninguno.

Valores válidos: este valor debe ser una ruta y un nombre de archivo. El archivo debe contener un token de acceso OAuth 2.0 o un token de OpenID Connect que le haya proporcionado un proveedor de identidad. Las rutas relativas se consideran relativas al directorio de trabajo del proceso.

Compatibilidad con AWS SDKs

Las siguientes opciones SDKs son compatibles con las funciones y configuraciones descritas en este tema. Se anotan todas las excepciones parciales. Cualquier configuración de propiedades del JVM sistema es compatible con la AWS SDK for Java y el AWS SDK para Kotlin únicamente.

SDK	Ce	Notas o más información
AWS CLI v2	Sí	
SDKpara C++	Parci	`credential_source` no admitido. `duration_seconds` no admitido. `mfa_serial` no admitido.
SDKpara Go V2 (1.x)	Sí	

SDK	Ce	Notas o más información
SDKpara Go 1.x (V1)	Sí	Para usar la configuración de archivos compartidos `config`, debe activar la carga desde el archivo de configuración; consulte Sesiones.
SDKpara Java 2.x	Parci	`mfa_serial` no admitidas. Úselo `AWS_ROLE_ARN` en lugar de. `AWS_IAM_ROLE_ARN` Úselo `AWS_ROLE_SESSION_NAME` en lugar de`AWS_IAM_ROLE_SESSION_NAME` .
SDKpara Java 1.x	Parci	`credential_source` no se admite. `mfa_serial` no se admite. JVMno se admiten las propiedades del sistema.
SDKpara JavaScript 3.x	Sí	
SDKpara JavaScript 2.x	Parci	`credential_source` no admitidas.
SDKpara Kotlin	Sí	Úselo `AWS_ROLE_ARN` en lugar de. `AWS_IAM_ROLE_ARN` Úselo `AWS_ROLE_SESSION_NAME` en lugar de`AWS_IAM_ROLE_SESSION_NAME` .
SDKpara. NET3.x	Sí	
SDKpara PHP 3.x	Sí	
SDKpara Python (Boto3)	Sí	
SDKpara Ruby 3.x	Sí	
SDKpara Rust	Sí	
SDKpara Swift	Sí	
Herramientas para PowerShell	Sí	

Proveedor de credenciales de contenedor

El proveedor de credenciales del contenedor obtiene las credenciales de la aplicación contenerizada del cliente. Este proveedor de credenciales es útil para los clientes de Amazon Elastic Container Service (AmazonECS) y Amazon Elastic Kubernetes Service (Amazon). EKS SDKsintenta cargar las credenciales desde el HTTP punto de enlace especificado mediante una solicitud. GET

Si utilizas AmazonECS, te recomendamos que utilices un IAM rol de tarea para mejorar el aislamiento, la autorización y la auditabilidad de las credenciales. Cuando se configura, Amazon ECS establece la variable de `AWS_CONTAINER_CREDENTIALS_RELATIVE_URI` entorno que utilizan las herramientas SDKs y para obtener las credenciales. Para configurar Amazon ECS para esta funcionalidad, consulte IAMFunción de tarea en la Guía para desarrolladores de Amazon Elastic Container Service.

Si utilizas AmazonEKS, te recomendamos que utilices Amazon EKS Pod Identity para mejorar el aislamiento de credenciales, los privilegios mínimos, la auditabilidad, el funcionamiento independiente, la reutilización y la escalabilidad. Tanto tu pod como un IAM rol están asociados a una cuenta de servicio de Kubernetes para administrar las credenciales de tus aplicaciones. Para obtener más información sobre Amazon EKS Pod Identity, consulta Amazon EKS Pod Identities en la Guía del EKS usuario de Amazon. Cuando se configura, Amazon EKS establece las variables de `AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE` entorno `AWS_CONTAINER_CREDENTIALS_FULL_URI` y las variables que utilizan las herramientas SDKs y para obtener las credenciales. Para obtener información sobre la configuración, consulta Cómo configurar el Amazon EKS Pod Identity Agent en la Guía del EKS usuario de Amazon o Amazon EKS Pod Identity simplifica IAM los permisos de las aplicaciones en los EKS clústeres de Amazon en AWS Sitio web del blog.

Configure esta funcionalidad mediante lo siguiente:

`AWS_CONTAINER_CREDENTIALS_FULL_URI`: variable de entorno

Especifica el HTTP URL punto final completo que se debe utilizar SDK al realizar una solicitud de credenciales. Esto incluye tanto el esquema como el host.

Valor predeterminado: ninguno.

Valores válidos: válidosURI.

Nota: Esta configuración es una alternativa a *`AWS_CONTAINER_CREDENTIALS_RELATIVE_URI`* y solo se usará si *`AWS_CONTAINER_CREDENTIALS_RELATIVE_URI`* no está establecida.

Ejemplo de configuración de variables de entorno en Linux/macOS mediante la línea de comandos:

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

o

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

`AWS_CONTAINER_CREDENTIALS_RELATIVE_URI`: variable de entorno

Especifica el HTTP URL punto final relativo que se debe utilizar SDK al realizar una solicitud de credenciales. El valor se añade al ECS nombre de host predeterminado de Amazon de. `169.254.170.2`

Valor predeterminado: ninguno.

Valores válidos: relativo válidoURI.

Ejemplo de configuración de variables de entorno en Linux/macOS mediante la línea de comandos:

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

`AWS_CONTAINER_AUTHORIZATION_TOKEN`: variable de entorno

Especifica un token de autorización en texto sin formato. Si se establece esta variable, se SDK establecerá el encabezado de autorización de la HTTP solicitud con el valor de la variable de entorno.

Valor predeterminado: ninguno.

Valores válidos: Cadena.

Nota: Esta configuración es una alternativa a *`AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE`* y solo se usará si *`AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE`* no está establecida.

Ejemplo de configuración de variables de entorno en Linux/macOS mediante la línea de comandos:

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

`AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE`: variable de entorno

Especifica una ruta de archivo absoluta a un archivo que contiene el token de autorización en texto sin formato.

Valor predeterminado: ninguno.

Valores válidos: Cadena.

Ejemplo de configuración de variables de entorno en Linux/macOS mediante la línea de comandos:

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

Compatibilidad con AWS SDKs

Las siguientes opciones SDKs son compatibles con las funciones y configuraciones descritas en este tema. Se anotan todas las excepciones parciales. Cualquier configuración de propiedades del JVM sistema es compatible con la AWS SDK for Java y el AWS SDK para Kotlin únicamente.

SDK	Ce	Notas o más información
AWS CLI v2	Sí	
SDKpara C++	Sí	
SDKpara Go V2 (1.x)	Sí	
SDKpara Go 1.x (V1)	Sí	
SDKpara Java 2.x	Sí	`AWS_CONTAINER_CREDENTIALS_FULL_URI` y también `AWS_CONTAINER_AUTHORIZATION_TOKEN` se utilizan para Lambda SnapStart para Java.
SDKpara Java 1.x	Sí	`AWS_CONTAINER_CREDENTIALS_FULL_URI` y también `AWS_CONTAINER_AUTHORIZATION_TOKEN` se utilizan para Lambda SnapStart para Java.

SDK	Ce	Notas o más información
SDKpara JavaScript 3.x	Sí	
SDKpara JavaScript 2.x	Sí	
SDKpara Kotlin	Sí	
SDKpara. NET3.x	Sí	
SDKpara PHP 3.x	Sí	
SDKpara Python (Boto3)	Sí	
SDKpara Ruby 3.x	Sí	
SDKpara Rust	Sí	
SDKpara Swift	No	
Herramientas para PowerShell	Sí	

IAMProveedor de credenciales de Identity Center

Este mecanismo de autenticación utiliza AWS IAM Identity Center para obtener acceso mediante un inicio de sesión único (SSO) a Servicios de AWS para tu código.

> **Note**
>
> En el navegador AWS SDKAPIdocumentación, el proveedor de credenciales del Centro de IAM Identidad se denomina proveedor de SSO credenciales.

Después de activar IAM Identity Center, defina un perfil para su configuración en su espacio compartido AWS `config`archivo. Este perfil se utiliza para conectarse al portal de acceso al IAM Identity Center. Cuando un usuario se autentica correctamente en IAM Identity Center, el portal devuelve las credenciales de corta duración para el IAM rol asociado a ese usuario. Para saber

cómo SDK obtiene las credenciales temporales de la configuración y las utiliza para Servicio de AWS solicitudes, consulteComprender la autenticación del Centro de identidades de IAM.

Hay dos formas de configurar IAM Identity Center a través del `config` archivo:

- SSOconfiguración del proveedor de tokens (recomendada): duraciones de sesión prolongadas.
- Configuración antigua que no se puede actualizar: utiliza una sesión fija de ocho horas.

En ambas configuraciones, tendrá que volver a iniciar sesión cuando caduque la sesión.

Para establecer duraciones de sesión personalizadas, debe usar la configuración del proveedor de token. SSO

Las dos guías siguientes contienen información adicional sobre IAM Identity Center:

- AWS IAM Identity Center Guía del usuario
- AWS IAM Identity Center APIReferencia del portal

Requisitos previos

Primero debe habilitar IAM Identity Center. Para obtener más información sobre cómo habilitar la autenticación de IAM Identity Center, consulte Primeros pasos en AWS IAM Identity Center Guía del usuario.

Como alternativa, siga las instrucciones Autenticación del Centro de identidades de IAM de esta guía. Estas instrucciones proporcionan una guía completa, desde la activación de IAM Identity Center hasta la configuración necesaria de los `config` archivos compartidos, que se indica a continuación.

SSOconfiguración del proveedor de tokens

> **Note**
>
> Para utilizar el AWS CLI para crear esta configuración por usted, consulte Configurar su perfil con el `aws configure sso` asistente en AWS CLI.

Cuando utiliza la configuración del proveedor de SSO token, su AWS SDKo la herramienta actualiza automáticamente la sesión hasta el período de sesión extendido. Para obtener más información sobre la duración y la duración máxima de la sesión, consulte Configurar la duración de la sesión del

AWS acceda al portal y a las aplicaciones integradas de IAM Identity Center en el AWS IAM Identity Center Guía del usuario.

La `sso-session` sección del `config` archivo se usa para agrupar las variables de configuración para adquirir los tokens de SSO acceso, que luego se pueden usar para adquirirlos AWS credenciales. Para obtener más información sobre el formato de las secciones de un archivo `config`, consulte Formato del archivo de configuración.

Defina una sección `sso-session` y asóciela a un perfil. `sso_region` y`sso_start_url` deben establecerse en la sección `sso-session`. Normalmente, `sso_account_id` y `sso_role_name` debe configurarse en la `profile` sección para que SDK pueda solicitar AWS credenciales.

> **Note**
>
> Para obtener información detallada sobre cómo las herramientas SDKs y herramientas utilizan y actualizan las credenciales con esta configuración, consulteComprender la autenticación del Centro de identidades de IAM.

El siguiente ejemplo lo configura SDK para solicitar las credenciales del IAM Identity Center. También admite la actualización automática de los tokens.

```
[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
```

Puede reutilizar las configuraciones de `sso-session` en varios perfiles.

```
[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
```

`sso_account_id`y `sso_role_name` no son necesarios para todos los escenarios de configuración de los SSO tokens. Si su aplicación solo usa Servicios de AWS que admitan la autenticación al portador, luego la tradicional AWS no se necesitan credenciales. La autenticación portadora es un esquema de HTTP autenticación que utiliza fichas de seguridad denominadas fichas portadoras. En este escenario, no se necesitan `sso_account_id` ni `sso_role_name`. Consulte la guía individual para su Servicio de AWS para determinar si admite la autorización del token al portador.

Los ámbitos de registro se configuran como parte de un `sso-session`. El alcance es un mecanismo en OAuth 2.0 para limitar el acceso de una aplicación a la cuenta de un usuario. Una solicitud puede pedir uno o varios ámbitos y el token de acceso emitido a la solicitud se limita a los ámbitos concedidos. Estos ámbitos definen los permisos que se solicitan para ser autorizados para el OIDC cliente registrado y los tokens de acceso recuperados por el cliente. Para ver las opciones de alcance de acceso admitidas, consulte Ámbitos de acceso en AWS IAM Identity Center Guía del usuario. El siguiente ejemplo establece `sso_registration_scopes` para proporcionar acceso para enumerar cuentas y roles.

```
[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
```

El token de autenticación se almacena en caché en el disco en el directorio `~/.aws/sso/cache` con un nombre de archivo basado en el nombre de la sesión.

Configuración heredada no actualizable

La actualización automática de tokens no se admite con la configuración no actualizable heredada. Se recomienda utilizar el SSOconfiguración del proveedor de tokens en su lugar.

Para utilizar la configuración heredada no renovable, debe especificar los siguientes parámetros en su perfil:

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

Debe especificar el portal de usuario para un perfil con la configuración de `sso_start_url` y `sso_region`. Los permisos se especifican con la configuración de `sso_account_id` y `sso_role_name`.

En el siguiente ejemplo se definen los cuatro valores obligatorios del archivo `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
```

El token de autenticación se almacena en caché en el disco en el directorio `~/.aws/sso/cache` con un nombre de archivo basado en el `sso_start_url`.

IAMConfiguración del proveedor de credenciales de Identity Center

Configure esta funcionalidad mediante lo siguiente:

`sso_start_url`- compartido AWS **`config`**configuración de archivos

El URL que apunta al portal de acceso al Centro de IAM Identidad de su organización. Para obtener más información sobre el portal de acceso al Centro de IAM Identidad, consulte Uso del AWS acceda al portal en el AWS IAM Identity Center Guía del usuario.

Para encontrar este valor, abra la consola de IAM Identity Center, consulte el panel de control y busque AWS acceda al portal URL.

`sso_region`- compartido AWS **`config`**configuración de archivos

La Región de AWS que contenga el host del portal de IAM Identity Center, es decir, la región que seleccionó antes de activar IAM Identity Center. Esto es independiente de la configuración predeterminada AWS Región y puede ser diferente.

Para obtener una lista completa de Regiones de AWS y sus códigos, consulte los puntos finales regionales en Referencia general de Amazon Web Services. Para encontrar este valor, abra la consola de IAM Identity Center, consulte el panel de control y busque la región.

`sso_account_id`- compartido AWS **`config`**configuración de archivos

El identificador numérico del Cuenta de AWS que se agregó a través del AWS Organizations servicio que se utilizará para la autenticación.

Para ver la lista de cuentas disponibles, vaya a la consola de IAM Identity Center y abra el Cuentas de AWSpágina. También puedes ver la lista de cuentas disponibles mediante el ListAccountsAPImétodo de AWS IAM Identity Center APIReferencia del portal. Por ejemplo, puede llamar al AWS CLI método list-accounts.

`sso_role_name`- compartido AWS **`config`**configuración de archivos

El nombre de un conjunto de permisos aprovisionado como un IAM rol que define los permisos resultantes del usuario. El rol debe existir en Cuenta de AWS especificado por`sso_account_id`. Utilice el nombre del rol, no el nombre de Amazon Resource Name (ARN) del rol.

Los conjuntos de permisos tienen asociadas IAM políticas y políticas de permisos personalizadas y definen el nivel de acceso que los usuarios tienen a los permisos que se les ha asignado Cuentas de AWS.

Para ver la lista de conjuntos de permisos disponibles por Cuenta de AWS, vaya a la consola de IAM Identity Center y abra Cuentas de AWSpágina. Elija el nombre correcto del conjunto de permisos que aparece en la Cuentas de AWS tabla. También puede ver la lista de conjuntos de permisos disponibles mediante el ListAccountRolesAPImétodo de AWS IAM Identity Center APIReferencia del portal. Por ejemplo, puede llamar al AWS CLI método list-account-roles.

`sso_registration_scopes`- compartido AWS **`config`**configuración de archivos

Una lista delimitada por comas de los ámbitos válidos que deben autorizarse para la `sso-session`. Los ámbitos autorizan el acceso a los puntos finales autorizados por el portador del identificador del IAM Identity Center. Se `sso:account:access` debe conceder un alcance mínimo de para que el servicio IAM Identity Center devuelva un token de actualización. Para ver las cadenas de alcance de acceso admitidas, consulte Ámbitos de acceso en AWS IAM Identity Center Guía del usuario. Esta configuración no aplica a la configuración heredada no actualizable. Los tokens emitidos con la configuración heredada tienen un alcance limitado de `sso:account:access` de forma implícita.

Compatibilidad con AWS SDKs

Las siguientes opciones SDKs son compatibles con las funciones y configuraciones descritas en este tema. Se anotan todas las excepciones parciales. Cualquier configuración de propiedades del JVM sistema es compatible con la AWS SDK for Java y el AWS SDK para Kotlin únicamente.

SDK	Ce	Notas o más información
AWS CLI v2	Sí	
SDKpara C++	Sí	
SDKpara Go V2 (1.x)	Sí	
SDKpara Go 1.x (V1)	Sí	Para usar la configuración de archivos compartidos `config`, debe activar la carga desde el archivo de configuración; consulte Sesiones.
SDKpara Java 2.x	Sí	Los valores de configuración también se admiten en el archivo `credentials` .
SDKpara Java 1.x	No	
SDKpara JavaScript 3.x	Sí	
SDKpara JavaScript 2.x	Sí	
SDKpara Kotlin	Sí	
SDKpara. NET3.x	Sí	
SDKpara PHP 3.x	Sí	
SDKpara Python (Boto3)	Sí	
SDKpara Ruby 3.x	Sí	
SDKpara Rust	Parci	Solo configuración heredada no actualizable.
SDKpara Swift	Sí	

SDK	Ce	Notas o más información
Herramientas para PowerShell	Sí	

IMDSproveedor de credenciales

Instance Metadata Service (IMDS) proporciona datos sobre la instancia que puede usar para configurar o administrar la instancia en ejecución. Para obtener más información sobre los datos disponibles, consulta Trabajar con metadatos de instancias en la Guía del EC2 usuario de Amazon. Amazon EC2 proporciona un punto de enlace local disponible para las instancias que puede proporcionar varios bits de información a la instancia. Si la instancia tiene una función asociada, puede proporcionar un conjunto de credenciales válidas para esa función. SDKsPueden usar ese punto final para resolver las credenciales como parte de su cadena de proveedores de credenciales predeterminada. De forma predeterminada, se usa la versión 2 (IMDSv2) del Servicio de Metadatos de Instancia, una versión más segura IMDS que usa un token de sesión. Si se produce un error debido a una condición que no se puede volver a intentar (códigos de HTTP error 403, 404, 405), IMDSv1 se utiliza como alternativa.

Configure esta funcionalidad mediante lo siguiente:

`AWS_EC2_METADATA_DISABLED`: variable de entorno

Si debe o no intentar utilizar Amazon EC2 Instance Metadata Service (IMDS) para obtener credenciales.

Valor predeterminado: `false`.

Valores válidos:

- **`true`**— No lo utilice IMDS para obtener credenciales.
- **`false`**— Se utiliza IMDS para obtener credenciales.

`ec2_metadata_v1_disabled`- compartido AWS **`config`**configuración de archivos, **`AWS_EC2_METADATA_V1_DISABLED`**: variable de entorno, **`aws.disableEc2MetadataV1`**- propiedad JVM del sistema: solo en Java/Kotlin

Si se debe utilizar o no la versión 1 (IMDSv1) del Servicio de Metadatos de Instancia como alternativa en caso de error. IMDSv2

> **Note**
>
> Los nuevos SDKs no admiten esta configuración IMDSv1 y, por lo tanto, no la admiten. Para obtener más información, consulte la tabla Compatibilidad con AWS SDKs.

Valor predeterminado: `false`.

Valores válidos:

- **`true`**— No lo utilices IMDSv1 como alternativa.
- **`false`**— Úselo IMDSv1 como alternativa.

`ec2_metadata_service_endpoint`- compartido AWS **`config`**configuración de archivos, **`AWS_EC2_METADATA_SERVICE_ENDPOINT`**: variable de entorno, **`aws.ec2MetadataServiceEndpoint`**- propiedad JVM del sistema: solo en Java/Kotlin

El punto final de. IMDS

Valor predeterminado: si el `ec2_metadata_service_endpoint_mode` es igual a `IPv4`, el punto de conexión predeterminado es `http://169.254.169.254`. Valor predeterminado: si el `ec2_metadata_service_endpoint_mode` es igual a `IPv6`, el punto de conexión predeterminado es `http://[fd00:ec2::254]`.

Valores válidos: válidosURI.

`ec2_metadata_service_endpoint_mode`- compartido AWS **`config`**configuración de archivos, **`AWS_EC2_METADATA_SERVICE_ENDPOINT_MODE`**: variable de entorno, **`aws.ec2MetadataServiceEndpointMode`**- propiedad JVM del sistema: solo en Java/Kotlin

El modo de punto final de. IMDS

Valor predeterminado:`IPv4`.

Valores válidos: `IPv4`, `IPv6`.

> **Note**
>
> El proveedor de IMDS credenciales forma parte deCadena de proveedores de credenciales. Sin embargo, el proveedor de IMDS credenciales solo se comprueba después de varios otros proveedores de esta serie. Por lo tanto, si desea que su programa utilice las

credenciales de este proveedor, debe eliminar otros proveedores de credenciales válidos de la configuración o utilizar un perfil diferente. Como alternativa, en lugar de confiar en la cadena de proveedores de credenciales para descubrir automáticamente qué proveedor devuelve credenciales válidas, especifique el uso del proveedor de IMDS credenciales en el código. Puede especificar las fuentes de credenciales directamente al crear clientes de servicio.

Seguridad de las credenciales IMDS

De forma predeterminada, cuando el AWS SDKno está configurado con credenciales válidas e SDK intentará utilizar el Amazon EC2 Instance Metadata Service (IMDS) para recuperar las credenciales de un AWS rol. Este comportamiento se puede deshabilitar configurando la variable del entorno de `AWS_EC2_METADATA_DISABLED` en `true`. Esto evita actividades de red innecesarias y mejora la seguridad en redes que no son de confianza en las que se puede suplantar el Amazon EC2 Instance Metadata Service.

> **Note**
>
> AWS SDKlos clientes configurados con credenciales válidas nunca las utilizarán IMDS para recuperar las credenciales, independientemente de cualquiera de estas configuraciones.

Inhabilitar el uso de las credenciales de Amazon EC2 IMDS

La forma de configurar esta variable de entorno depende del sistema operativo que se utilice y de si desea o no que el cambio sea persistente.

Linux y macOS

Los clientes que utilizan Linux o macOS pueden configurar esta variable de entorno con el siguiente comando:

```
$ export AWS_EC2_METADATA_DISABLED=true
```

Si desea que esta configuración se mantenga durante varias sesiones del intérprete de comandos y se reinicie el sistema, puede añadir el comando anterior al archivo de perfil de intérprete de comandos, como `.bash_profile`, `.zsh_profile` o `.profile`.

Windows

Los clientes que utilizan Windows pueden configurar esta variable de entorno con el siguiente comando:

```
$ set AWS_EC2_METADATA_DISABLED=true
```

Si desea que esta configuración sea persistente en varias sesiones de intérprete de comandos y se reinicie el sistema, utilice el siguiente comando en su lugar:

```
$ setx AWS_EC2_METADATA_DISABLED=true
```

> **Note**
>
> El comando setx no aplica el valor a la sesión de shell actual, por lo que tendrá que volver a cargar o volver a abrir el intérprete de comandos para que el cambio surta efecto.

Compatibilidad con AWS SDKs

Las siguientes opciones SDKs son compatibles con las funciones y configuraciones descritas en este tema. Se anotan todas las excepciones parciales. Cualquier configuración de propiedades del JVM sistema es compatible con la AWS SDK for Java y el AWS SDK para Kotlin únicamente.

SDK	Ce	Notas o más información
AWS CLI v2	Sí	
SDKpara C++	Sí	
SDKpara Go V2 (1.x)	Sí	
SDKpara Go 1.x (V1)	Sí	Para usar la configuración de archivos compartidos `config`, debe activar la carga desde el archivo de configuración; consulte Sesiones.
SDKpara Java 2.x	Sí	

SDK	Ce	Notas o más información
SDKpara Java 1.x	Parci	JVMpropiedades del sistema: se usa `com.amazonaws.sdk.disableEc2MetadataV1` en lugar de `aws.disableEc2MetadataV1`; `aws.ec2MetadataServiceEndpoint` y `aws.ec2MetadataServiceEndpointMode` no se admite.
SDKpara JavaScript 3.x	Sí	
SDKpara JavaScript 2.x	Sí	
SDKpara Kotlin	Sí	No utiliza la opción IMDSv1 alternativa.
SDKpara. NET3.x	Sí	
SDKpara PHP 3.x	Sí	
SDKpara Python (Boto3)	Sí	
SDKpara Ruby 3.x	Sí	
SDKpara Rust	Sí	No utiliza el IMDSv1 respaldo.
SDKpara Swift	Sí	
Herramientas para PowerShell	Sí	Puede deshabilitar la opción IMDSv1 alternativa de forma explícita en el código mediante`[Amazon.Util.EC2InstanceMetadata]::EC2MetadataV1Disabled = $true`.

Proveedor de credenciales de proceso

SDKsproporcionan una forma de ampliar la cadena de proveedores de credenciales para casos de uso personalizados. Este proveedor se puede utilizar para proporcionar implementaciones personalizadas, como recuperar credenciales de un almacén de credenciales local o integrarlas con su proveedor de identificación local.

Por ejemplo, IAM Roles Anywhere utiliza `credential_process` para obtener credenciales temporales en nombre de su aplicación. Para configurar `credential_process` para este uso, consulte Funciones de IAM en cualquier lugar.

> **Note**
>
> A continuación se describe un método para obtener credenciales de un proceso externo y se puede utilizar si se ejecuta software fuera de AWS. Si está construyendo sobre una AWS recurso informático, utilice otros proveedores de credenciales. Si usa esta opción, debe asegurarse de que el archivo de configuración esté lo más bloqueado posible siguiendo las mejores prácticas de seguridad para su sistema operativo. Confirme que su herramienta de credenciales personalizada no escriba ninguna información secreta en ella`StdErr`, ya que y SDKs AWS CLI puede capturar y registrar dicha información, lo que podría exponerla a usuarios no autorizados.

Configure esta funcionalidad mediante lo siguiente:

`credential_process`- compartido AWS **`config`**configuración de archivos

Especifica un comando externo que la herramienta SDK o herramienta ejecuta en su nombre para generar o recuperar las credenciales de autenticación que se van a utilizar. La configuración especifica el nombre del programa o comando que se SDK invocará. Cuando SDK invoca el proceso, espera a que el proceso escriba datos en él. JSON `stdout` El proveedor personalizado debe devolver la información en un formato específico. Esa información contiene las credenciales que la herramienta SDK o herramienta puede usar para autenticarte.

> **Note**
>
> El proveedor de credenciales del proceso forma parte del Cadena de proveedores de credenciales. Sin embargo, el proveedor de credenciales del proceso solo se comprueba después de varios otros proveedores de esta serie. Por lo tanto, si desea que su programa utilice las credenciales de este proveedor, debe eliminar otros proveedores de credenciales válidos de la configuración o utilizar un perfil diferente. Como alternativa, en lugar de confiar en la cadena de proveedores de credenciales para descubrir automáticamente qué proveedor devuelve credenciales válidas, especifique el uso del proveedor de credenciales

de proceso en el código. Puede especificar las fuentes de credenciales directamente al crear clientes de servicio.

Especificar la ruta al programa de credenciales

El valor de la configuración es una cadena que contiene la ruta a un programa que la herramienta de desarrollo SDK o la herramienta de desarrollo ejecutan en tu nombre:

- La ruta y el nombre del archivo solo pueden constar de los siguientes caracteres: A-Z, a-z, 0-9, guion (-), guion bajo (_), punto (.), barra oblicua (/), barra diagonal inversa (\ ) y espacio.
- Si la ruta de acceso o el nombre del archivo contienen un espacio, rodee la ruta completa y el nombre del archivo con comillas dobles (" ").
- Si un nombre de parámetro o un valor de parámetro contienen un espacio, rodee ese elemento con comillas dobles (" "). Incluya solo el nombre o el valor, no el par.
- No incluya ninguna variable de entorno en las cadenas. Por ejemplo, no puede incluir `$HOME` ni `%USERPROFILE%`.
- No especifique la carpeta de inicio como ~. * En la solicitud debe especificar la ruta completa o el nombre del archivo base. Si hay un nombre de archivo base, el sistema intentará encontrar el programa en las carpetas especificadas por la variable del entorno `PATH`. La ruta varía en función del sistema operativo:

  El siguiente ejemplo muestra la configuración de credential_process en el archivo `config` compartido en Linux/macOS.

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

  El siguiente ejemplo muestra la configuración de credential_process en el archivo `config` compartido en Windows.

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

- Se puede especificar dentro de un perfil específico:

  ```
  [profile cred_process]
  credential_process = /Users/username/process.sh
  ```

```
region = us-east-1
```

Salida válida del programa de credenciales

SDKEjecuta el comando tal como se especifica en el perfil y, a continuación, lee los datos del flujo de salida estándar. El comando que especifique, ya sea un script o un programa binario, debe generar un JSON resultado STD0UT que coincida con la siguiente sintaxis.

```
{
    "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**
>
> En la fecha de publicación del presente documento, la clave `Version` debe establecerse en 1. Puede aumentar con el paso del tiempo a medida que la estructura evolucione.

La `Expiration` clave es una marca de tiempo RFC3339 formateada. Si la `Expiration` clave no aparece en el resultado de la herramienta, se SDK supone que las credenciales son credenciales de larga duración que no se actualizan. De otro modo, las credenciales se consideran credenciales temporales y se actualizan automáticamente volviendo a ejecutar el comando `credential_process` antes de que caduquen las credenciales.

> **Note**
>
> SDKNo almacena en caché las credenciales de los procesos externos de la misma forma en que lo hace con las credenciales de rol. Si se requiere el almacenamiento en caché, debe implementarlo en el proceso externo.

El proceso externo puede devolver un código de devolución distinto de cero para indicar que se ha producido un error al intentar recuperar las credenciales.

Compatibilidad con AWS SDKs

Las siguientes opciones SDKs son compatibles con las funciones y configuraciones descritas en este tema. Se anotan todas las excepciones parciales. Cualquier configuración de propiedades del JVM sistema es compatible con la AWS SDK for Java y el AWS SDK para Kotlin únicamente.

SDK	Ce	Notas o más información
AWS CLI v2	Sí	
SDKpara C++	Sí	
SDKpara Go V2 (1.x)	Sí	
SDKpara Go 1.x (V1)	Sí	Para usar la configuración de archivos compartidos `config`, debe activar la carga desde el archivo de configuración; consulte Sesiones.
SDKpara Java 2.x	Sí	
SDKpara Java 1.x	Sí	
SDKpara JavaScript 3.x	Sí	
SDKpara JavaScript 2.x	Sí	
SDKpara Kotlin	Sí	
SDKpara. NET3.x	Sí	
SDKpara PHP 3.x	Sí	
SDKpara Python (Boto3)	Sí	
SDKpara Ruby 3.x	Sí	
SDKpara Rust	Sí	
SDKpara Swift	Sí	

SDK	Ce	Notas o más información
Herramientas para PowerShell	Sí	

AWS SDKsy funciones estandarizadas de Tools

Muchas funciones se han estandarizado con valores predeterminados consistentes y funcionan de la misma manera en muchas SDKs de ellas. Esta coherencia aumenta la productividad y la claridad a la hora de codificar en variosSDKs. Todos los ajustes se pueden anular en el código, consulta tus especificaciones SDK API para obtener más información.

> **Important**
>
> No todas son SDKs compatibles con todas las funciones, ni siquiera con todos los aspectos de una función.

Temas

- Puntos finales basados en cuentas
- Application ID
- Metadatos de EC2 instancias de Amazon
- Puntos de acceso de Amazon S3
- Puntos de acceso multirregión de Amazon S3
- Región de AWS
- AWS STS Puntos de conexión regionales
- Pila doble y puntos finales FIPS
- Detección de puntos de conexión
- Ajustes de configuración general
- IMDScliente
- Comportamiento de los reintentos
- Compresión de solicitudes

- Puntos de conexión específicos del servicio
- Valores predeterminados de configuración inteligente

Puntos finales basados en cuentas

Los puntos finales basados en cuentas ayudan a garantizar un alto rendimiento y escalabilidad mediante el uso de Cuenta de AWS ID para agilizar el enrutamiento de Servicio de AWS solicitudes de servicios que admiten esta función. Cuando usas un AWS SDK, el proveedor de credenciales y el servicio que admiten puntos de enlace basados en cuentas, construirá y SDK utilizará automáticamente un punto de enlace basado en cuentas en lugar de un punto de enlace regional.

De forma predeterminada, el ID de la cuenta se recopila cuando se procesa la solicitud y se utiliza para crear un punto final. La resolución de credenciales también se produce cuando se procesa la solicitud y puede cambiar el método de resolución del punto final. Según el proveedor de credenciales que utilice, es posible que el ID de la cuenta tenga un origen diferente.

Configure esta funcionalidad mediante lo siguiente:

`aws_account_id`- compartido AWS **`config`**configuración de archivos, **`AWS_ACCOUNT_ID`**: variable de entorno, **`aws.accountId`**- propiedad JVM del sistema: solo en Java/Kotlin

La Cuenta de AWS ID. Se utiliza para el enrutamiento de puntos finales basado en cuentas. Un registro Cuenta de AWS El ID tiene un formato similar al 111122223333.

El enrutamiento de puntos finales basado en cuentas proporciona un mejor rendimiento de las solicitudes para algunos servicios.

`account_id_endpoint_mode`- compartido AWS **`config`**configuración de archivos, **`AWS_ACCOUNT_ID_ENDPOINT_MODE`**: variable de entorno, **`aws.accountIdEndpointMode`**- propiedad JVM del sistema: solo en Java/Kotlin

Esta configuración se usa para desactivar el enrutamiento de puntos finales basado en cuentas, si es necesario, y omitir las reglas basadas en cuentas.

Valor predeterminado: `preferred`

Valores válidos:

- **`preferred`**— El punto final debe incluir el ID de cuenta, si está disponible.
- **`disabled`**— Un punto final resuelto no incluye el ID de cuenta.

- **`required`**— El punto final debe incluir el ID de la cuenta. Si el ID de la cuenta no está disponible, se SDK produce un error.

Compatibilidad con AWS SDKs

Las siguientes opciones SDKs son compatibles con las funciones y configuraciones descritas en este tema. Se anotan todas las excepciones parciales. Cualquier configuración de propiedades del JVM sistema es compatible con la AWS SDK for Java y el AWS SDK para Kotlin únicamente.

SDK	Ce	Notas o más información
AWS CLI v2	No	
SDKpara C++	No	
SDKpara Go V2 (1.x)	No	
SDKpara Go 1.x (V1)	No	
SDKpara Java 2.x	No	
SDKpara Java 1.x	Sí	
SDKpara JavaScript 3.x	No	
SDKpara JavaScript 2.x	No	
SDKpara Kotlin	No	
SDKpara. NET3.x	No	
SDKpara PHP 3.x	No	
SDKpara Python (Boto3)	No	
SDKpara Ruby 3.x	No	
SDKpara Rust	No	
SDKpara Swift	No	

SDK	C e	Notas o más información
Herramientas para PowerShell	No	

Application ID

Una sola Cuenta de AWS puede ser utilizado por múltiples aplicaciones de clientes para realizar llamadas a Servicios de AWS. El identificador de aplicación permite a los clientes identificar qué aplicación de origen realizó una serie de llamadas mediante un Cuenta de AWS. AWS SDKsy los servicios no utilizan ni interpretan este valor más que para mostrarlo en las comunicaciones con los clientes. Por ejemplo, este valor se puede incluir en los correos electrónicos operativos o en el AWS Health Dashboard para identificar de forma exclusiva cuáles de sus aplicaciones están asociadas a la notificación.

Configure esta funcionalidad mediante lo siguiente:

`sdk_ua_app_id`- compartido AWS **`config`**configuración de archivos, **`AWS_SDK_UA_APP_ID`**: variable de entorno, **`aws.userAgentAppId`**- propiedad JVM del sistema: solo en Java/Kotlin

> Esta configuración es una cadena única que se asigna a la aplicación para identificar cuáles de las aplicaciones están dentro de una determinada Cuenta de AWS hace llamadas a AWS.
>
> Valor predeterminado: `None`
>
> Valores válidos: cadena con una longitud máxima de 50. Se permiten letras, números y los siguientes caracteres especiales: `!`,`$`,`%`,`&`,`*`,`+`,`-`,`.`,`,`,`^`,`_`,`` ` ``,`|`,`~`.

Ejemplo de configuración de este valor en el archivo `config`:

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

Ejemplo de configuración de variables de entorno en Linux/macOS mediante la línea de comandos:

```
export AWS_SDK_UA_APP_ID=ABCDEF
```

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

Ejemplo de configuración de variables de entorno en Windows mediante la línea de comandos:

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

Si incluye símbolos que tienen un significado especial para la concha que se utiliza, evite el valor según corresponda.

Compatibilidad con AWS SDKs

Las siguientes opciones SDKs son compatibles con las funciones y configuraciones descritas en este tema. Se anotan todas las excepciones parciales. Cualquier configuración de propiedades del JVM sistema es compatible con la AWS SDK for Java y el AWS SDK para Kotlin únicamente.

SDK	Ce	Notas o más información
AWS CLI v2	Sí	
SDKpara C++	Sí	No se admite el archivo compartido `config`.
SDKpara Go V2 (1.x)	Sí	
SDKpara Go 1.x (V1)	No	
SDKpara Java 2.x	Parci	No se admite la configuración de `config` archivos compartidos; no se admite la variable de entorno.
SDKpara Java 1.x	No	
SDKpara JavaScript 3.x	Sí	
SDKpara JavaScript 2.x	No	
SDKpara Kotlin	Sí	
SDKpara. NET3.x	Sí	No se admiten variables de entorno.
SDKpara PHP 3.x	Sí	

SDK	Ce	Notas o más información
SDKpara Python (Boto3)	Sí	
SDKpara Ruby 3.x	Sí	
SDKpara Rust	Sí	
SDKpara Swift	No	
Herramientas para PowerShell	No	

Metadatos de EC2 instancias de Amazon

Amazon EC2 proporciona un servicio para las instancias denominado Instance Metadata Service (IMDS). Para obtener más información sobre este servicio, consulta Trabajar con metadatos de instancias en la Guía del EC2 usuario de Amazon. Al intentar recuperar las credenciales de una EC2 instancia de Amazon que se ha configurado con un IAM rol, la conexión al servicio de metadatos de la instancia se puede ajustar.

Configure esta funcionalidad mediante lo siguiente:

`metadata_service_num_attempts`- compartido AWS **`config`**configuración de archivos, **`AWS_METADATA_SERVICE_NUM_ATTEMPTS`**: variable de entorno

Esta configuración especifica la cantidad total de intentos que hay que realizar antes de intentar recuperar datos desde el servicio de metadatos de instancias.

Valor predeterminado: 1

Valores válidos: número mayor o igual a 1.

`metadata_service_timeout`- compartido AWS **`config`**configuración de archivos, **`AWS_METADATA_SERVICE_TIMEOUT`**: variable de entorno

Especifica el número de segundos antes de que se agote el tiempo de espera cuando se intentan recuperar datos desde el servicio de metadatos de instancias.

Valor predeterminado: 1

Valores válidos: número mayor o igual a 1.

Ejemplo de configuración de este valor en el archivo `config`:

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

Ejemplo de configuración de variables de entorno en Linux/macOS mediante la línea de comandos:

```
export AWS_METADATA_SERVICE_NUM_ATTEMPTS=10
export AWS_METADATA_SERVICE_TIMEOUT=10
```

Ejemplo de configuración de variables de entorno en Windows mediante la línea de comandos:

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

Compatibilidad con AWS SDKs

Las siguientes opciones SDKs son compatibles con las funciones y configuraciones descritas en este tema. Se anotan todas las excepciones parciales. Cualquier configuración de propiedades del JVM sistema es compatible con la AWS SDK for Java y el AWS SDK para Kotlin únicamente.

SDK	Ce	Notas o más información
AWS CLI v2	Sí	
SDKpara C++	No	
SDKpara Go V2 (1.x)	No	
SDKpara Go 1.x (V1)	No	
SDKpara Java 2.x	No	
SDKpara Java 1.x	Parci	Solo se admite `AWS_METADATA_SERVICE_TIMEOUT` .

SDK	Ce	Notas o más información
SDKpara JavaScript 3.x	No	
SDKpara JavaScript 2.x	No	
SDKpara Kotlin	No	
SDKpara. NET3.x	No	
SDKpara PHP 3.x	Sí	
SDKpara Python (Boto3)	Sí	
SDKpara Ruby 3.x	No	
SDKpara Rust	No	
SDKpara Swift	No	
Herramientas para PowerShell	No	

Puntos de acceso de Amazon S3

El servicio Amazon S3 proporciona puntos de acceso como una forma alternativa de interactuar con los buckets de Amazon S3. Los puntos de acceso pueden tener políticas y configuraciones únicas que se pueden aplicar a ellos en lugar de directamente al bucket. Con AWS SDKs, puede utilizar el punto de acceso Amazon Resource Names (ARNs) en el campo bucket para API las operaciones en lugar de especificar el nombre del bucket de forma explícita. Se utilizan para operaciones específicas, como utilizar un punto ARN de acceso `GetObject`para recuperar un objeto de un depósito o utilizar un punto ARN de acceso `PutObject`para añadir un objeto a un depósito.

Para obtener más información sobre los puntos de acceso de Amazon S3ARNs, consulte Uso de puntos de acceso en la Guía del usuario de Amazon S3.

Configure esta funcionalidad mediante lo siguiente:

`s3_use_arn_region`- compartido AWS **`config`**configuración de archivos, **`AWS_S3_USE_ARN_REGION`**: variable de entorno, **`aws.s3UseArnRegion`**- propiedad JVM del sistema: solo Java/Kotlin, Para configurar el valor directamente en el código, consulte su especificación directamente. SDK

Esta configuración controla si SDK utiliza el punto de acceso ARN Región de AWS para construir el punto final regional de la solicitud. Esto SDK valida que el ARN Región de AWS es servido por el mismo AWS partición según la configuración del cliente Región de AWS para evitar las llamadas entre particiones que muy probablemente fallarán. Si se ha definido de forma múltiple, prevalece la configuración por código, seguida de la configuración de la variable de entorno.

Valor predeterminado: `false`

Valores válidos:

- **`true`**— Los SDK usos, los ARN Región de AWS al construir el punto final en lugar del configurado por el cliente Región de AWS. Excepción: si el cliente está configurado Región de AWS es un FIPS Región de AWS, entonces debe coincidir con ARN la Región de AWS. De lo contrario, se producirá un error.
- **`false`**— Los SDK usos configurados por el cliente Región de AWS al construir el punto final.

Compatibilidad con AWS SDKs

Las siguientes opciones SDKs son compatibles con las funciones y configuraciones descritas en este tema. Se anotan todas las excepciones parciales. Cualquier configuración de propiedades del JVM sistema es compatible con la AWS SDK for Java y el AWS SDK para Kotlin únicamente.

SDK	Ce	Notas o más información
AWS CLI v2	Sí	
SDKpara C++	Sí	
SDKpara Go V2 (1.x)	Sí	
SDKpara Go 1.x (V1)	Sí	Para usar la configuración de archivos compartidos `config`, debe activar la carga desde el archivo de configuración; consulte Sesiones.

SDK	Ce	Notas o más información
SDKpara Java 2.x	Sí	
SDKpara Java 1.x	Sí	JVMpropiedad del sistema no compatible.
SDKpara JavaScript 3.x	Sí	
SDKpara JavaScript 2.x	Sí	
SDKpara Kotlin	Sí	
SDKpara. NET3.x	Sí	No sigue la prioridad estándar; el valor del archivo compartido `config` tiene prioridad sobre la variable de entorno.
SDKpara PHP 3.x	Sí	
SDKpara Python (Boto3)	Sí	
SDKpara Ruby 3.x	Sí	
SDKpara Rust	No	
SDKpara Swift	No	
Herramientas para PowerShell	Sí	No sigue la prioridad estándar; el valor del archivo compartido `config` tiene prioridad sobre la variable de entorno.

Puntos de acceso multirregión de Amazon S3

Los puntos de acceso multirregionales de Amazon S3 proporcionan un punto de enlace global que las aplicaciones pueden utilizar para gestionar las solicitudes de los buckets de Amazon S3 ubicados en varios Regiones de AWS. Puede utilizar puntos de acceso multirregionales para crear aplicaciones multirregionales con la misma arquitectura utilizada en una sola región y, a continuación, ejecutar esas aplicaciones en cualquier parte del mundo.

Para obtener más información acerca de los puntos de acceso multirregión, consulte Puntos de acceso multirregión de Amazon S3 en la Guía del usuario de Amazon S3.

Para obtener más información sobre los nombres de recursos de Amazon (ARNs) de puntos de acceso multirregionales, consulte Realizar solicitudes mediante un punto de acceso multirregional en la Guía del usuario de Amazon S3.

Para obtener más información acerca de los puntos de acceso multirregión, consulte Puntos de acceso multirregión de Amazon S3 en la Guía del usuario de Amazon S3.

El algoritmo SigV4a es la implementación de firma que se utiliza para firmar las solicitudes regionales globales. Este algoritmo se obtiene SDK mediante una dependencia de. Bibliotecas de Common Runtime (CRT) AWS

Configure esta funcionalidad mediante lo siguiente:

`s3_disable_multiregion_access_points`- compartido AWS **`config`**configuración de archivos, **`AWS_S3_DISABLE_MULTIREGION_ACCESS_POINTS`**: variable de entorno, **`aws.s3DisableMultiRegionAccessPoints`**- propiedad JVM del sistema: solo en Java/Kotlin, Para configurar el valor directamente en el código, consulte su especificación directamente. SDK

Esta configuración controla si es SDK posible que intente realizar solicitudes entre regiones. Si se ha definido de forma múltiple, prevalece la configuración por código, seguida de la configuración de la variable de entorno.

Valor predeterminado: `false`

Valores válidos:

- **`true`**: detiene el uso de solicitudes entre regiones.
- **`false`**: permite las solicitudes entre regiones mediante puntos de acceso multirregionales.

Compatibilidad con AWS SDKs

Las siguientes opciones SDKs son compatibles con las funciones y configuraciones descritas en este tema. Se anotan todas las excepciones parciales. Cualquier configuración de propiedades del JVM sistema es compatible con la AWS SDK for Java y el AWS SDK para Kotlin únicamente.

SDK	Ce	Notas o más información
AWS CLI v2	Sí	

SDK	Ce	Notas o más información
SDKpara C++	Sí	
SDKpara Go V2 (1.x)	Sí	
SDKpara Go 1.x (V1)	No	
SDKpara Java 2.x	Sí	
SDKpara Java 1.x	No	
SDKpara JavaScript 3.x	Sí	
SDKpara JavaScript 2.x	No	
SDKpara Kotlin	Sí	
SDKpara. NET3.x	Sí	
SDKpara PHP 3.x	Sí	
SDKpara Python (Boto3)	Sí	
SDKpara Ruby 3.x	Sí	
SDKpara Rust	Sí	
SDKpara Swift	No	
Herramientas para PowerShell	Sí	

Región de AWS

Regiones de AWS son un concepto importante que hay que entender cuando se trabaja con Servicios de AWS.

¿Con Regiones de AWS, puedes acceder Servicios de AWS que residen físicamente en un área geográfica específica. Esto puede ser útil para evitar redundancias y para que sus datos y

aplicaciones se ejecuten cerca del lugar desde donde usted y sus usuarios accederán a ellos. Las regiones proporcionar tolerancia a errores, estabilidad y resistencia, y también pueden reducir la latencia. Con las regiones, puede crear recursos redundantes que sigan estando disponibles y no resulten afectados por una interrupción regional.

La mayoría Servicio de AWS las solicitudes están asociadas a una región geográfica determinada. Los recursos que se crean en una región no existen en ninguna otra, a menos que se utilice explícitamente una función de replicación ofrecida por un Servicio de AWS. Por ejemplo, Amazon S3 y Amazon EC2 admiten la replicación entre regiones. Algunos servicios, por ejemploIAM, no tienen recursos regionales.

Con la Referencia general de AWScontiene información sobre lo siguiente:

- Para comprender la relación entre las regiones y los puntos de enlace, y para ver una lista de los puntos de enlace regionales existentes, consulte AWS puntos finales de servicio.
- Para ver la lista actual de todas las regiones y puntos de conexión compatibles con cada uno Servicio de AWS, consulte Puntos finales y cuotas del servicio.

Cómo crear clientes de servicio

Para acceder mediante programación Servicios de AWS, SDKs utilice una clase/objeto de cliente para cada Servicio de AWS. Si tu aplicación necesita acceder a AmazonEC2, por ejemplo, crearía un objeto EC2 cliente de Amazon para interactuar con ese servicio.

Si no se especifica explícitamente ninguna región para el cliente, el cliente utilizará de forma predeterminada la región establecida mediante la siguiente configuración de `region`. Sin embargo, la región activa de un cliente se puede establecer explícitamente para cualquier objeto de cliente individual. La configuración de la región de esta manera prevalece sobre cualquier configuración global para ese cliente de servicio concreto. La región alternativa se especifica durante la creación de instancias de ese cliente y es específica para usted SDK (consulte su SDK guía específica o su base SDK de código).

Configure esta funcionalidad mediante lo siguiente:

`region`- compartido AWS **`config`**configuración de archivos, **`AWS_REGION`**: variable de entorno, **`aws.region`**- propiedad JVM del sistema: solo en Java/Kotlin

Especifica el valor por defecto Región de AWS para usar AWS solicitudes. Esta región se usa para las solicitudes de SDK servicio que no se proporcionan con una región específica para su uso.

Valor predeterminado: ninguno. Debe especificar este valor de forma explícita.

Valores válidos:

- Cualquiera de los códigos de región disponibles para el servicio elegido, tal como se indica en AWS puntos finales de servicio en el AWS Referencia general. Por ejemplo, el valor `us-east-1` establece el punto final en Región de AWS Este de EE. UU. (Virginia del Norte).
- `aws-global`especifica el punto final global para los servicios que admiten un punto final global independiente además de los puntos finales regionales, como AWS Security Token Service (AWS STS) y Amazon Simple Storage Service (Amazon S3).

Ejemplo de configuración de este valor en el archivo `config`:

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

Ejemplo de configuración de variables de entorno en Linux/macOS mediante la línea de comandos:

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

Ejemplo de configuración de variables de entorno en Windows mediante la línea de comandos:

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

La mayoría SDKs tienen un objeto de «configuración» que está disponible para establecer la región predeterminada desde el código de la aplicación. Para obtener más información, consulta tu AWS SDKguía para desarrolladores.

Compatibilidad con AWS SDKs

Las siguientes opciones SDKs son compatibles con las funciones y configuraciones descritas en este tema. Se anotan todas las excepciones parciales. Cualquier configuración de propiedades del JVM sistema es compatible con la AWS SDK for Java y el AWS SDK para Kotlin únicamente.

SDK	Ce	Notas o más información
AWS CLI v2	Sí	AWS CLI v2 usa cualquier valor `AWS_REGION` antes de cualquier valor incluido `AWS_DEFAULT_REGION` (ambas variables están marcadas).
AWS CLI v1	Sí	AWS CLI v1 usa una variable de entorno nombrada `AWS_DEFAULT_REGION` para este propósito.
SDKpara C++	Sí	
SDKpara Go V2 (1.x)	Sí	
SDKpara Go 1.x (V1)	Sí	Para usar la configuración de archivos compartidos `config`, debe activar la carga desde el archivo de configuración; consulte Sesiones.
SDKpara Java 2.x	Sí	
SDKpara Java 1.x	Sí	
SDKpara JavaScript 3.x	Sí	
SDKpara JavaScript 2.x	Sí	
SDKpara Kotlin	Sí	
SDKpara. NET3.x	Sí	
SDKpara PHP 3.x	Sí	
SDKpara Python (Boto3)	Sí	SDKUtiliza una variable de entorno nombrada `AWS_DEFAULT_REGION` para este propósito.
SDKpara Ruby 3.x	Sí	
SDKpara Rust	Sí	
SDKpara Swift	Sí	

SDK	C e	Notas o más información
Herramientas para PowerShell	Sí	

AWS STS Puntos de conexión regionales

AWS Security Token Service (AWS STS) está disponible como servicio global y regional. Algunos de AWS SDKsy CLIs utilizan el punto final del servicio global (`https://sts.amazonaws.com`) de forma predeterminada, mientras que otros utilizan los puntos finales del servicio regional (`https://sts.{region_identifier}.{partition_domain}`). Las solicitudes globales se asignan a la región EE. UU. Este (Virginia del Norte). Para obtener más información sobre las AWS STS puntos finales, consulte Puntos finales en la AWS Security Token Service APIReferencia.

Es un AWS la mejor práctica es utilizar puntos finales regionales siempre que sea posible y configurar susRegión de AWS. Los clientes que se encuentren en particiones distintas de las comerciales deben utilizar puntos finales regionales. No todas SDKs las herramientas de Internet admiten esta configuración, pero todas tienen un comportamiento definido en cuanto a los puntos finales globales y regionales. Consulte la siguiente sección para obtener más información.

En el SDKs caso de las herramientas compatibles con esta configuración, los clientes pueden configurar la funcionalidad de la siguiente manera:

`sts_regional_endpoints`- compartido AWS **`config`**configuración de archivos,
`AWS_STS_REGIONAL_ENDPOINTS`: variable de entorno

Esta configuración especifica la forma en que la herramienta SDK o determina la Servicio de AWS punto final que utiliza para comunicarse con el AWS Security Token Service (AWS STS).

Valor predeterminado: `legacy`

> **Note**
>
> Todas las nuevas versiones SDK principales que se publiquen después de julio de 2022 se instalarán de forma predeterminada en`regional`. Es posible que las nuevas versiones SDK principales eliminen esta configuración y este `regional` comportamiento

de uso. Para reducir el impacto futuro de este cambio, le recomendamos que comience a usar `regional` en su aplicación siempre que sea posible.

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

- **`legacy`**— Utiliza el global AWS STS punto final,`sts.amazonaws.com`.
- **`regional`**— La herramienta SDK o siempre utiliza el AWS STS punto final para la región actualmente configurada. Por ejemplo, si el cliente está configurado para usar`us-west-2`, todas las llamadas a AWS STS se realizan en el punto final regional`sts.us-west-2.amazonaws.com`, en lugar de en el `sts.amazonaws.com` punto final global. Para enviar una solicitud al punto de enlace global mientras esta configuración está habilitada, puede establecer la región en `aws-global`.

Ejemplo de configuración de este valor en el archivo `config`:

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

Ejemplo de configuración de variables de entorno en Linux/macOS mediante la línea de comandos:

```
export AWS_STS_REGIONAL_ENDPOINTS=regional
```

Ejemplo de configuración de variables de entorno en Windows mediante la línea de comandos:

```
setx AWS_STS_REGIONAL_ENDPOINTS regional
```

Compatibilidad con AWS SDKs

> **Note**
>
> Es un AWS la mejor práctica es utilizar puntos finales regionales siempre que sea posible y configurar susRegión de AWS.

En la siguiente tabla se resume lo siguiente, para su herramientaSDK:

- Admite la configuración: si se admiten la variable de `config` archivo compartido y la variable de entorno para los puntos finales STS regionales.
- Valor de configuración predeterminado: el valor predeterminado de la configuración si es compatible.
- STSPunto final de destino del cliente de servicio predeterminado: qué punto final predeterminado utiliza el cliente, incluso si la configuración para cambiarlo no está disponible.
- Comportamiento alternativo del cliente de servicio: qué SDK hace cuando se supone que debe utilizar un punto final regional pero no se ha configurado ninguna región. Este es el comportamiento independientemente de si utiliza un punto final regional debido a un valor predeterminado o porque lo `regional` ha seleccionado la configuración.

La tabla también utiliza los siguientes valores:

- Punto final global:`https://sts.amazonaws.com`.
- Punto final regional: según la configuración Región de AWS utilizada por su aplicación.
- **`us-east-1`**(Regional): utiliza el punto final de la `us-east-1` región, pero con identificadores de sesión más largos que las solicitudes globales habituales.

SDK	Admite la configuración	Valor de configuración predeterminado	STSEndpoint de destino del cliente de servicio predeterminado	Comportamiento alternativo del cliente de servicio	Notas o más información
AWS CLI v2	N	N/A	Punto de conexión regional	Punto final global	
AWS CLI v1	S	`legacy`	Punto final global	Punto final global	
SDKpara C++	N	N/A	Punto de conexión regional	`us-east-1` (Regional)	

SDK	S	Valor de configuración predeterminado	STSEndpoint de destino del cliente de servicio predeterminado	Comportamiento alternativo del cliente de servicio	Notas o más información
SDKpara Go V2 (1.x)	N	N/A	Punto de conexión regional	Error en la solicitud	
SDKpara Go 1.x (V1)	S	`legacy`	Punto final global	Punto final global	Para usar la configuración de archivos compartidos `config`, debe activar la carga desde el archivo de configuración; consulte Sesiones.
SDKpara Java 2.x	N	N/A	Punto de conexión regional	Error en la solicitud	Si no hay ninguna región configurada, `AssumeRole` y `AssumeRoleWithWebIdentity` utilizará el STS punto final global
SDKpara Java 1.x	S	`legacy`	Punto final global	Punto final global	
SDKpara JavaScript 3.x	N	N/A	Punto de conexión regional	Error en la solicitud	
SDKpara JavaScript 2.x	S	`legacy`	Punto final global	Punto final global	
SDKpara Kotlin	N	N/A	Punto de conexión regional	Punto final global	

SDK	S	Valor de configuración predeterminado	STSEndpoint de destino del cliente de servicio predeterminado	Comportamiento alternativo del cliente de servicio	Notas o más información
SDKpara. NET3.x	S	`legacy`	Punto final global	Punto final global	
SDKpara PHP 3.x	S	`legacy`	Punto final global	Error en la solicitud	
SDKpara Python (Boto3)	S	`legacy`	Punto final global	Punto final global	
SDKpara Ruby 3.x	S	`regional`	Punto de conexión regional	Error en la solicitud	
SDKpara Rust	N	N/A	Punto de conexión regional	Error en la solicitud	
SDKpara Swift	N	N/A	Punto de conexión regional	Error en la solicitud	
Herramientas para PowerShell	S	`legacy`	Punto final global	Punto final global	

Pila doble y puntos finales FIPS

Configure esta funcionalidad mediante lo siguiente:

`use_dualstack_endpoint`- compartido AWS **`config`**configuración de archivos, **`AWS_USE_DUALSTACK_ENDPOINT`**: variable de entorno, **`aws.useDualstackEndpoint`**- propiedad JVM del sistema: solo Java/Kotlin

Activa o desactiva si SDK enviará las solicitudes a puntos finales de doble pila. Para obtener más información sobre los puntos de enlace de doble pila, que admiten tanto como IPv4 el IPv6 tráfico, consulte Uso de los puntos de enlace de doble pila de Amazon S3 en la Guía del usuario de Amazon Simple Storage Service. Los puntos de conexión de doble pila están disponibles para algunos servicios en algunas regiones.

Valor predeterminado: `false`

Valores válidos:

- **`true`**— La herramienta SDK o herramienta intentará utilizar puntos de enlace de doble pila para realizar solicitudes de red. Si no existe un punto final de doble pila para el servicio y/o Región de AWS, la solicitud fallará.
- **`false`**— La herramienta SDK or no utilizará puntos de conexión de doble pila para realizar solicitudes de red.

`use_fips_endpoint`- compartido AWS **`config`**configuración de archivos, **`AWS_USE_FIPS_ENDPOINT`**: variable de entorno, **`aws.useFipsEndpoint`**- propiedad JVM del sistema: solo Java/Kotlin

Activa o desactiva si la herramienta SDK o herramienta enviará solicitudes a FIPS puntos finales compatibles. Los estándares federales de procesamiento de información (FIPS) son un conjunto de requisitos de seguridad del Gobierno de EE. UU. para los datos y su cifrado. Las agencias gubernamentales, los socios y quienes deseen hacer negocios con el gobierno federal deben cumplir con FIPS las pautas. A diferencia del estándar AWS Los puntos finales utilizan una biblioteca de TLS software que cumple con FIPS la norma 140-2. FIPS Si esta configuración está habilitada y no existe un FIPS punto final para el servicio en su Región de AWS, el AWS la llamada puede fallar. Puntos de conexión específicos del servicioy la `--endpoint-url` opción para el AWS Command Line Interface anular esta configuración.

Para obtener más información sobre otras formas de especificar puntos FIPS finales, consulte Región de AWS, consulte FIPSPuntos finales por servicio. Para obtener más información sobre los puntos de enlace del servicio Amazon Elastic Compute Cloud, consulte los puntos de enlace de doble pila (IPv4yIPv6) en Amazon Reference. EC2 API

Valor predeterminado: `false`

Valores válidos:

- **`true`**— La herramienta SDK o herramienta enviará las solicitudes a los puntos de enlace que cumplan con los requisitosFIPS.
- **`false`**— La herramienta SDK o herramienta no enviará solicitudes a puntos finales que cumplan con los requisitosFIPS.

Compatibilidad con AWS SDKs

Las siguientes opciones SDKs son compatibles con las funciones y configuraciones descritas en este tema. Se anotan todas las excepciones parciales. Cualquier configuración de propiedades del JVM sistema es compatible con la AWS SDK for Java y el AWS SDK para Kotlin únicamente.

SDK	C e	Notas o más información
AWS CLI v2	Sí	
SDKpara C++	Sí	
SDKpara Go V2 (1.x)	Sí	
SDKpara Go 1.x (V1)	Sí	Para usar la configuración de archivos compartidos `config`, debe activar la carga desde el archivo de configuración; consulte Sesiones.
SDKpara Java 2.x	Sí	
SDKpara Java 1.x	No	
SDKpara JavaScript 3.x	Sí	
SDKpara JavaScript 2.x	Sí	
SDKpara Kotlin	Sí	
SDKpara. NET3.x	Sí	
SDKpara PHP 3.x	Sí	

SDK	Ce	Notas o más información
SDKpara Python (Boto3)	Sí	
SDKpara Ruby 3.x	Sí	
SDKpara Rust	Sí	
SDKpara Swift	Sí	
Herramientas para PowerShell	Sí	

Detección de puntos de conexión

SDKsutilice la detección de puntos finales para acceder a los puntos finales del servicio (URLspara acceder a varios recursos) y, al mismo tiempo, mantener la flexibilidad para AWS para modificarlos URLs según sea necesario. De esta forma, el código puede detectar automáticamente nuevos puntos de conexión. No hay puntos de conexión fijos para algunos servicios. En su lugar, para obtener los puntos de conexión disponibles durante el tiempo de ejecución, debe realizar una solicitud para obtener primero los puntos de conexión. Tras recuperar los puntos de conexión disponibles, el código utiliza los puntos de conexión para acceder a otras operaciones. Por ejemplo, en Amazon Timestream, realiza `DescribeEndpoints` una solicitud para recuperar SDK los puntos de enlace disponibles y, a continuación, los utiliza para completar operaciones específicas, como o. `CreateDatabase` `CreateTable`

Configure esta funcionalidad mediante lo siguiente:

`endpoint_discovery_enabled`- compartido AWS **`config`**configuración de archivos, **`AWS_ENABLE_ENDPOINT_DISCOVERY`**: variable de entorno, **`aws.endpointDiscoveryEnabled`**-propiedad JVM del sistema: solo Java/Kotlin, Para configurar el valor directamente en el código, consulte su especificación directamente. SDK

Activa o desactiva la detección de puntos finales para DynamoDB.

La detección de puntos de conexión es obligatoria en Timestream y opcional en Amazon DynamoDB. Esta configuración se establece de forma predeterminada en uno `true` o en `false` función de si el servicio requiere la detección de puntos finales. Las solicitudes Timestream se

establecen de forma predeterminada y las solicitudes de Amazon DynamoDB se establecen de forma predeterminada en. `true` `false`

Valores válidos:

- **`true`**— SDK Debería intentar descubrir automáticamente un punto final para los servicios en los que la detección de puntos finales sea opcional.
- **`false`**— No SDK deberían intentar detectar automáticamente un punto final en el caso de servicios en los que la detección de puntos finales sea opcional.

Compatibilidad con AWS SDKs

Las siguientes opciones SDKs son compatibles con las funciones y configuraciones descritas en este tema. Se anotan todas las excepciones parciales. Cualquier configuración de propiedades del JVM sistema es compatible con la AWS SDK for Java y el AWS SDK para Kotlin únicamente.

SDK	C e	Notas o más información
AWS CLI v2	Sí	
SDKpara C++	Sí	
SDKpara Go V2 (1.x)	Sí	
SDKpara Go 1.x (V1)	Sí	Para usar la configuración de archivos compartidos `config`, debe activar la carga desde el archivo de configuración; consulte Sesiones.
SDKpara Java 2.x	Sí	SDKPara Java 2.x utiliza el nombre de `AWS_ENDPO INT_DISCOVERY_ENABLED` la variable de entorno.
SDKpara Java 1.x	Parci	JVMpropiedad del sistema no compatible.
SDKpara JavaScript 3.x	Sí	
SDKpara JavaScript 2.x	Sí	
SDKpara Kotlin	Sí	

SDK	Ce	Notas o más información
SDKpara. NET3.x	Sí	
SDKpara PHP 3.x	Sí	
SDKpara Python (Boto3)	Sí	
SDKpara Ruby 3.x	Sí	
SDKpara Rust	Parci	Compatible solo con Timestream.
SDKpara Swift	No	
Herramientas para PowerShell	Sí	

Ajustes de configuración general

SDKsadmiten algunos ajustes generales que configuran SDK los comportamientos generales.

Configure esta funcionalidad mediante lo siguiente:

`api_versions`- compartido AWS **`config`**configuración de archivos

Alguno AWS los servicios mantienen varias API versiones para admitir la compatibilidad con versiones anteriores. De forma predeterminada, SDK y AWS CLI las operaciones utilizan la última API versión disponible. Si desea solicitar una API versión específica para utilizarla en sus solicitudes, incluya la `api_versions` configuración en su perfil.

Valor predeterminado: ninguno. (La última API versión la utiliza elSDK.)

Valores válidos: se trata de una configuración anidada seguida de una o más líneas sangradas, cada una de las cuales identifica una AWS el servicio y la API versión que se va a utilizar. Consulte la documentación del AWS servicio para saber qué API versiones están disponibles.

El ejemplo establece una API versión específica para dos AWS servicios del `config` archivo. Estas API versiones se utilizan únicamente para los comandos que se ejecutan en el perfil que

contiene esta configuración. Los comandos de cualquier otro servicio utilizan la versión más reciente de ese servicioAPI.

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

`ca_bundle`- compartido AWS **`config`**configuración de archivos, **`AWS_CA_BUNDLE`**: variable de entorno

Especifica la ruta a un paquete de certificados personalizado (un archivo con una `.pem` extensión) que se utilizará al establecer TLS las conexionesSSL/.

Valor predeterminado: ninguno

Valores válidos: especifique la ruta completa o el nombre del archivo base. Si hay un nombre de archivo base, el sistema intentará encontrar el programa en las carpetas especificadas por la variable del entorno `PATH`.

Ejemplo de configuración de este valor en el archivo `config`:

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

Debido a las diferencias en la forma en que los sistemas operativos gestionan las rutas y el escape de los caracteres de las rutas, a continuación se muestra un ejemplo de cómo configurar este valor en el `config` archivo en Windows:

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

Ejemplo de configuración de variables de entorno en Linux/macOS mediante la línea de comandos:

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

Ejemplo de configuración de variables de entorno en Windows mediante la línea de comandos:

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

`output`- compartido AWS **`config`**configuración de archivos

Especifica el formato de los resultados en el AWS CLI y otros AWS SDKsy herramientas.

Valor predeterminado: `json`

Valores válidos:

- **`json`**— La salida tiene el formato de una JSONcadena.
- **`yaml`**— La salida tiene el formato de una YAMLcadena.
- **`yaml-stream`**— La salida se transmite y se formatea como una cadena. YAML El streaming permite un manejo más rápido de tipos de datos de gran tamaño.
- **`text`**: la salida tiene el formato de varias líneas de valores de cadena separados por tabuladores. Esto puede ser útil para pasar la salida a un procesador de texto, como `grep`, `sed` o `awk`.
- **`table`**: el resultado tiene el formato de una tabla en la que se usan los caracteres +|- para los bordes de celda. Normalmente, la información se presenta en un formato que es más fácil de leer que los demás formatos, pero que no es útil para programar.

`parameter_validation`- compartido AWS **`config`**configuración de archivos

Especifica si la herramienta SDK o la herramienta intentarán validar los parámetros de la línea de comandos antes de enviarlos al AWS punto final del servicio.

Valor predeterminado: `true`

Valores válidos:

- **`true`**: el valor predeterminado. La herramienta SDK o realiza la validación de los parámetros de la línea de comandos por parte del cliente. Esto ayuda a la herramienta SDK o a confirmar que los parámetros son válidos y detecta algunos errores. La herramienta SDK o puede rechazar las solicitudes que no sean válidas antes de enviarlas al AWS punto final del servicio.
- **`false`**— La herramienta SDK or no valida los parámetros de la línea de comandos antes de enviarlos al AWS punto final del servicio. La AWS el punto final del servicio es responsable de validar todas las solicitudes y rechazar las que no sean válidas.

Compatibilidad con AWS SDKs

Las siguientes opciones SDKs son compatibles con las funciones y configuraciones descritas en este tema. Se anotan todas las excepciones parciales. Cualquier configuración de propiedades del JVM sistema es compatible con la AWS SDK for Java y el AWS SDK para Kotlin únicamente.

SDK	Ce	Notas o más información
AWS CLI v2	Parci	`api_versions` no admitidas.
SDKpara C++	Sí	
SDKpara Go V2 (1.x)	Parci	Las `api_versions` y la `parameter_validation` no son compatibles.
SDKpara Go 1.x (V1)	Parci	Las `api_versions` y la `parameter_validation` no son compatibles. Para usar la configuración de archivos compartidos `config`, debe activar la carga desde el archivo de configuración; consulte Sesiones.
SDKpara Java 2.x	No	
SDKpara Java 1.x	No	
SDKpara JavaScript 3.x	Sí	
SDKpara JavaScript 2.x	Sí	
SDKpara Kotlin	No	
SDKpara. NET3.x	No	
SDKpara PHP 3.x	Sí	
SDKpara Python (Boto3)	Sí	
SDKpara Ruby 3.x	Sí	
SDKpara Rust	No	
SDKpara Swift	No	
Herramientas para PowerShell	No	

IMDScliente

SDKsimplemente un cliente de Instance Metadata Service versión 2 (IMDSv2) mediante solicitudes orientadas a la sesión. Para obtener más informaciónIMDSv2, consulte Uso IMDSv2 en la Guía del EC2 usuario de Amazon. El IMDS cliente se puede configurar mediante un objeto de configuración de cliente disponible en la base SDK de código.

Configure esta funcionalidad mediante lo siguiente:

`retries`: miembro del objeto de configuración del cliente

El número de reintentos adicionales de cualquier solicitud fallida.

Valor predeterminado: 3

Valores válidos: un número mayor que 0.

`port`: miembro del objeto de configuración del cliente

El puerto del punto de conexión.

Valor predeterminado: 80

Valores válidos: un número.

`token_ttl`: miembro del objeto de configuración del cliente

El TTL del token.

Valor predeterminado: 21.600 segundos (6 horas, el tiempo máximo asignado).

Valores válidos: un número.

`endpoint`: miembro del objeto de configuración del cliente

El punto final deIMDS.

Valor predeterminado: si el `endpoint_mode` es igual a `IPv4`, el punto de conexión predeterminado es `http://169.254.169.254`. Valor predeterminado: si el `endpoint_mode` es igual a `IPv6`, el punto de conexión predeterminado es `http://[fd00:ec2::254]`.

Valores válidos: válidosURI.

La mayoría admite las siguientes opcionesSDKs. Consulte su base de SDK código específica para obtener más información.

`endpoint_mode`: miembro del objeto de configuración del cliente

El modo de punto final deIMDS.

Valor predeterminado: `IPv4`

Valores válidos: `IPv4`, `IPv6`

`http_open_timeout`: miembro del objeto de configuración del cliente (puede variar el nombre)

La cantidad de segundos que se va a esperar para que se abra la conexión.

Valor predeterminado: 1 segundo.

Valores válidos: un número mayor que 0.

`http_read_timeout`: miembro del objeto de configuración del cliente (puede variar el nombre)

El número de segundos que tarda en leerse un fragmento de datos.

Valor predeterminado: 1 segundo.

Valores válidos: un número mayor que 0.

`http_debug_output`: miembro del objeto de configuración del cliente (puede variar el nombre)

Establece un flujo de salida para la depuración.

Valor predeterminado: ninguno.

Valores válidos: un flujo de E/S válido, por ejemplo. STDOUT

`backoff`: miembro del objeto de configuración del cliente (puede variar el nombre)

El número de segundos que permanecen inactivos entre los reintentos o la función de espera proporcionada por el cliente para llamar. Esto reemplaza la estrategia de retroceso exponencial predeterminada.

Valor predeterminado: varía segúnSDK.

Valores válidos: varían segúnSDK. Puede ser un valor numérico o una llamada a una función personalizada.

Compatibilidad con AWS SDKs

Las siguientes opciones SDKs son compatibles con las funciones y configuraciones descritas en este tema. Se anotan todas las excepciones parciales. Todos los ajustes de propiedades del JVM sistema son compatibles con el AWS SDK for Java y el AWS SDK para Kotlin únicamente.

SDK	Ce	Notas o más información
AWS CLI v2	Sí	
SDKpara C++	No	IMDSv2se usa solo internamente. Consulte IMDSproveedor de credenciales.
SDKpara Go V2 (1.x)	Sí	
SDKpara Go 1.x (V1)	Sí	
SDKpara Java 2.x	Sí	
SDKpara Java 1.x	Sí	
SDKpara JavaScript 3.x	Sí	
SDKpara JavaScript 2.x	Sí	
SDKpara Kotlin	Sí	
SDKpara. NET3.x	Sí	
SDKpara PHP 3.x	Sí	
SDKpara Python (Boto3)	Sí	
SDKpara Ruby 3.x	Sí	
SDKpara Rust	Sí	
SDKpara Swift	Sí	

SDK	C e	Notas o más información
Herramientas para PowerShell	Sí	

Comportamiento de los reintentos

El comportamiento de SDKs reintento incluye la configuración relativa a la forma en que se intenta recuperarse de los errores resultantes de las solicitudes realizadas a Servicios de AWS.

Configure esta funcionalidad mediante lo siguiente:

`retry_mode`- compartido AWS **`config`**configuración de archivos, **`AWS_RETRY_MODE`**: variable de entorno, **`aws.retryMode`**- propiedad JVM del sistema: solo en Java/Kotlin

Especifica el modo en que la herramienta para desarrolladores SDK intenta volver a intentarlo.

Valor predeterminado: este valor es específico de su. SDK Consulta tu SDK guía específica o tu base SDK de código para ver si está predeterminada`retry_mode`.

Valores válidos:

- `standard`— (Recomendado) El conjunto recomendado de reglas de reintento para todos AWS SDKs. Este modo incluye un conjunto estándar de errores que se repiten y ajusta automáticamente el número de reintentos para maximizar la disponibilidad y la estabilidad. Este modo es seguro para su uso en aplicaciones multiusuario. El número máximo predeterminado de intentos con este modo es tres, a menos que los `max_attempts` se configuren de forma explícita.
- `adaptive`— Un modo de reintento, adecuado solo para casos de uso especializados, que incluye la funcionalidad del modo estándar, así como una limitación automática de la velocidad por parte del cliente. Este modo de reintento no se recomienda para aplicaciones con varios usuarios, a menos que se procure aislar a los inquilinos de las aplicaciones. Para obtener más información, consulte Elegir entre los modos `standard` y `adaptive` volver a intentarlo. Este modo es experimental y podría cambiar su comportamiento en el futuro.
- `legacy`— (No recomendado) Específico para SDK ti (consulta tu SDK guía específica o tu SDK código base).

`max_attempts`- compartido AWS **`config`**configuración de archivos, **`AWS_MAX_ATTEMPTS`**: variable de entorno, **`aws.maxAttempts`**- propiedad JVM del sistema: solo en Java/Kotlin

Especifica el número máximo de intentos que se pueden realizar en una solicitud.

Valor predeterminado: si no se especifica este valor, su valor predeterminado depende del valor de la configuración `retry_mode`:

- Si `retry_mode` lo está`legacy`: usa un valor predeterminado específico para usted SDK (consulte su SDK guía específica o su base SDK de código para `max_attempts` ver el valor predeterminado).
- Si el `retry_mode` es `standard`: realiza tres intentos.
- Si el `retry_mode` es `adaptive`: realiza tres intentos.

Valores válidos: un número mayor que 0.

Elegir entre los modos **`standard`** y **`adaptive`** volver a intentarlo

Le recomendamos que utilice el modo de `standard` reintento a menos que esté seguro de que su uso es el más adecuado. `adaptive`

> **Note**
>
> El `adaptive` modo presupone que se agrupan los clientes en función del ámbito en el que el servicio de backend puede limitar las solicitudes. Si no lo haces, la limitación de un recurso podría retrasar las solicitudes de un recurso no relacionado si utilizas el mismo cliente para ambos recursos.

Estándar	Adaptativo
Casos de uso de aplicaciones: todos.	Casos de uso de aplicaciones: 1. No es sensible a la latencia. 2. El cliente solo accede a un único recurso, o bien, usted proporciona una lógica para agrupar a sus clientes por separado según el recurso de servicio al que se accede.

Estándar	Adaptativo
Admite la interrupción de circuitos para evitar que se vuelvan a intentar durante SDK las interrupciones.	Admite la interrupción del circuito para evitar que se vuelva a intentar durante las interrupc iones. SDK
Utiliza un retroceso exponencial fluctuante en caso de averías.	Utiliza tiempos de espera dinámicos para intentar minimizar el número de solicitudes fallidas, a cambio de la posibilidad de aumentar la latencia.
Nunca retrasa el primer intento de solicitud, solo los reintentos.	Puede acelerar o retrasar el intento de solicitud inicial.

Si opta por utilizar `adaptive` el modo, la aplicación debe crear clientes diseñados en función de cada recurso que pueda estar limitado. Un recurso, en este caso, está más ajustado que solo pensar en cada uno Servicio de AWS. Servicios de AWS pueden tener dimensiones adicionales que utilizan para limitar las solicitudes. Usemos el servicio Amazon DynamoDB como ejemplo. DynamoDB utiliza Región de AWS además de la tabla a la que se accede para acelerar las solicitudes. Esto significa que una tabla a la que está accediendo tu código podría estar más restringida que otras. Si el código utilizaba el mismo cliente para acceder a todas las tablas y las solicitudes a una de esas tablas están restringidas, el modo de reintento adaptativo reducirá la tasa de solicitudes de todas las tablas. Su código debe diseñarse para tener un cliente por egion-and-table par R. Si experimentas una latencia inesperada al usar `adaptive` el modo, consulta la sección específica AWS guía de documentación del servicio que está utilizando.

Detalles de implementación del modo de reintento

La AWS SDKsutilice cubos de fichas para decidir si se debe volver a intentar una solicitud y (en el caso del modo de `adaptive` reintento) con qué rapidez se deben enviar las solicitudes. El grupo utiliza dos grupos de fichasSDK: un grupo de fichas de reintentos y un grupo de fichas de tasa de solicitudes.

- El depósito de reintentos se utiliza para determinar si se SDK deben deshabilitar temporalmente los reintentos a fin de proteger los servicios ascendentes y descendentes durante las interrupciones. Los tokens se obtienen del depósito antes de que se intenten reintentarlo y, cuando

las solicitudes se realizan correctamente, se devuelven al depósito. Si el depósito está vacío cuando se intenta reintentar, no SDK volverá a intentar la solicitud.

- El depósito de fichas de tasa de solicitudes solo se usa en el modo de `adaptive` reintento para determinar la velocidad a la que se envían las solicitudes. Los tokens se adquieren del depósito antes de que se envíe la solicitud y se devuelven al depósito a un ritmo determinado dinámicamente en función de la limitación de las respuestas devueltas por el servicio.

A continuación se muestra el pseudocódigo de alto nivel para ambos modos de reintento `standard` y `adaptive`:

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

A continuación se muestran más detalles sobre los componentes utilizados en el pseudocódigo:

`GetSendToken`:

Este paso solo se utiliza en el modo de reintento. `adaptive` Este paso adquiere un token del grupo de tokens de tasa de solicitud. Si un token no está disponible, esperará a que esté disponible. SDKEs posible que tengas opciones de configuración disponibles para rechazar la solicitud en lugar de esperar. Los tokens del depósito se rellenan a un ritmo que se determina de forma dinámica, en función del número de respuestas restrictivas que reciba el cliente.

`SendHTTPRequest`:

En este paso, se envía la solicitud a AWS. La mayoría AWS SDKsutilizan una HTTP biblioteca que utilice grupos de conexiones para reutilizar una conexión existente al realizar una HTTP solicitud. Por lo general, las conexiones se reutilizan si una solicitud falla debido a errores de limitación, pero no si la solicitud falla debido a un error transitorio.

`RequestBookkeeping`:

Los tokens se añaden al depósito de fichas si la solicitud se realiza correctamente. Solo en el modo de `adaptive` reintento, la tasa de llenado del depósito de fichas de tasa de solicitudes se actualiza en función del tipo de respuesta recibida.

`Retryable`:

Este paso determina si se puede volver a intentar una respuesta en función de lo siguiente:

- El código HTTP de estado.
- El código de error devuelto por el servicio.
- Los errores de conexión, definidos como cualquier error recibido por SDK el que no se recibe una HTTP respuesta del servicio.

Los errores transitorios (códigos de HTTP estado 400, 408, 500, 502, 503 y 504) y los errores de regulación (códigos de HTTP estado 400, 403, 429, 502, 503 y 509) pueden volver a intentarse. SDKEl comportamiento de los reintentos se determina en combinación con los códigos de error u otros datos del servicio.

`MAX_ATTEMPTS`:

El número máximo de intentos predeterminado lo establece la `retry_mode` configuración, a menos que la configuración lo anule. `max_attempts`

`HasRetryQuota`

En este paso, se obtiene un token del grupo de reintentos. Si el depósito de fichas de reintento está vacío, no se volverá a intentar la solicitud.

`ExponentialBackoff`

En el caso de un error que se pueda volver a intentar, el retraso del reintento se calcula mediante un retroceso exponencial truncado. El SDKs uso de un retardo exponencial binario truncado con

fluctuación de fase. El siguiente algoritmo muestra cómo se define la cantidad de tiempo de reposo, en segundos, para una respuesta a una solicitud `i`:

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

En el algoritmo anterior, se aplican los siguientes valores:

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

`r = 2`

`MAX_BACKOFF = 20 seconds`SDKspara la mayoría. Consulta tu SDK guía específica o tu código fuente para confirmarlo.

Compatibilidad con AWS SDKs

Las siguientes opciones SDKs son compatibles con las funciones y configuraciones descritas en este tema. Se anotan todas las excepciones parciales. Cualquier configuración de propiedades del JVM sistema es compatible con la AWS SDK for Java y el AWS SDK para Kotlin únicamente.

SDK	C e	Notas o más información
AWS CLI v2	Sí	
SDKpara C++	Sí	
SDKpara Go V2 (1.x)	Sí	
SDKpara Go 1.x (V1)	No	
SDKpara Java 2.x	Sí	
SDKpara Java 1.x	Sí	JVMpropiedades del sistema: usar `com.amazonaws.sdk.maxAttempts` en lugar de`aws.maxAttempts` ; usar `com.amazonaws.sdk.retryMode` en lugar de`aws.retryMode` .
SDKpara JavaScript 3.x	Sí	

SDK	Ce	Notas o más información
SDKpara JavaScript 2.x	No	Admite un número máximo de reintentos, el retroceso exponencial con fluctuación de fase y la opción de un método personalizado para el retraso de los reintentos.
SDKpara Kotlin	Sí	
SDKpara. NET3.x	Sí	
SDKpara PHP 3.x	Sí	
SDKpara Python (Boto3)	Sí	
SDKpara Ruby 3.x	Sí	
SDKpara Rust	Sí	
SDKpara Swift	Sí	
Herramientas para PowerShell	Sí	

Compresión de solicitudes

> **Note**
>
> Para obtener ayuda para comprender el diseño de las páginas de configuración o para interpretar la compatibilidad con AWS SDKsen la siguiente tabla, consultePáginas de configuración.

AWS SDKsy las herramientas pueden comprimir automáticamente las cargas útiles al enviar solicitudes a Servicios de AWS que admiten la recepción de cargas útiles comprimidas. Comprimir la carga útil en el cliente antes de enviarla a un servicio puede reducir el número total de solicitudes y el ancho de banda necesario para enviar datos al servicio, así como reducir las solicitudes que se realizan incorrectamente debido a las limitaciones del servicio en cuanto al tamaño de la carga útil.

Para la compresión, la herramienta SDK o selecciona un algoritmo de codificación compatible tanto con el servicio como con elSDK. Sin embargo, la lista actual de codificaciones posibles solo incluye gzip, pero es posible que se amplíe en el futuro.

La compresión de solicitudes puede resultar especialmente útil si tu aplicación utiliza Amazon CloudWatch. CloudWatch es un servicio de monitoreo y observabilidad que recopila datos operativos y de monitoreo en forma de registros, métricas y eventos. Un ejemplo de una operación de servicio que admite la compresión CloudWatch es el PutMetricDataAPImétodo.

Configure esta funcionalidad mediante lo siguiente:

`disable_request_compression`- compartido AWS **`config`**configuración de archivos, **`AWS_DISABLE_REQUEST_COMPRESSION`**: variable de entorno, **`aws.disableRequestCompression`**- propiedad JVM del sistema: solo Java/Kotlin

Activa o desactiva si la herramienta SDK o comprimirá una carga útil antes de enviar una solicitud.

Valor predeterminado: `false`

Valores válidos:

- **`true`**: desactive la compresión de solicitudes.
- **`false`**: utilice la compresión de solicitudes siempre que sea posible.

`request_min_compression_size_bytes`- compartido AWS **`config`**configuración de archivos, **`AWS_REQUEST_MIN_COMPRESSION_SIZE_BYTES`**: variable de entorno, **`aws.requestMinCompressionSizeBytes`**- propiedad JVM del sistema: solo Java/Kotlin

Establece el tamaño mínimo en bytes del cuerpo de la solicitud que la herramienta SDK o debe comprimir. Las cargas útiles pequeñas pueden aumentar de longitud al comprimirse, por lo que existe un límite inferior para realizar la compresión. Este valor está incluido, un tamaño de solicitud mayor o igual al valor se comprimirá.

Valor predeterminado: 10 240 bytes

Valores válidos: valor entero comprendido entre 0 y 10 485 760 bytes, ambos incluidos.

Compatibilidad con AWS SDKs

Las siguientes opciones SDKs son compatibles con las funciones y configuraciones descritas en este tema. Se anotan todas las excepciones parciales. Cualquier configuración de propiedades del JVM sistema es compatible con la AWS SDK for Java y el AWS SDK para Kotlin únicamente.

SDK	Ce	Notas o más información
AWS CLI v2	Sí	
SDKpara C++	Sí	
SDKpara Go V2 (1.x)	Sí	
SDKpara Go 1.x (V1)	No	
SDKpara Java 2.x	Sí	
SDKpara Java 1.x	No	
SDKpara JavaScript 3.x	Sí	
SDKpara JavaScript 2.x	No	
SDKpara Kotlin	Sí	
SDKpara. NET3.x	Sí	
SDKpara PHP 3.x	Sí	
SDKpara Python (Boto3)	Sí	
SDKpara Ruby 3.x	Sí	
SDKpara Rust	Sí	
SDKpara Swift	No	
Herramientas para PowerShell	Sí	

Puntos de conexión específicos del servicio

La configuración del punto final específico del servicio ofrece la opción de utilizar el punto final que elija para API las solicitudes y mantener esa opción. Esta configuración proporciona la flexibilidad necesaria para admitir puntos de conexión locales, puntos de conexión y puntos de conexión VPC locales de terceros AWS entornos de desarrollo. Se pueden usar diferentes puntos de conexión para los entornos de prueba y producción. Puede especificar un punto final URL para cada usuario Servicios de AWS.

Configure esta funcionalidad mediante lo siguiente:

`endpoint_url`- compartido AWS **`config`**configuración de archivos, **`AWS_ENDPOINT_URL`**: variable de entorno, **`aws.endpointUrl`**- propiedad JVM del sistema: solo en Java/Kotlin

Cuando se especifica directamente en un perfil o como variable de entorno, esta configuración especifica el punto de conexión que se utiliza para todas las solicitudes de servicio. Este punto final es anulado por cualquier punto de conexión específico del servicio configurado.

También puedes usar esta configuración dentro de una `services` sección de un espacio compartido AWS `config`archivo para configurar un punto final personalizado para un servicio específico. Para obtener una lista de todas las claves de identificación de servicio que se van a utilizar para las subsecciones en la sección `services`, consulte Identificadores de punto de conexión específicos del servicio.

Valor predeterminado: `none`

Valores válidos: AURL, incluidos el esquema y el host del punto final. Opcionalmente, URL puede contener un componente de ruta que contenga uno o más segmentos de ruta.

`AWS_ENDPOINT_URL_<SERVICE>`: variable de entorno, **`aws.endpointUrl<ServiceName>`**- propiedad JVM del sistema: solo en Java/Kotlin

`AWS_ENDPOINT_URL_<SERVICE>`, ¿dónde está el `<SERVICE>` Servicio de AWS identificador, establece un punto final personalizado para un servicio específico. Para obtener una lista de todas las variables de entorno específicas del servicio, consulte Identificadores de punto de conexión específicos del servicio.

Este punto de conexión específico del servicio anula cualquier punto de conexión global establecido en `AWS_ENDPOINT_URL`.

Valor predeterminado: `none`

Valores válidos: AURL, incluidos el esquema y el host del punto final. Opcionalmente, URL puede contener un componente de ruta que contenga uno o más segmentos de ruta.

`ignore_configured_endpoint_urls`- compartido AWS **`config`**configuración de archivos, **`AWS_IGNORE_CONFIGURED_ENDPOINT_URLS`**: variable de entorno, **`aws.ignoreConfiguredEndpointUrls`**- propiedad JVM del sistema: solo en Java/Kotlin

Esta configuración se utiliza para ignorar todas las configuraciones de puntos de conexión personalizadas.

Tenga en cuenta que cualquier punto de conexión explícito establecido en el código o en el propio cliente de servicio se utiliza independientemente de esta configuración. Por ejemplo, incluir el parámetro de línea de `--endpoint-url` comandos con un AWS CLI El comando o el paso de un punto final URL a un constructor de clientes siempre surtirán efecto.

Valor predeterminado: `false`

Valores válidos:

- **`true`**— La herramienta SDK o no lee ninguna opción de configuración personalizada del `config` archivo compartido ni de las variables de entorno para configurar un punto finalURL.
- **`false`**— La herramienta SDK o utiliza todos los puntos finales disponibles proporcionados por el usuario del `config` archivo compartido o de las variables de entorno.

Configuración de puntos de conexión mediante variables de entorno

Para dirigir las solicitudes de todos los servicios a un punto final personalizadoURL, defina la variable de entorno `AWS_ENDPOINT_URL` global.

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

Para enrutar las solicitudes de un determinado Servicio de AWS a un punto final personalizadoURL, utilice la variable de `AWS_ENDPOINT_URL_<SERVICE>` entorno. Amazon DynamoDB tiene un `serviceId` de `DynamoDB`. Para este servicio, la variable de URL entorno del punto final es`AWS_ENDPOINT_URL_DYNAMODB`. Este punto de conexión tiene prioridad sobre el punto de conexión global establecido en `AWS_ENDPOINT_URL` para este servicio.

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

Como otro ejemplo, AWS Elastic Beanstalk tiene un `serviceId` de `Elastic Beanstalk`. La Servicio de AWS el identificador se basa en el API modelo sustituyendo todos los espacios `serviceId` por guiones bajos y mayúsculas en todas las letras. Para este servicio, la variable de entorno de la URL del punto de conexión es `AWS_ENDPOINT_URL_ELASTIC_BEANSTALK`. Para obtener una lista de todas las variables de entorno específicas del servicio, consulte Identificadores de punto de conexión específicos del servicio.

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

Configure los puntos de conexión mediante el archivo compartido `config`

En el archivo compartido `config`, `endpoint_url` se usa en diferentes lugares para diferentes funciones.

- Si se especifica `endpoint_url` directamente en un `profile`, ese punto de conexión se convierte en el punto de conexión global.
- El `endpoint_url` anidado bajo una clave identificadora de servicio en una sección `services`, hace que ese punto de conexión se aplique únicamente a las solicitudes realizadas a ese servicio. Para obtener más información sobre cómo definir una sección de `services` en el archivo compartido `config`, consulte Formato del archivo de configuración.

En el siguiente ejemplo, se utiliza una `services` definición para configurar un punto de enlace específico del servicio URL que se utilizará para Amazon S3 y un punto de enlace global personalizado que se utilizará para todos los demás servicios:

```
[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
```

Un único perfil puede configurar puntos de conexión para varios servicios. En este ejemplo se muestra cómo configurar el punto final específico del servicio URLs para Amazon S3 y AWS Elastic Beanstalk en el mismo perfil. AWS Elastic Beanstalk tiene un `serviceId` de `Elastic Beanstalk`. La Servicio de AWS el identificador se basa en el API modelo sustituyendo todos los espacios `serviceId` por guiones bajos y minúsculas todas las letras. Por lo tanto, la clave identificadora del

servicio pasa a ser `elastic_beanstalk` y la configuración de este servicio comienza en la línea `elastic_beanstalk =` . Para obtener una lista de todas las claves de identificación de servicio que se van a utilizar en la sección de `services`, consulte Identificadores de punto de conexión específicos del servicio.

```
[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
```

La sección de configuración de servicios se puede utilizar en varios perfiles. Por ejemplo, dos perfiles pueden usar la misma definición de `services` y, al mismo tiempo, modificar otras propiedades del 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 los puntos de conexión de los perfiles mediante credenciales basadas en roles

Si su perfil tiene credenciales basadas en funciones configuradas mediante un `source_profile` parámetro para la funcionalidad de IAM asumir funciones, SDK solo utilizará configuraciones de servicio para el perfil especificado. No utiliza perfiles que estén vinculados a él por roles. Por ejemplo, mediante el siguiente archivo `config` compartido:

```
[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
```

Si utilizas el perfil `B` y haces una llamada en tu código a AmazonEC2, el punto final se resuelve como`https://profile-b-ec2-endpoint.aws`. Si el código realiza una solicitud a cualquier otro servicio, la resolución del punto de conexión no seguirá ninguna lógica personalizada. El punto de conexión no se convierte en el punto de conexión global definido en el perfil `A`. Para que un punto de conexión global surta efecto en el perfil `B`, tendrá que configurar `endpoint_url` directamente dentro del perfil `B`. Para obtener más información sobre la configuración de `source_profile`, consulte Asumir el rol de proveedor de credenciales.

Precedencia de configuración

La configuración de esta característica se puede usar al mismo tiempo, pero solo tendrá prioridad un valor por servicio. Para API llamadas realizadas a una persona determinada Servicio de AWS, se utiliza el siguiente orden para seleccionar un valor:

1. Cualquier ajuste explícito establecido en el código o en el propio cliente de un servicio tiene prioridad sobre cualquier otra cosa.
   - Para el registro AWS CLI, es el valor que proporciona el parámetro de la línea de `--endpoint-url` comandos. En el caso de unaSDK, las asignaciones explícitas pueden adoptar la forma de un parámetro que se establece al crear una instancia Servicio de AWS cliente u objeto de configuración.
2. El valor proporcionado por una variable de entorno específica del servicio, como `AWS_ENDPOINT_URL_DYNAMODB`.
3. El valor proporcionado por la variable de entorno de punto de conexión `AWS_ENDPOINT_URL` global.
4. El valor que proporciona la configuración anidada `endpoint_url` bajo una clave de identificación de servicio dentro de una sección `services` del archivo compartido `config`.
5. El valor proporcionado por la configuración de `endpoint_url` en un `profile` de un archivo compartido `config`.

6. Cualquier punto final predeterminado URL para el respectivo Servicio de AWS se usa por última vez.

Compatibilidad con AWS SDKs

Las siguientes opciones SDKs son compatibles con las funciones y configuraciones descritas en este tema. Se anotan todas las excepciones parciales. Cualquier configuración de propiedades del JVM sistema es compatible con la AWS SDK for Java y el AWS SDK para Kotlin únicamente.

SDK	Ce	Notas o más información
AWS CLI v2	Sí	
SDKpara C++	No	
SDKpara Go V2 (1.x)	Sí	
SDKpara Go 1.x (V1)	No	
SDKpara Java 2.x	Sí	
SDKpara Java 1.x	No	
SDKpara JavaScript 3.x	Sí	
SDKpara JavaScript 2.x	No	
SDKpara Kotlin	Sí	
SDKpara. NET3.x	Sí	
SDKpara PHP 3.x	Sí	
SDKpara Python (Boto3)	Sí	
SDKpara Ruby 3.x	Sí	
SDKpara Rust	No	
SDKpara Swift	No	

SDK	C e	Notas o más información
Herramientas para PowerShell	Sí	

Identificadores de punto de conexión específicos del servicio

Para obtener información sobre cómo y dónde usar los identificadores de la siguiente tabla, consulte Puntos de conexión específicos del servicio.

serviceId	Cl id ac de se pa us cc o A **c** fil	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
AccessAnalyzer	ac ly	AWS_ENDPOINT_URL_ACCESSANALYZER
Account	ac	AWS_ENDPOINT_URL_ACCOUNT
ACM	ac	AWS_ENDPOINT_URL_ACM
ACM PCA	ac	AWS_ENDPOINT_URL_ACM_PCA
Alexa For Business	a _I	AWS_ENDPOINT_URL_ALEXA_FOR_BUSINESS

serviceId	Cl id ac de se pa us cc o A\ **cc** fil	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
amp	ar	AWS_ENDPOINT_URL_AMP
Amplify	ar	AWS_ENDPOINT_URL_AMPLIFY
AmplifyBackend	ar cl	AWS_ENDPOINT_URL_AMPLIFYBACKEND
AmplifyUIBuilder	ar bu	AWS_ENDPOINT_URL_AMPLIFYUIBUILDER
API Gateway	ap ay	AWS_ENDPOINT_URL_API_GATEWAY
ApiGatewayManagementApi	ap yr nt	AWS_ENDPOINT_URL_APIGATEWAYMANAGEMENTAPI
ApiGatewayV2	ap yv	AWS_ENDPOINT_URL_APIGATEWAYV2
AppConfig	ap	AWS_ENDPOINT_URL_APPCONFIG
AppConfigData	ap da	AWS_ENDPOINT_URL_APPCONFIGDATA

serviceId	Cl id ac de se pa us co o AV **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
AppFabric	ap	AWS_ENDPOINT_URL_APPFABRIC
Appflow	ap	AWS_ENDPOINT_URL_APPFLOW
AppIntegrations	ap at	AWS_ENDPOINT_URL_APPINTEGRATIONS
Application Auto Scaling	ap or ca	AWS_ENDPOINT_URL_APPLICATION_AUTO_SCALING
Application Insights	ap or ts	AWS_ENDPOINT_URL_APPLICATION_INSIGHTS
ApplicationCostPro filer	ap or f:	AWS_ENDPOINT_URL_APPLICATIONCOSTPROFILER
App Mesh	ap	AWS_ENDPOINT_URL_APP_MESH
AppRunner	ap	AWS_ENDPOINT_URL_APPRUNNER

serviceId	Cl id ac de se pa us cc o A\ **ce** fil	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
AppStream	ap	AWS_ENDPOINT_URL_APPSTREAM
AppSync	ap	AWS_ENDPOINT_URL_APPSYNC
ARC Zonal Shift	a: _s	AWS_ENDPOINT_URL_ARC_ZONAL_SHIFT
Artifact	a:	AWS_ENDPOINT_URL_ARTIFACT
Athena	at	AWS_ENDPOINT_URL_ATHENA
AuditManager	au ge	AWS_ENDPOINT_URL_AUDITMANAGER
Auto Scaling	au in	AWS_ENDPOINT_URL_AUTO_SCALING
Auto Scaling Plans	au in	AWS_ENDPOINT_URL_AUTO_SCALING_PLANS
b2bi	b2	AWS_ENDPOINT_URL_B2BI
Backup	ba	AWS_ENDPOINT_URL_BACKUP

serviceId	Cl id ac de se pa us co o AV **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
Backup Gateway	ba te	AWS_ENDPOINT_URL_BACKUP_GATEWAY
BackupStorage	ba ra	AWS_ENDPOINT_URL_BACKUPSTORAGE
Batch	ba	AWS_ENDPOINT_URL_BATCH
BCM Data Exports	bc ex	AWS_ENDPOINT_URL_BCM_DATA_EXPORTS
Bedrock	be	AWS_ENDPOINT_URL_BEDROCK
Bedrock Agent	be ge	AWS_ENDPOINT_URL_BEDROCK_AGENT
Bedrock Agent Runtime	be ge in	AWS_ENDPOINT_URL_BEDROCK_AGENT_RUNTIME
Bedrock Runtime	be un	AWS_ENDPOINT_URL_BEDROCK_RUNTIME
billingconductor	bi nd	AWS_ENDPOINT_URL_BILLINGCONDUCTOR

serviceId	Cl id ac de se pa us cc o AV **cc** fil	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
Braket	b:	AWS_ENDPOINT_URL_BRAKET
Budgets	bu	AWS_ENDPOINT_URL_BUDGETS
Cost Explorer	cc o:	AWS_ENDPOINT_URL_COST_EXPLORER
chatbot	ch	AWS_ENDPOINT_URL_CHATBOT
Chime	ch	AWS_ENDPOINT_URL_CHIME
Chime SDK Identity	ch _:	AWS_ENDPOINT_URL_CHIME_SDK_IDENTITY
Chime SDK Media Pipelines	ch _r pe	AWS_ENDPOINT_URL_CHIME_SDK_MEDIA_PIPELINES
Chime SDK Meetings	ch _r	AWS_ENDPOINT_URL_CHIME_SDK_MEETINGS

serviceId	Cl id ac de se pa us co o AV **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
Chime SDK Messaging	ch _r g	AWS_ENDPOINT_URL_CHIME_SDK_MESSAGING
Chime SDK Voice	ch _v	AWS_ENDPOINT_URL_CHIME_SDK_VOICE
CleanRooms	cl s	AWS_ENDPOINT_URL_CLEANROOMS
CleanRoomsML	cl sr	AWS_ENDPOINT_URL_CLEANROOMSML
Cloud9	cl	AWS_ENDPOINT_URL_CLOUD9
CloudControl	cl rc	AWS_ENDPOINT_URL_CLOUDCONTROL
CloudDirectory	cl cf	AWS_ENDPOINT_URL_CLOUDDIRECTORY
CloudFormation	cl af	AWS_ENDPOINT_URL_CLOUDFORMATION

serviceId	Cl id ac de se pa us co o AV **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
CloudFront	cl t	AWS_ENDPOINT_URL_CLOUDFRONT
CloudFront KeyValueStore	cl t_ es	AWS_ENDPOINT_URL_CLOUDFRONT_KEYVALUESTORE
CloudHSM	cl	AWS_ENDPOINT_URL_CLOUDHSM
CloudHSM V2	cl v2	AWS_ENDPOINT_URL_CLOUDHSM_V2
CloudSearch	cl ch	AWS_ENDPOINT_URL_CLOUDSEARCH
CloudSearch Domain	cl ch	AWS_ENDPOINT_URL_CLOUDSEARCH_DOMAIN
CloudTrail	cl l	AWS_ENDPOINT_URL_CLOUDTRAIL
CloudTrail Data	cl l_	AWS_ENDPOINT_URL_CLOUDTRAIL_DATA

serviceId	Cl id ac de se pa us co o A\ **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
CloudWatch	c1 h	AWS_ENDPOINT_URL_CLOUDWATCH
codeartifact	co ac	AWS_ENDPOINT_URL_CODEARTIFACT
CodeBuild	co	AWS_ENDPOINT_URL_CODEBUILD
CodeCatalyst	co ys	AWS_ENDPOINT_URL_CODECATALYST
CodeCommit	co t	AWS_ENDPOINT_URL_CODECOMMIT
CodeDeploy	co y	AWS_ENDPOINT_URL_CODEDEPLOY
CodeGuru Reviewer	co re	AWS_ENDPOINT_URL_CODEGURU_REVIEWER
CodeGuru Security	co se	AWS_ENDPOINT_URL_CODEGURU_SECURITY

serviceId	Cl id ac de se pa us co o A\ **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
CodeGuruProfiler	co ro	AWS_ENDPOINT_URL_CODEGURUPROFILER
CodePipeline	co ir	AWS_ENDPOINT_URL_CODEPIPELINE
CodeStar	co	AWS_ENDPOINT_URL_CODESTAR
CodeStar connections	co co ns	AWS_ENDPOINT_URL_CODESTAR_CONNECTIONS
codestar notifications	co no ic	AWS_ENDPOINT_URL_CODESTAR_NOTIFICATIONS
Cognito Identity	co de	AWS_ENDPOINT_URL_COGNITO_IDENTITY
Cognito Identity Provider	co de ro	AWS_ENDPOINT_URL_COGNITO_IDENTITY_PROVIDER
Cognito Sync	co yr	AWS_ENDPOINT_URL_COGNITO_SYNC

serviceId	Cl id ac de se pa us co o A\ **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
Comprehend	co d	AWS_ENDPOINT_URL_COMPREHEND
ComprehendMedical	co dr	AWS_ENDPOINT_URL_COMPREHENDMEDICAL
Compute Optimizer	co pt	AWS_ENDPOINT_URL_COMPUTE_OPTIMIZER
Config Service	co rv	AWS_ENDPOINT_URL_CONFIG_SERVICE
Connect	co	AWS_ENDPOINT_URL_CONNECT
Connect Contact Lens	co or ns	AWS_ENDPOINT_URL_CONNECT_CONTACT_LENS
ConnectCampaigns	co mp	AWS_ENDPOINT_URL_CONNECTCAMPAIGNS
ConnectCases	co se	AWS_ENDPOINT_URL_CONNECTCASES

serviceId	Cl id ac de se pa us co o AV **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
ConnectParticipant	co rt	AWS_ENDPOINT_URL_CONNECTPARTICIPANT
ControlTower	co we	AWS_ENDPOINT_URL_CONTROLTOWER
Cost Optimization Hub	co m: hu	AWS_ENDPOINT_URL_COST_OPTIMIZATION_HUB
Cost and Usage Report Service	co us o: ce	AWS_ENDPOINT_URL_COST_AND_USAGE_REPO RT_SERVICE
Customer Profiles	cu p:	AWS_ENDPOINT_URL_CUSTOMER_PROFILES
DataBrew	da	AWS_ENDPOINT_URL_DATABREW
DataExchange	da ng	AWS_ENDPOINT_URL_DATAEXCHANGE

serviceId	Cl id ac de se pa us cc o A\ **ce** fil	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
Data Pipeline	da l:	AWS_ENDPOINT_URL_DATA_PIPELINE
DataSync	da	AWS_ENDPOINT_URL_DATASYNC
DataZone	da	AWS_ENDPOINT_URL_DATAZONE
DAX	da	AWS_ENDPOINT_URL_DAX
Detective	de	AWS_ENDPOINT_URL_DETECTIVE
Device Farm	de rr	AWS_ENDPOINT_URL_DEVICE_FARM
DevOps Guru	de ru	AWS_ENDPOINT_URL_DEVOPS_GURU
Direct Connect	d: nr	AWS_ENDPOINT_URL_DIRECT_CONNECT
Application Discovery Service	ap or e: ce	AWS_ENDPOINT_URL_APPLICATION_DISCOVE RY_SERVICE

serviceId	Cl id ac de se pa us cc o A\ **cc** fil	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
DLM	dl	AWS_ENDPOINT_URL_DLM
Database Migration Service	da m: _s	AWS_ENDPOINT_URL_DATABASE_MIGRATION_ SERVICE
DocDB	dc	AWS_ENDPOINT_URL_DOCDB
DocDB Elastic	dc st	AWS_ENDPOINT_URL_DOCDB_ELASTIC
drs	dr	AWS_ENDPOINT_URL_DRS
Directory Service	di _s	AWS_ENDPOINT_URL_DIRECTORY_SERVICE
DynamoDB	dy	AWS_ENDPOINT_URL_DYNAMODB
DynamoDB Streams	dy st	AWS_ENDPOINT_URL_DYNAMODB_STREAMS
EBS	eb	AWS_ENDPOINT_URL_EBS
EC2	ec	AWS_ENDPOINT_URL_EC2

serviceId	Cl id ac de se pa us co o A\ **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
EC2 Instance Connect	ec nc ct	AWS_ENDPOINT_URL_EC2_INSTANCE_CONNECT
ECR	ec	AWS_ENDPOINT_URL_ECR
ECR PUBLIC	ec c	AWS_ENDPOINT_URL_ECR_PUBLIC
ECS	ec	AWS_ENDPOINT_URL_ECS
EFS	ef	AWS_ENDPOINT_URL_EFS
EKS	el	AWS_ENDPOINT_URL_EKS
EKS Auth	el	AWS_ENDPOINT_URL_EKS_AUTH
Elastic Inference	el nf	AWS_ENDPOINT_URL_ELASTIC_INFERENCE
ElastiCache	el he	AWS_ENDPOINT_URL_ELASTICACHE
Elastic Beanstalk	el ea	AWS_ENDPOINT_URL_ELASTIC_BEANSTALK

serviceId	Cl id ac de se pa us co o AV **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
Elastic Transcoder	el ra	AWS_ENDPOINT_URL_ELASTIC_TRANSCODER
Elastic Load Balancing	el oa c:	AWS_ENDPOINT_URL_ELASTIC_LOAD_BALANCING
Elastic Load Balancing v2	el oa c:	AWS_ENDPOINT_URL_ELASTIC_LOAD_BALANCING_V2
EMR	er	AWS_ENDPOINT_URL_EMR
EMR containers	er in	AWS_ENDPOINT_URL_EMR_CONTAINERS
EMR Serverless	er r:	AWS_ENDPOINT_URL_EMR_SERVERLESS
EntityResolution	er o:	AWS_ENDPOINT_URL_ENTITYRESOLUTION

serviceId	Cl id ac de se pa us co o AV **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
Elasticsearch Service	e: a: i(	AWS_ENDPOINT_URL_ELASTICSEARCH_SERVICE
EventBridge	ev ge	AWS_ENDPOINT_URL_EVENTBRIDGE
Evidently	ev	AWS_ENDPOINT_URL_EVIDENTLY
finspace	f:	AWS_ENDPOINT_URL_FINSPACE
finspace data	f: da	AWS_ENDPOINT_URL_FINSPACE_DATA
Firehose	f:	AWS_ENDPOINT_URL_FIREHOSE
fis	f:	AWS_ENDPOINT_URL_FIS
FMS	fr	AWS_ENDPOINT_URL_FMS
forecast	fo	AWS_ENDPOINT_URL_FORECAST
forecastquery	fo ue	AWS_ENDPOINT_URL_FORECASTQUERY

serviceId	Cl id ac de se pa us cc o A\ **ce** fil	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
FraudDetector	f: ct	AWS_ENDPOINT_URL_FRAUDDETECTOR
FreeTier	f:	AWS_ENDPOINT_URL_FREETIER
FSx	fs	AWS_ENDPOINT_URL_FSX
GameLift	ga	AWS_ENDPOINT_URL_GAMELIFT
Glacier	gl	AWS_ENDPOINT_URL_GLACIER
Global Accelerator	gl ce	AWS_ENDPOINT_URL_GLOBAL_ACCELERATOR
Glue	gl	AWS_ENDPOINT_URL_GLUE
grafana	gr	AWS_ENDPOINT_URL_GRAFANA
Greengrass	gr s	AWS_ENDPOINT_URL_GREENGRASS
GreengrassV2	gr sv	AWS_ENDPOINT_URL_GREENGRASSV2

serviceId	Cl id ac de se pa us cc o A\ **cc** fil	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
GroundStation	gr t:	AWS_ENDPOINT_URL_GROUNDSTATION
GuardDuty	gu	AWS_ENDPOINT_URL_GUARDDUTY
Health	he	AWS_ENDPOINT_URL_HEALTH
HealthLake	he e	AWS_ENDPOINT_URL_HEALTHLAKE
Honeycode	ho	AWS_ENDPOINT_URL_HONEYCODE
IAM	ia	AWS_ENDPOINT_URL_IAM
identitystore	id to	AWS_ENDPOINT_URL_IDENTITYSTORE
imagebuilder	im de	AWS_ENDPOINT_URL_IMAGEBUILDER
ImportExport	im o:	AWS_ENDPOINT_URL_IMPORTEXPORT

serviceId	Cl id ac de se pa us cc o A\ **cc** fil	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
Inspector	ir	AWS_ENDPOINT_URL_INSPECTOR
Inspector Scan	ir _s	AWS_ENDPOINT_URL_INSPECTOR_SCAN
Inspector2	ir 2	AWS_ENDPOINT_URL_INSPECTOR2
InternetMonitor	ir or	AWS_ENDPOINT_URL_INTERNETMONITOR
IoT	ic	AWS_ENDPOINT_URL_IOT
IoT Data Plane	ic p:	AWS_ENDPOINT_URL_IOT_DATA_PLANE
IoT Jobs Data Plane	ic da e	AWS_ENDPOINT_URL_IOT_JOBS_DATA_PLANE
IoT 1Click Devices Service	ic k_ _s	AWS_ENDPOINT_URL_IOT_1CLICK_DEVICES_ SERVICE

serviceId	Cl id ac de se pa us co o A\ **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
IoT 1Click Projects	ic k_ s	AWS_ENDPOINT_URL_IOT_1CLICK_PROJECTS
IoTAnalytics	ic ic	AWS_ENDPOINT_URL_IOTANALYTICS
IotDeviceAdvisor	ic ac	AWS_ENDPOINT_URL_IOTDEVICEADVISOR
IoT Events	ic s	AWS_ENDPOINT_URL_IOT_EVENTS
IoT Events Data	ic s_	AWS_ENDPOINT_URL_IOT_EVENTS_DATA
IoTFleetHub	ic ul	AWS_ENDPOINT_URL_IOTFLEETHUB
IoTFleetWise	ic is	AWS_ENDPOINT_URL_IOTFLEETWISE
IoTSecureTunneling	ic tu	AWS_ENDPOINT_URL_IOTSECURETUNNELING

serviceId	Cl id ac de se pa us co o AV **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
IoTSiteWise	ic se	AWS_ENDPOINT_URL_IOTSITEWISE
IoTThingsGraph	ic gr	AWS_ENDPOINT_URL_IOTTHINGSGRAPH
IoTTwinMaker	ic ke	AWS_ENDPOINT_URL_IOTTWINMAKER
IoT Wireless	ic es	AWS_ENDPOINT_URL_IOT_WIRELESS
ivs	iv	AWS_ENDPOINT_URL_IVS
IVS RealTime	iv ir	AWS_ENDPOINT_URL_IVS_REALTIME
ivschat	iv	AWS_ENDPOINT_URL_IVSCHAT
Kafka	ka	AWS_ENDPOINT_URL_KAFKA
KafkaConnect	ka ec	AWS_ENDPOINT_URL_KAFKACONNECT
kendra	ke	AWS_ENDPOINT_URL_KENDRA

serviceId	Cl id ac de se pa us co o A\ **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
Kendra Ranking	ke nl	AWS_ENDPOINT_URL_KENDRA_RANKING
Keyspaces	ke	AWS_ENDPOINT_URL_KEYSPACES
Kinesis	ki	AWS_ENDPOINT_URL_KINESIS
Kinesis Video Archived Media	ki id iv a	AWS_ENDPOINT_URL_KINESIS_VIDEO_ARCHI VED_MEDIA
Kinesis Video Media	ki id a	AWS_ENDPOINT_URL_KINESIS_VIDEO_MEDIA
Kinesis Video Signaling	ki id a:	AWS_ENDPOINT_URL_KINESIS_VIDEO_SIGNALING
Kinesis Video WebRTC Storage	ki id to e	AWS_ENDPOINT_URL_KINESIS_VIDEO_WEBRT C_STORAGE

serviceId	Cl id ac de se pa us co o AV **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
Kinesis Analytics	k: na	AWS_ENDPOINT_URL_KINESIS_ANALYTICS
Kinesis Analytics V2	k: na v2	AWS_ENDPOINT_URL_KINESIS_ANALYTICS_V2
Kinesis Video	k: id	AWS_ENDPOINT_URL_KINESIS_VIDEO
KMS	kr	AWS_ENDPOINT_URL_KMS
LakeFormation	la t:	AWS_ENDPOINT_URL_LAKEFORMATION
Lambda	la	AWS_ENDPOINT_URL_LAMBDA
Launch Wizard	la za	AWS_ENDPOINT_URL_LAUNCH_WIZARD
Lex Model Building Service	le _l _s	AWS_ENDPOINT_URL_LEX_MODEL_BUILDING_ SERVICE

serviceId	Cl id ac de se pa us co o AV **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
Lex Runtime Service	le me e	AWS_ENDPOINT_URL_LEX_RUNTIME_SERVICE
Lex Models V2	le s_	AWS_ENDPOINT_URL_LEX_MODELS_V2
Lex Runtime V2	le me	AWS_ENDPOINT_URL_LEX_RUNTIME_V2
License Manager	l: ar	AWS_ENDPOINT_URL_LICENSE_MANAGER
License Manager Linux Subscriptions	l: ar nu r:	AWS_ENDPOINT_URL_LICENSE_MANAGER_LIN UX_SUBSCRIPTIONS
License Manager User Subscriptions	l: ar e: ip	AWS_ENDPOINT_URL_LICENSE_MANAGER_USE R_SUBSCRIPTIONS
Lightsail	l:	AWS_ENDPOINT_URL_LIGHTSAIL

serviceId	Cl id ac de se pa us cc o AV **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
Location	lo	AWS_ENDPOINT_URL_LOCATION
CloudWatch Logs	cl h_	AWS_ENDPOINT_URL_CLOUDWATCH_LOGS
CloudWatch Logs	cl h_	AWS_ENDPOINT_URL_CLOUDWATCH_LOGS
LookoutEquipment	lo u:	AWS_ENDPOINT_URL_LOOKOUTEQUIPMENT
LookoutMetrics	lo t:	AWS_ENDPOINT_URL_LOOKOUTMETRICS
LookoutVision	lo s:	AWS_ENDPOINT_URL_LOOKOUTVISION
m2	m2	AWS_ENDPOINT_URL_M2
Machine Learning	ma ea	AWS_ENDPOINT_URL_MACHINE_LEARNING
Macie2	ma	AWS_ENDPOINT_URL_MACIE2
ManagedBlockchain	ma oc	AWS_ENDPOINT_URL_MANAGEDBLOCKCHAIN

serviceId	Cl id ac de se pa us co o AV **ce** fil	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
ManagedBlockchain Query	ma oc qu	AWS_ENDPOINT_URL_MANAGEDBLOCKCHAIN_QUERY
Marketplace Agreement	ma ce er	AWS_ENDPOINT_URL_MARKETPLACE_AGREEMENT
Marketplace Catalog	ma ce g	AWS_ENDPOINT_URL_MARKETPLACE_CATALOG
Marketplace Deployment	ma ce me	AWS_ENDPOINT_URL_MARKETPLACE_DEPLOYMENT
Marketplace Entitlement Service	ma ce er vi	AWS_ENDPOINT_URL_MARKETPLACE_ENTITLEMENT_SERVICE

serviceId	Cl id ac de se pa us co o AV **ce** fil	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
Marketplace Commerce Analytics	ma ce ce ic	AWS_ENDPOINT_URL_MARKETPLACE_COMMERCE_ANALYTICS
MediaConnect	me ec	AWS_ENDPOINT_URL_MEDIACONNECT
MediaConvert	me e:	AWS_ENDPOINT_URL_MEDIACONVERT
MediaLive	me	AWS_ENDPOINT_URL_MEDIALIVE
MediaPackage	me ac	AWS_ENDPOINT_URL_MEDIAPACKAGE
MediaPackage Vod	me ac	AWS_ENDPOINT_URL_MEDIAPACKAGE_VOD
MediaPackageV2	me ac	AWS_ENDPOINT_URL_MEDIAPACKAGEV2
MediaStore	me e	AWS_ENDPOINT_URL_MEDIASTORE

serviceId	Cl id ac de se pa us co o A\ **ce** fil	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
MediaStore Data	me e_	AWS_ENDPOINT_URL_MEDIASTORE_DATA
MediaTailor	me o:	AWS_ENDPOINT_URL_MEDIATAILOR
Medical Imaging	me ma	AWS_ENDPOINT_URL_MEDICAL_IMAGING
MemoryDB	me	AWS_ENDPOINT_URL_MEMORYDB
Marketplace Metering	ma ce ng	AWS_ENDPOINT_URL_MARKETPLACE_METERING
Migration Hub	m: _I	AWS_ENDPOINT_URL_MIGRATION_HUB
mgn	mg	AWS_ENDPOINT_URL_MGN
Migration Hub Refactor Spaces	m: _I ct es	AWS_ENDPOINT_URL_MIGRATION_HUB_REFAC TOR_SPACES

serviceId	Cl id ac de se pa us co o A\ **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
MigrationHub Config	m: hu g	AWS_ENDPOINT_URL_MIGRATIONHUB_CONFIG
MigrationHubOrches trator	m: hu t:	AWS_ENDPOINT_URL_MIGRATIONHUBORCHESTRATOR
MigrationHubStrategy	m: hu gy	AWS_ENDPOINT_URL_MIGRATIONHUBSTRATEGY
Mobile	mo	AWS_ENDPOINT_URL_MOBILE
mq	mo	AWS_ENDPOINT_URL_MQ
MTurk	mt	AWS_ENDPOINT_URL_MTURK
MWAA	mw	AWS_ENDPOINT_URL_MWAA
Neptune	ne	AWS_ENDPOINT_URL_NEPTUNE
Neptune Graph	ne ra	AWS_ENDPOINT_URL_NEPTUNE_GRAPH

serviceId	Cl id ac de se pa us co o AV **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
neptunedata	ne ta	AWS_ENDPOINT_URL_NEPTUNEDATA
Network Firewall	ne i:	AWS_ENDPOINT_URL_NETWORK_FIREWALL
NetworkManager	ne na	AWS_ENDPOINT_URL_NETWORKMANAGER
NetworkMonitor	ne n:	AWS_ENDPOINT_URL_NETWORKMONITOR
nimble	n:	AWS_ENDPOINT_URL_NIMBLE
OAM	oa	AWS_ENDPOINT_URL_OAM
Omics	or	AWS_ENDPOINT_URL_OMICS
OpenSearch	op h	AWS_ENDPOINT_URL_OPENSEARCH
OpenSearchServerless	op hs ss	AWS_ENDPOINT_URL_OPENSEARCHSERVERLESS
OpsWorks	op	AWS_ENDPOINT_URL_OPSWORKS

serviceId	Cl id ac de se pa us cc o A\ **cc** fil	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
OpsWorksCM	op m	AWS_ENDPOINT_URL_OPSWORKSCM
Organizations	or ic	AWS_ENDPOINT_URL_ORGANIZATIONS
OSIS	os	AWS_ENDPOINT_URL_OSIS
Outposts	ou	AWS_ENDPOINT_URL_OUTPOSTS
p8data	p8	AWS_ENDPOINT_URL_P8DATA
p8data	p8	AWS_ENDPOINT_URL_P8DATA
Panorama	pa	AWS_ENDPOINT_URL_PANORAMA
Payment Cryptography	pa ry hy	AWS_ENDPOINT_URL_PAYMENT_CRYPTOGRAPHY
Payment Cryptography Data	pa ry hy	AWS_ENDPOINT_URL_PAYMENT_CRYPTOGRAPHY_DATA
Pca Connector Ad	pc ct	AWS_ENDPOINT_URL_PCA_CONNECTOR_AD

serviceId	Cl id ac de se pa us co o A\ **ce** fil	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
Personalize	pe ze	AWS_ENDPOINT_URL_PERSONALIZE
Personalize Events	pe ze	AWS_ENDPOINT_URL_PERSONALIZE_EVENTS
Personalize Runtime	pe ze e	AWS_ENDPOINT_URL_PERSONALIZE_RUNTIME
PI	p:	AWS_ENDPOINT_URL_PI
Pinpoint	p:	AWS_ENDPOINT_URL_PINPOINT
Pinpoint Email	p: er	AWS_ENDPOINT_URL_PINPOINT_EMAIL
Pinpoint SMS Voice	p: sr	AWS_ENDPOINT_URL_PINPOINT_SMS_VOICE
Pinpoint SMS Voice V2	p: sr _\	AWS_ENDPOINT_URL_PINPOINT_SMS_VOICE_V2

serviceId	Cl id ac de se pa us cc o A\ **cc** fil	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
Pipes	p:	AWS_ENDPOINT_URL_PIPES
Polly	p(	AWS_ENDPOINT_URL_POLLY
Pricing	p:	AWS_ENDPOINT_URL_PRICING
PrivateNetworks	p: tv	AWS_ENDPOINT_URL_PRIVATENETWORKS
Proton	p:	AWS_ENDPOINT_URL_PROTON
QBusiness	qI	AWS_ENDPOINT_URL_QBUSINESS
QConnect	q(	AWS_ENDPOINT_URL_QCONNECT
QLDB	q:	AWS_ENDPOINT_URL_QLDB
QLDB Session	q: i(	AWS_ENDPOINT_URL_QLDB_SESSION
QuickSight	qu t	AWS_ENDPOINT_URL_QUICKSIGHT
RAM	r:	AWS_ENDPOINT_URL_RAM

serviceId	Cl id ac de se pa us co o AV **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
rbin	rb	AWS_ENDPOINT_URL_RBIN
RDS	rd	AWS_ENDPOINT_URL_RDS
RDS Data	rd	AWS_ENDPOINT_URL_RDS_DATA
Redshift	re	AWS_ENDPOINT_URL_REDSHIFT
Redshift Data	re da	AWS_ENDPOINT_URL_REDSHIFT_DATA
Redshift Serverless	re se s	AWS_ENDPOINT_URL_REDSHIFT_SERVERLESS
Rekognition	re or	AWS_ENDPOINT_URL_REKOGNITION
repostspace	re ce	AWS_ENDPOINT_URL_REPOSTSPACE
resiliencehub	re eh	AWS_ENDPOINT_URL_RESILIENCEHUB

serviceId	Cl id ac de se pa us co o A\ **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
Resource Explorer 2	re ex 2	AWS_ENDPOINT_URL_RESOURCE_EXPLORER_2
Resource Groups	re gr	AWS_ENDPOINT_URL_RESOURCE_GROUPS
Resource Groups Tagging API	re gr gg	AWS_ENDPOINT_URL_RESOURCE_GROUPS_TAG GING_API
RoboMaker	ro	AWS_ENDPOINT_URL_ROBOMAKER
RolesAnywhere	ro he	AWS_ENDPOINT_URL_ROLESANYWHERE
Route 53	ro	AWS_ENDPOINT_URL_ROUTE_53
Route53 Recovery Cluster	ro ec lu	AWS_ENDPOINT_URL_ROUTE53_RECOVERY_CLUSTER

serviceId	Cl id ac de se pa us co o AV **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
Route53 Recovery Control Config	rc ec or nt	AWS_ENDPOINT_URL_ROUTE53_RECOVERY_CO NTROL_CONFIG
Route53 Recovery Readiness	rc ec ea	AWS_ENDPOINT_URL_ROUTE53_RECOVERY_RE ADINESS
Route 53 Domains	rc dc	AWS_ENDPOINT_URL_ROUTE_53_DOMAINS
Route53Resolver	rc sc	AWS_ENDPOINT_URL_ROUTE53RESOLVER
RUM	ru	AWS_ENDPOINT_URL_RUM
S3	s3	AWS_ENDPOINT_URL_S3
S3 Control	s3 l	AWS_ENDPOINT_URL_S3_CONTROL
S3Outposts	s3 s	AWS_ENDPOINT_URL_S3OUTPOSTS

serviceId	Cl id ac de se pa us co o AV **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
SageMaker	sa	AWS_ENDPOINT_URL_SAGEMAKER
SageMaker A2I Runtime	sa _a ir	AWS_ENDPOINT_URL_SAGEMAKER_A2I_RUNTIME
Sagemaker Edge	sa _e	AWS_ENDPOINT_URL_SAGEMAKER_EDGE
SageMaker FeatureSt ore Runtime	sa _f to ir	AWS_ENDPOINT_URL_SAGEMAKER_FEATUREST ORE_RUNTIME
SageMaker Geospatial	sa _g a:	AWS_ENDPOINT_URL_SAGEMAKER_GEOSPATIAL
SageMaker Metrics	sa _r	AWS_ENDPOINT_URL_SAGEMAKER_METRICS
SageMaker Runtime	sa _r	AWS_ENDPOINT_URL_SAGEMAKER_RUNTIME

serviceId	Cl id ac de se pa us cc o A\ **cc** fil	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
savingsplans	sa ar	AWS_ENDPOINT_URL_SAVINGSPLANS
Scheduler	sc	AWS_ENDPOINT_URL_SCHEDULER
schemas	sc	AWS_ENDPOINT_URL_SCHEMAS
SimpleDB	sd	AWS_ENDPOINT_URL_SIMPLEDB
Secrets Manager	se ar	AWS_ENDPOINT_URL_SECRETS_MANAGER
SecurityHub	se ub	AWS_ENDPOINT_URL_SECURITYHUB
SecurityLake	se al	AWS_ENDPOINT_URL_SECURITYLAKE
ServerlessApplicationRepository	se sa io to	AWS_ENDPOINT_URL_SERVERLESSAPPLICATIONREPOSITORY

serviceId	Cl id ac de se pa us cc o A\ **cc** fil	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
Service Quotas	se uc	AWS_ENDPOINT_URL_SERVICE_QUOTAS
Service Catalog	se at	AWS_ENDPOINT_URL_SERVICE_CATALOG
Service Catalog AppRegistry	se at p:	AWS_ENDPOINT_URL_SERVICE_CATALOG_APP REGISTRY
ServiceDiscovery	se sc	AWS_ENDPOINT_URL_SERVICEDISCOVERY
SES	se	AWS_ENDPOINT_URL_SES
SESv2	se	AWS_ENDPOINT_URL_SESV2
Shield	sh	AWS_ENDPOINT_URL_SHIELD
signer	si	AWS_ENDPOINT_URL_SIGNER
SimSpaceWeaver	si ea	AWS_ENDPOINT_URL_SIMSPACEWEAVER
SMS	sr	AWS_ENDPOINT_URL_SMS

serviceId	Cl id ac de se pa us co o AV **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
Snow Device Managemen t	sr ce me	AWS_ENDPOINT_URL_SNOW_DEVICE_MANAGEMENT
Snowball	sr	AWS_ENDPOINT_URL_SNOWBALL
SNS	sr	AWS_ENDPOINT_URL_SNS
SQS	sc	AWS_ENDPOINT_URL_SQS
SSM	ss	AWS_ENDPOINT_URL_SSM
SSM Contacts	ss ct	AWS_ENDPOINT_URL_SSM_CONTACTS
SSM Incidents	ss er	AWS_ENDPOINT_URL_SSM_INCIDENTS
Ssm Sap	ss	AWS_ENDPOINT_URL_SSM_SAP
SSO	ss	AWS_ENDPOINT_URL_SSO
SSO Admin	ss	AWS_ENDPOINT_URL_SSO_ADMIN
SSO OIDC	ss	AWS_ENDPOINT_URL_SSO_OIDC

serviceId	Cl id ac de se pa us cc o A\ **cc** fil	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
SFN	st	AWS_ENDPOINT_URL_SFN
Storage Gateway	st at	AWS_ENDPOINT_URL_STORAGE_GATEWAY
STS	st	AWS_ENDPOINT_URL_STS
SupplyChain	su in	AWS_ENDPOINT_URL_SUPPLYCHAIN
Support	su	AWS_ENDPOINT_URL_SUPPORT
Support App	su pp	AWS_ENDPOINT_URL_SUPPORT_APP
SWF	sw	AWS_ENDPOINT_URL_SWF
synthetics	sy s	AWS_ENDPOINT_URL_SYNTHETICS
Textract	te	AWS_ENDPOINT_URL_TEXTRACT
Timestream InfluxDB	ti m_ b	AWS_ENDPOINT_URL_TIMESTREAM_INFLUXDB

serviceId	Cl id ac de se pa us co o AV **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
Timestream Query	t: m_	AWS_ENDPOINT_URL_TIMESTREAM_QUERY
Timestream Write	t: m_	AWS_ENDPOINT_URL_TIMESTREAM_WRITE
tnb	tr	AWS_ENDPOINT_URL_TNB
Transcribe	t: e	AWS_ENDPOINT_URL_TRANSCRIBE
Transfer	t:	AWS_ENDPOINT_URL_TRANSFER
Translate	t:	AWS_ENDPOINT_URL_TRANSLATE
TrustedAdvisor	t: v:	AWS_ENDPOINT_URL_TRUSTEDADVISOR
VerifiedPermissions	ve e: s	AWS_ENDPOINT_URL_VERIFIEDPERMISSIONS
Voice ID	vo	AWS_ENDPOINT_URL_VOICE_ID

serviceId	Cl id ac de se pa us co o AV **co** fil	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
VPC Lattice	vp ce	AWS_ENDPOINT_URL_VPC_LATTICE
WAF	wa	AWS_ENDPOINT_URL_WAF
WAF Regional	wa na	AWS_ENDPOINT_URL_WAF_REGIONAL
WAFV2	wa	AWS_ENDPOINT_URL_WAFV2
WellArchitected	we te	AWS_ENDPOINT_URL_WELLARCHITECTED
Wisdom	wi	AWS_ENDPOINT_URL_WISDOM
WorkDocs	wo	AWS_ENDPOINT_URL_WORKDOCS
WorkLink	wo	AWS_ENDPOINT_URL_WORKLINK
WorkMail	wo	AWS_ENDPOINT_URL_WORKMAIL
WorkMailMessageFlow	wo es w	AWS_ENDPOINT_URL_WORKMAILMESSAGEFLOW

serviceId	Cl id ac de se pa us co o A\ **ce** fil	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
WorkSpaces	wo s	AWS_ENDPOINT_URL_WORKSPACES
WorkSpaces Thin Client	wo s_ ie	AWS_ENDPOINT_URL_WORKSPACES_THIN_CLIENT
WorkSpaces Web	wo s_	AWS_ENDPOINT_URL_WORKSPACES_WEB
XRay	x:	AWS_ENDPOINT_URL_XRAY

Valores predeterminados de configuración inteligente

Con la función de configuración inteligente por defecto, AWS SDKspuede proporcionar valores predeterminados optimizados y predefinidos para otros ajustes de configuración.

Configure esta funcionalidad mediante lo siguiente:

defaults_mode- compartido AWS **config**configuración de archivos, **AWS_DEFAULTS_MODE**: variable de entorno, **aws.defaultsMode**- propiedad JVM del sistema: solo en Java/Kotlin

> Con esta configuración, puede elegir un modo que se alinee con la arquitectura de la aplicación y, a continuación, proporcionar valores predeterminados optimizados para la aplicación. Si un AWS SDKla configuración tiene un valor establecido explícitamente, entonces ese valor siempre

tiene prioridad. Si un AWS SDKla configuración no tiene un valor establecido de forma explícita y no `defaults_mode` es igual al heredado, esta función puede proporcionar diferentes valores predeterminados para diversas configuraciones optimizadas para su aplicación. La configuración puede incluir lo siguiente: configuración de HTTP comunicación, comportamiento de reintento, configuración de punto final regional del servicio y, posiblemente, cualquier configuración SDK relacionada. Los clientes que utilizan esta característica pueden obtener nuevos valores predeterminados de configuración adaptados a los escenarios de uso habituales. Si el tuyo no `defaults_mode` es igual `alegacy`, te recomendamos que realices pruebas en la aplicación cuando la actualicesSDK, ya que los valores predeterminados proporcionados podrían cambiar a medida que evolucionen las prácticas recomendadas.

Valor predeterminado: `legacy`

Nota: Las nuevas versiones principales de SDKs se establecerán de forma predeterminada en`standard`.

Valores válidos:

- `legacy`— Proporciona la configuración predeterminada que varía según el establecimiento de SDK y que existía antes de su establecimiento`defaults_mode`.
- `standard`: proporciona los últimos valores predeterminados recomendados que deberían poder ejecutarse de forma segura en la mayoría de los escenarios.
- `in-region`— Se basa en el modo estándar e incluye una optimización adaptada a las aplicaciones que requieren Servicios de AWS desde dentro del mismo Región de AWS.
- `cross-region`— Se basa en el modo estándar e incluye una optimización adaptada a las aplicaciones que requieren Servicios de AWS en una región diferente.
- `mobile`: se basa en el modo estándar e incluye una optimización adaptada a las aplicaciones móviles.
- `auto`: se basa en el modo estándar e incluye funciones experimentales. SDKIntenta descubrir el entorno de ejecución para determinar automáticamente la configuración adecuada. La detección automática se basa en la heurística y no proporciona una precisión del 100 %. Si no se puede determinar el tiempo de ejecución, se utiliza el modo `standard`. La autodetección podría consultar los metadatos de la instancia, lo que podría introducir latencia. Si la startup es fundamental para tu aplicación, te recomendamos que elijas un `defaults_mode` explícito en su lugar.

Ejemplo de configuración de este valor en el archivo `config`:

```
[default]
defaults_mode = standard
```

Los siguientes parámetros pueden optimizarse en función de la selección de `defaults_mode`:

- `retryMode`— Especifica cómo se SDK reintentan los intentos. Consulte Comportamiento de los reintentos.
- `stsRegionalEndpoints`— Especifica cómo SDK determina el Servicio de AWS punto final que utiliza para comunicarse con el AWS Security Token Service (AWS STS). MiraAWS STS Puntos de conexión regionales .
- `s3UsEast1RegionalEndpoints`— Especifica cómo SDK determina la AWS punto final de servicio que utiliza para comunicarse con Amazon S3 de la `us-east-1` región.
- `connectTimeoutInMillis`: tras realizar un intento de conexión inicial en un socket, el tiempo transcurrido hasta que se agote el tiempo de espera. Si el cliente no recibe la finalización del apretón de manos de conexión, se da por vencido y no se realiza la operación.
- `tlsNegotiationTimeoutInMillis`— El tiempo máximo que puede tardar un TLS apretón de manos desde el momento en que se envía el CLIENT HELLO mensaje hasta el momento en que el cliente y el servidor negocian completamente los cifrados e intercambian claves.

El valor predeterminado de cada configuración cambia en función del valor `defaults_mode` seleccionado para la aplicación. Estos valores se configuran actualmente de la siguiente manera (sujetos a cambios):

Parámetro	modo **`standard`**	modo **`in-region`**	modo **`cross-region`**	modo **`mobile`**
`retryMode`	`standard`	`standard`	`standard`	`standard`
`stsRegionalEndpoints`	`regional`	`regional`	`regional`	`regional`
`s3UsEast1RegionalEndpoints`	`regional`	`regional`	`regional`	`regional`

Parámetro	modo **standard**	modo **in-region**	modo **cross-region**	modo **mobile**
`connectTimeoutInMillis`	3100	1 100	3100	30000
`tlsNegotiationTimeoutInMillis`	3100	1 100	3100	30000

Por ejemplo, si el `defaults_mode` que ha seleccionado es `standard`, entonces el valor `standard` se asignará para `retry_mode` (de las opciones `retry_mode` válidas) y el valor `regional` se asignará para `stsRegionalEndpoints` (de las opciones `stsRegionalEndpoints` válidas).

Compatibilidad con AWS SDKs

Las siguientes opciones SDKs son compatibles con las funciones y configuraciones descritas en este tema. Se anotan todas las excepciones parciales. Cualquier configuración de propiedades del JVM sistema es compatible con la AWS SDK for Java y el AWS SDK para Kotlin únicamente.

SDK	Compatible	Notas o más información
AWS CLI v2	No	
SDKpara C++	Sí	Parámetros no optimizados: `stsRegionalEndpoints` , `s3UsEast1RegionalEndpoints` , `tlsNegotiationTimeoutInMillis` .
SDKpara Go V2 (1.x)	Sí	Parámetros no optimizados: `retryMode` , `stsRegionalEndpoints` ,

SDK	Compatible	Notas o más información
		`s3UsEast1RegionalEndpoints` .
SDKpara Go 1.x (V1)	No	
SDKpara Java 2.x	Sí	Parámetros no optimizados: `stsRegionalEndpoints` .
SDKpara Java 1.x	No	
SDKpara JavaScript 3.x	Sí	Parámetros no optimizados: `stsRegionalEndpoints` , `s3UsEast1RegionalEndpoints` , `tlsNegotiationTimeoutInMillis` . `connectTimeoutInMillis` se llama `connectionTimeout` .
SDKpara JavaScript 2.x	No	
SDKpara Kotlin	No	
SDKpara. NET3.x	Sí	Parámetros no optimizados: `connectTimeoutInMillis` , `tlsNegotiationTimeoutInMillis` .
SDKpara PHP 3.x	Sí	Parámetros no optimizados: `tlsNegotiationTimeoutInMillis` .
SDKpara Python (Boto3)	Sí	Parámetros no optimizados: `tlsNegotiationTimeoutInMillis` .

SDK	Compatible	Notas o más información
SDKpara Ruby 3.x	Sí	
SDKpara Rust	No	
SDKpara Swift	No	
Herramientas para PowerShell	Sí	Parámetros no optimizados: `connectTimeoutInMillis` , `tlsNegotiationTimeoutInMillis` .

Bibliotecas de Common Runtime (CRT) AWS

Las bibliotecas Common Runtime (CRT) de AWS son una biblioteca base de los SDK. El CRT es una familia modular de paquetes independientes, escrita en C. Cada paquete ofrece un buen rendimiento y ocupa un espacio mínimo para las diferentes funcionalidades requeridas. Estas funcionalidades son comunes y se comparten en todos los SDK, lo que proporciona una mejor reutilización, optimización y precisión del código. Los paquetes son:

- `awslabs/aws-c-auth`: autenticación de AWS del lado del cliente (proveedores de credenciales estándar y firma (sigv4))
- `awslabs/aws-c-cal`: tipos primitivos criptográficos, hashes (MD5, SHA256, SHA256 HMAC), firmantes, AES
- `awslabs/aws-c-common`: estructuras de datos básicas, tipos primitivos de subproceso/sincronización, administración de búferes, funciones relacionadas con stdlib
- `awslabs/aws-c-compression`: algoritmos de compresión (codificación/decodificación de Huffman)
- `awslabs/aws-c-event-stream`: procesamiento de mensajes de flujo de eventos (encabezados, preludio, carga útil, crc/trailer), implementación de llamadas a procedimientos remotos (RPC) sobre transmisiones de eventos
- `awslabs/aws-c-http`: implementación de las especificaciones de HTTP/1.1 y de HTTP/2 en C99
- `awslabs/aws-c-io`: sockets (TCP, UDP), DNS, canalizaciones, bucles de eventos, canales, SSL/TLS
- `awslabs/aws-c-iot`: implementación C99 de la integración de servicios AWS de IoT en la nube con dispositivos
- `awslabs/aws-c-mqtt`: protocolo de mensajería ligero y estándar para Internet de las cosas (IoT)
- `awslabs/aws-c-s3`: implementación de la biblioteca C99 para comunicarse con el servicio Amazon S3, diseñada para maximizar el rendimiento en las instancias Amazon EC2 de gran ancho de banda
- `awslabs/aws-c-sdkutils`: una biblioteca de utilidades para analizar y administrar perfiles de AWS
- `awslabs/aws-checksums`: CRC32c y CRC32 multiplataforma acelerados por hardware, que recurren a implementaciones de software eficientes

- awslabs/aws-lc: biblioteca criptográfica de uso general mantenida por el equipo de criptografía de AWS para AWS y sus clientes, basada en el código del proyecto Google BoringSSL y el proyecto OpenSSL
- awslabs/s2n: implementación C99 de los protocolos TLS/SSL, diseñada para ser pequeña y rápida, con la seguridad como prioridad

El CRT está disponible en todos los SDK excepto en Go.

Dependencias de CRT

Las bibliotecas CRT forman una red compleja de relaciones y dependencias. Conocer estas relaciones es útil si necesita crear el CRT directamente desde la fuente. Sin embargo, la mayoría de los usuarios acceden a la funcionalidad CRT a través del SDK de su idioma (como el SDK de AWS para C++ o el SDK de AWS para Java) o el SDK para dispositivos IoT de su idioma (como el SDK de AWS IoT para C++ o el SDK de AWS IoT para Java). En el siguiente diagrama, el recuadro de enlaces CRT de idiomas hace referencia al paquete que contiene las bibliotecas CRT de un SDK de lenguaje específico. Se trata de una colección de paquetes con este formato `aws-crt-*`, donde "*" es un lenguaje del SDK (como `aws-crt-cpp` o `aws-crt-java`).

La siguiente es una ilustración de las dependencias jerárquicas de las bibliotecas CRT.

AWS Política de mantenimiento de SDK y herramientas

Información general

Este documento describe la política de mantenimiento de los kits y herramientas de desarrollo de AWS software (SDK), incluidos los SDK móviles y de IoT, y sus dependencias subyacentes. AWS proporciona periódicamente a los AWS SDK y las herramientas actualizaciones que pueden incluir compatibilidad con AWS API nuevas o actualizadas, nuevas funciones, mejoras, correcciones de errores, parches de seguridad o actualizaciones de la documentación. Las actualizaciones también pueden abordar los cambios en las dependencias, los idiomas, los tiempos de ejecución y los sistemas operativos. AWS Las versiones del SDK se publican en los administradores de paquetes (por ejemplo NuGet, Maven o PyPI) y están disponibles como código fuente en. GitHub

Recomendamos a los usuarios que utilicen up-to-date las versiones del SDK para mantenerse al día con las últimas funciones, actualizaciones de seguridad y dependencias subyacentes. No se recomienda el uso continuo de una versión del SDK no admitida, y debe hacerse según el criterio del usuario.

Control de versiones

Las versiones de lanzamiento del AWS SDK tienen el formato X.Y.Z, donde X representa la versión principal. El aumento de la versión principal de un SDK indica que este ha tenido cambios considerables y sustanciales para admitir nuevos modismos y patrones en el idioma. Las versiones principales se introducen cuando las interfaces públicas (como las clases, métodos, tipos, etc.), los comportamientos o la semántica cambian. Las aplicaciones deben actualizarse para que funcionen con la versión más reciente del SDK. Es importante actualizar las versiones principales con cuidado y de acuerdo con las pautas de actualización proporcionadas por AWS.

Ciclo de vida de las versiones principales del

El ciclo de vida de las principales versiones de SDK y herramientas consta de 5 fases, que se describen a continuación.

- Vista previa para desarrolladores (fase 0): durante esta fase, los SDK no son compatibles, no deben usarse en entornos de producción y están pensados únicamente para facilitar el acceso anticipado y para enviar comentarios. Es posible que en futuras versiones se introduzcan cambios

importantes. Una vez que AWS identifique una versión como un producto estable, puede marcarla como versión candidata. Las versiones candidatas a ser lanzadas están listas para su publicación en GA, a menos que surjan errores importantes, y recibirán soporte técnico completo de AWS .

- Disponibilidad general (GA) (fase 1): durante esta fase, los SDK son totalmente compatibles. AWS proporcionará versiones periódicas del SDK que incluyen soporte para nuevos servicios, actualizaciones de API para los servicios existentes y correcciones de errores y de seguridad. En el caso de Tools, AWS se publicarán versiones periódicas que incluyen nuevas actualizaciones de funciones y correcciones de errores. AWS será compatible con la versión GA de un SDK durante al menos 24 meses.
- Anuncio de mantenimiento (fase 2): AWS se publicará un anuncio público al menos 6 meses antes de que el SDK entre en modo de mantenimiento. Durante este período, el SDK seguirá siendo totalmente compatible. Por lo general, el modo de mantenimiento se anuncia al mismo tiempo que la siguiente versión principal pasa a GA.
- Mantenimiento (fase 3): durante el modo de mantenimiento, AWS limita las versiones del SDK para abordar únicamente las correcciones de errores críticos y los problemas de seguridad. Un SDK no recibirá actualizaciones de API para servicios nuevos o existentes, ni se actualizará para que sea compatible con nuevas regiones. El modo de mantenimiento tiene una duración predeterminada de 12 meses, a menos que se especifique lo contrario.
- Fin del soporte (fase 4): cuando un SDK llega al final del soporte, ya no recibirá actualizaciones ni versiones. Las versiones publicadas anteriormente seguirán estando disponibles a través de los administradores de paquetes públicos y el código permanecerá activo GitHub. El GitHub repositorio puede estar archivado. El uso de un SDK disponible end-of-support queda a discreción del usuario. Recomendamos a los usuarios que actualicen a la nueva versión principal.

La siguiente es una ilustración visual del ciclo de vida de la versión principal del SDK. Tenga en cuenta que los plazos que se muestran a continuación son ilustrativos y no vinculantes.

Ciclo de vida de

La mayoría de AWS los SDK tienen dependencias subyacentes, como los tiempos de ejecución de los idiomas, los sistemas operativos o las bibliotecas y marcos de terceros. Estas dependencias suelen estar vinculadas a la comunidad lingüística o al proveedor propietario de ese componente en particular. Cada comunidad o proveedor publica su propio end-of-support cronograma para su producto.

Los siguientes términos se utilizan para clasificar las dependencias subyacentes de terceros:

- Sistema operativo (SO): algunos ejemplos incluyen Amazon Linux AMI, Amazon Linux 2, Windows 2008, Windows 2012, Windows 2016, etc.
- Lenguaje del tiempo de ejecución: algunos ejemplos son Java 7, Java 8, Java 11, .NET Core, .NET Standard, .NET PCL, etc.
- Biblioteca/Marco de trabajo de terceros: algunos ejemplos incluyen OpenSSL, .NET Framework 4.5, Java EE, etc.

Nuestra política consiste en seguir dando soporte a las dependencias del SDK durante al menos 6 meses después de que la comunidad o el proveedor hayan dejado de dar soporte a la dependencia. Sin embargo, esta política puede variar en función de la dependencia específica.

> **Note**
>
> AWS se reserva el derecho de interrumpir el soporte para una dependencia subyacente sin aumentar la versión principal del SDK

Métodos de comunicación

Los anuncios de mantenimiento se comunican de varias maneras:

- Se envía un anuncio por correo electrónico a las cuentas afectadas en el que anunciamos nuestros planes de dejar de ofrecer soporte para la versión específica del SDK. El correo electrónico describirá la ruta de acceso end-of-support, especificará los plazos de la campaña y proporcionará una guía de actualización.
- AWS La documentación del SDK, como la documentación de referencia de la API, las guías de usuario, las páginas de marketing de los productos del SDK y los GitHub archivos readme (s), se actualiza para indicar el calendario de la campaña y proporcionar orientación sobre la actualización de las aplicaciones afectadas.
- Se publica una entrada de AWS blog en la que se describe el camino a seguir end-of-support y se reiteran los plazos de la campaña.
- Se añaden advertencias de obsolescencia a los SDK, en las que se describe la ruta de acceso a la documentación del SDK end-of-support y se enlaza con ella.

Para ver la lista de las principales versiones disponibles de los AWS SDK y las herramientas y en qué punto del ciclo de vida de mantenimiento se encuentran, consulte. Compatibilidad de versiones

AWS SDKsy compatibilidad con las versiones de Tools

En la siguiente tabla se muestra la lista de las disponibles AWS Versiones principales del kit de desarrollo de software (SDK) y en qué parte del ciclo de vida del mantenimiento se encuentran, con los plazos correspondientes. Para obtener información detallada sobre el ciclo de vida de las versiones principales de AWS SDKsy Tools y sus dependencias subyacentes, consultePolítica de mantenimiento.

SDK	Versión principal	Fase actual	Fecha de disponibilidad general	Notas
AWS CLI	1.x	Disponibilidad general	2 de septiembre de 2013	
AWS CLI	2.x	Disponibilidad general	2/10/2020	
SDKpara C++	1.x	Disponibilidad general	2 de septiembre de 2015	
SDKpara Go V2	V2 1.x	Disponibilidad general	19/1/2021	
SDKpara Go	1.x	Mantenimiento	19/11/2015	Consulte el anuncio para conocer los detalles y las fechas
SDKpara Java	1.x	Mantenimiento	25/03/2010	Consulte el anuncio para conocer los detalles y las fechas

SDK	Versión principal	Fase actual	Fecha de disponibilidad general	Notas
SDKpara Java	2.x	Disponibilidad general	20/11/2018	
SDKpara JavaScript	1.x	Fin-del-soporte	6/5/2013	
SDKpara JavaScript	2.x	Mantenimiento	19/06/2014	Consulte el anuncio para conocer los detalles y las fechas
SDKpara JavaScript	3.x	Disponibilidad general	15/12/2020	
SDKpara Kotlin	1.x	Disponibilidad general	27 de noviembre de 2023	
SDKpara. NET	1.x	Fin-del-soporte	11/2009	
SDKpara. NET	2.x	Fin-del-soporte	8/11/2013	
SDKpara. NET	3.x	Disponibilidad general	28/7/2015	
SDK para PHP	2.x	Fin-del-soporte	2/11/2012	
SDK para PHP	3.x	Disponibilidad general	27/05/2015	
SDKpara Python (Boto2)	1.x	Fin-del-soporte	13 de julio de 2011	
SDKpara Python (Boto3)	1.x	Disponibilidad general	22/06/2015	

SDK	Versión principal	Fase actual	Fecha de disponibilidad general	Notas
SDKpara Python (Botocore)	1.x	Disponibilidad general	22/06/2015	
SDKpara Ruby	1.x	Fin-del-soporte	14/7/2011	
SDKpara Ruby	2.x	Fin-del-soporte	15/02/2015	
SDKpara Ruby	3.x	Disponibilidad general	29 de agosto de 2017	
SDKpara Rust	1.x	Disponibilidad general	27/11/2023	
SDKpara Swift	1.x	Disponibilidad general	17/09/2024	
Herramientas para PowerShell	2.x	Fin-del-soporte	8/11/2013	
Herramientas para PowerShell	3.x	Fin-del-soporte	29/7/2015	
Herramientas para PowerShell	4.x	Disponibilidad general	21/11/2019	

¿Busca una herramienta SDK o herramienta que no se menciona? El cifradoSDKs, los dispositivos SDKs IoT y los dispositivos mótilesSDKs, por ejemplo, no se incluyen en esta guía. Para encontrar documentación sobre estas otras herramientas, consulte Herramientas útiles AWS.

Historial de documentos para AWS SDKsy Guía de referencia de herramientas

En la siguiente tabla se describen las adiciones y actualizaciones importantes del AWS SDKsy la Guía de referencia de herramientas. Para recibir notificaciones sobre las actualizaciones de esta documentación, puede suscribirse al RSS feed.

Cambio	Descripción	Fecha
Añadir Swift SDK a la referencia de ajustes	Añadir la SDK compatibilidad con Swift a todas las referencias de configuración: compatibilidad con AWS SDKstablas.	17 de septiembre de 2024
SDKpara las propiedades del sistema Java 1.x	Añada detalles sobre las opciones de configuración JVM del sistema compatibles mediante el AWS SDK for Java 1.x.	30 de mayo de 2024
Actualizaciones de configuraciones	Añada los ajustes JVM de configuración del sistema.	27 de marzo de 2024
Actualizaciones de la tabla de compatibilidad	Actualizaciones de la compatibilidad para SDK brindar soporte, actualizaciones de los procedimientos del Centro de IAM Identidad.	20 de febrero de 2024
Actualización de credenciales del contenedor. IMDSactualización.	Añadir soporte para AmazonEKS. Se ha añadido una configuración para deshabilitar la opción IMDSv1 alternativa.	29 de diciembre de 2023

Compresión de solicitudes	Agregar configuración para la característica de compresión de solicitudes.	27 de diciembre de 2023
Tablas de compatibilidad	Se actualizaron las tablas de compatibilidad SDK y las funciones de las herramien tas para SDK incluirlas para Kotlin, SDK Rust y AWS Tools for PowerShell.	10 de diciembre de 2023
Actualizaciones de autentica ción	Actualizaciones de los métodos de autenticación SDKs y las herramientas compatibles.	1 de julio de 2023
IAMactualizaciones de mejores prácticas	Guía actualizada para alinearla con las IAM mejores prácticas. Para obtener más información, consulte las mejores prácticas de seguridad en IAM.	27 de febrero de 2023
SSOactualizaciones	Actualizaciones de SSO las credenciales para la nueva configuración del SSO token.	19 de noviembre de 2022
Actualizaciones de configura ciones	Actualizaciones de la tabla de soporte para configuración general y puntos de acceso de varias regiones de Amazon S3.	17 de noviembre de 2022
Actualizaciones de configura ciones	Actualizaciones en cuanto a la claridad del IMDS cliente y IMDS las credenciales. Actualizaciones de las variables de entorno.	4 de noviembre de 2022

Actualización de la página de bienvenida	Anunciamos Amazon CodeWhisperer.	22 de septiembre de 2022
Cambio de nombre de servicio para inicio de sesión único	Actualizaciones para reflejar eso AWS SSOahora se conoce como AWS IAM Identity Center.	26 de julio de 2022
Actualización de configuraciones	Actualizaciones menores en los detalles del archivo de configuración y en los ajustes compatibles.	15 de junio de 2022
Actualización	Actualización masiva de casi todas las partes de esta guía.	1 de febrero de 2022
Versión inicial	La primera versión de esta guía está disponible para el público.	13 de marzo de 2020

Glosario de AWS

Para ver la terminología más reciente de AWS, consulte el Glosario de AWS en la Referencia de Glosario de AWS.

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la version original de inglés, prevalecerá la version en inglés.