Guía de referencia

AWS SDKs y herramientas

AWS SDKs y herramientas: Guía de referencia

Table of Contents

Qué se trata en la Guía de referencia de herramientas AWS SDKs 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 AWS SDKs y herramientas globales— Cómo utilizar los `credentials` archivos `config` y variables de entorno compartidos para configurar sus AWS SDKs propias herramientas.
- Autenticación y acceso: uso AWS SDKs y herramientas— Establece cómo se autentica tu código o herramienta AWS cuando desarrollas con Servicios de AWS ellos.
- AWS SDKs y referencia de configuración de herramientas: referencia para todos los ajustes estandarizados disponibles para la autenticación y la configuración.
- AWS Bibliotecas de Common Runtime (CRT)— Descripción general de las bibliotecas compartidas de AWS Common Runtime (CRT) que están disponibles para casi todos. SDKs
- AWS SDKs Política de mantenimiento de herramientas 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 para el SDK o la herramienta que esté utilizando debe utilizarse además de la información que se presenta aquí. Los siguientes son el SDK y las herramientas, que incluyen secciones de material relevantes en esta guía:

Si utiliza:	Las secciones relevantes de esta guía para usted son:
• Cualquier SDK o herramienta	AWS SDKs Política de mantenimiento de herramientas y herramientas
• AWS Cloud9 • AWS CDK	Configuración AWS SDKs y herramientas globales

Si utiliza:	Las secciones relevantes de esta guía para usted son:
• Kit de herramientas de AWS para Azure DevOps • 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	Autenticación y acceso: uso AWS SDKs y herramientas AWS SDKs Política de mantenimiento de herramientas y herramientas
• AWS CLI • AWS SDK para C++ • AWS SDK para Go • AWS SDK para Java • AWS SDK para JavaScript • AWS SDK para Kotlin • AWS SDK para .NET • AWS SDK para PHP • AWS SDK para Python (Boto3) • AWS SDK para Ruby • AWS SDK para Rust • AWS SDK para Swift • AWS Tools for Windows PowerShell	Configuración AWS SDKs y herramientas globales Autenticación y acceso: uso AWS SDKs y herramientas AWS SDKs y referencia de configuración de herramientas AWS Bibliotecas de Common Runtime (CRT) AWS SDKs Política de mantenimiento de herramientas y herramientas AWS SDKs y ciclo de vida de las versiones de Tools

- Para obtener una descripción general de las herramientas que pueden ayudarle a desarrollar aplicaciones AWS, consulte Herramientas sobre las que desarrollar aplicaciones AWS.
- Para obtener más información sobre el soporte, consulte el Centro de conocimiento de AWS.
- Para conocer AWS la terminología, consulte el AWS glosario de la Glosario de AWS Referencia.

Recursos para desarrolladores

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, consulte What is Amazon Q Developer? en la Guía del usuario de Amazon Q Developer.

Notificación de telemetría de kits 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 de su IDE. Los complementos y extensiones del IDE Amazon Q permiten la asistencia de IA generativa en su IDE. Para obtener información detallada sobre cada uno de los kits de herramientas del IDE, consulte estas guías de usuario del kit de herramientas en la tabla precedente. Para obtener más información sobre el uso de Amazon Q en su IDE, consulte el tema Uso de Amazon Q en el IDE de la guía para desarrolladores de Amazon Q.

AWS IDE Toolkits y Amazon Q pueden recopilar y almacenar datos de telemetría del lado del cliente para tomar decisiones informadas sobre futuras versiones de Toolkit AWS y Amazon Q. Los datos recopilados cuantifican su uso del AWS kit de herramientas y Amazon Q.

Para obtener más información sobre los datos de telemetría recopilados en todos los kits de herramientas del AWS IDE y Amazon Q, consulte el documento CommonDefinitions.json en el repositorio de Github. `aws-toolkit-common`

Para obtener información detallada sobre los datos de telemetría recopilados por cada uno de los kits de herramientas del AWS IDE y las extensiones de Amazon Q, consulte los documentos de recursos en los siguientes AWS repositorios del kit de herramientas: GitHub

- AWS Kit de herramientas de Visual Studio con Amazon Q
- AWS Toolkit for Visual Studio Code y la extensión Amazon Q para VS Code

- AWS Toolkit for JetBrains y el complemento Amazon Q para JetBrains
- Amazon Q para Eclipse

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 AWS SDKs y herramientas globales

Con AWS SDKs otras herramientas para AWS desarrolladores, como AWS Command Line Interface (AWS CLI), puede interactuar con el AWS servicio APIs. 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 ella. 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, las herramientas AWS SDKs y las herramientas admiten configuraciones comunes de fuentes primarias más allá del propio código. Esta sección abarca los siguientes temas:

Temas

Uso de **`credentials`** archivos **`config`** y compartidos para configurar AWS SDKs y herramientas de forma global

Los `credentials` archivos AWS `config` y compartidos son la forma más común de especificar la autenticación y la configuración de un AWS SDK o una herramienta.

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

Tanto los archivos compartidos `config` como `credentials` son archivos de texto sin formato que contienen únicamente caracteres ASCII (codificados en UTF-8). Adoptan la forma de lo que generalmente se denomina archivos INI.

Profiles

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 ajustes de configuración y aplicarlas en diferentes entornos de desarrollo.

El perfil `[default]` contiene los valores que utiliza un SDK o una operación de 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 la situación.

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

Establezca un perfil con nombre

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

Configure esta funcionalidad mediante los siguientes ajustes:

`AWS_PROFILE`- variable de entorno

Cuando esta variable de entorno se establece en un perfil con nombre o «predeterminado», todos los AWS CLI comandos y códigos del SDK 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 del sistema JVM

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

> **Note**
>
> Si su aplicación está en un servidor que ejecuta varias aplicaciones, le recomendamos que utilice siempre perfiles con nombre en lugar del perfil predeterminado. El perfil predeterminado lo selecciona automáticamente cualquier AWS aplicación del entorno y lo comparte entre ellas. Por lo tanto, si otra persona actualiza el perfil predeterminado de su aplicación, puede afectar involuntariamente a los demás. Para evitarlo, defina un perfil con nombre en el archivo `config` compartido y, a continuación, utilice ese perfil con nombre en

su aplicación configurándolo en su código. Puede usar la variable de entorno o la propiedad del sistema JVM para establecer el perfil con nombre si sabe que su alcance solo afecta a su aplicación.

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 sección `profile`.

En el siguiente ejemplo, se muestra un archivo `config` con un perfil `[default]`. Establece la configuración `region`. Todos los ajustes que sigan esta línea, hasta que se encuentre otra definición de sección, se incluirán en 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 sección `profile` es una agrupación de configuración con nombre que se puede aplicar a diferentes escenarios de desarrollo. Para conocer mejor los perfiles con nombre, consulte la sección anterior sobre Perfiles.

El siguiente ejemplo muestra un archivo `config` con una línea de definición de sección `profile` y un perfil con nombre llamado `foo`. Todos los ajustes que sigan esta línea, hasta que se encuentre otra definición de sección, se incluirán en 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 indica un grupo de ajustes que se utilizan para configurar un perfil con el que resolver AWS las credenciales AWS IAM Identity Center. Para obtener más información sobre la configuración de la autenticación de inicio de sesión único, consulte Uso del Centro de identidades de IAM para autenticar el AWS SDK y las herramientas. 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>`.

En el siguiente ejemplo, se configura un perfil que obtendrá AWS credenciales a corto plazo para el rol de IAM en la cuenta «111122223333" mediante un token de «my-sso». SampleRole 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 característica está disponible para tu SDK, consulta Support by AWS SDKs and tools para encontrar los puntos de conexión específicos del servicio.

La línea de definición de la `services` sección indica un grupo de ajustes que configuran puntos de enlace personalizados para las solicitudes. Servicio de AWS 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>`.

Además, la `services` sección está separada en subsecciones por `<SERVICE> =` líneas, donde `<SERVICE>` está la Servicio de AWS clave identificadora. El Servicio de AWS identificador se basa en el modelo de la API, 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 definición de `services` para configurar el punto de conexión que se utilizará únicamente en las solicitudes realizadas únicamente al servicio Amazon DynamoDB . La sección `"local-dynamodb"` de `services` se menciona en la sección `profile` por su nombre mediante la clave `services`. La clave del Servicio de AWS identificador es. `dynamodb` La subsección de Amazon DynamoDB 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 nombre exclusivo del perfil entre corchetes. El siguiente ejemplo muestra un archivo `credentials` con una línea de definición de sección de perfil llamada `foo`.

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

En el archivo `credentials` 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 archivo `config` compartido, le recomendamos que mantenga estos valores confidenciales en un archivo `credentials` 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 de `aws_access_key_id`, `aws_secret_access_key` y `aws_session_token`.

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

Independientemente de si utiliza un perfil específico o "`default`" en el archivo `config`, cualquier configuración que aparezca aquí se combinará con la configuración del archivo `credentials` 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.

Buscar y cambiar la ubicación de los **`credentials`** archivos compartidos **`config`** AWS SDKs y las herramientas

Los `credentials` archivos AWS `config` y compartidos son archivos de texto sin formato que contienen información de configuración de las herramientas AWS SDKs y. 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 su propia computadora 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 crea 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 de inicio

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

- Inicia la ruta
- Va seguido inmediatamente por / o un separador específico de la plataforma. En Windows, tanto ~/ como ~\ se resuelven en el directorio de inicio.

Al determinar el directorio de inicio, 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 entorno `HOMEDRIVE` y `HOMEPATH` (`$HOMEDRIVE$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`).

Cambiar la ubicación predeterminada de estos archivos

Puede 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 acerca de la configuración del sistema con variables de entorno, consulte Uso de variables de entorno para configurar AWS SDKs y herramientas de forma global.

Uso de las propiedades del sistema JVM

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

- Propiedad del sistema JVM del archivo `config`: **`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 establecer propiedades del sistema JVM”. El SDK para Java 1.x no admite estas propiedades del sistema.

Uso de variables de entorno para configurar AWS SDKs y herramientas de forma global

Las variables de entorno proporcionan otra forma de especificar las opciones de configuración y las credenciales al utilizar AWS SDKs las herramientas. Las variables de entorno pueden ser útiles para crear scripts o configurar temporalmente un perfil con nombre como predeterminado. Para ver la lista de variables de entorno compatibles con la mayoría SDKs, 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 los `credentials` archivos AWS `config` AND compartidos.
- Si especifica una configuración mediante un parámetro de la línea de AWS CLI comandos, anulará 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 de variables de AWS Lambda entorno en la Guía para AWS Lambda desarrolladores.

En Serverless Framework, a menudo puede configurar las variables de entorno del SDK en el archivo `serverless.yml`, en la clave del proveedor, en la pestaña de 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.

Uso de las propiedades del sistema JVM para configurar AWS SDK para Java globalmente y AWS SDK para Kotlin

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

Prioridad de las opciones

- Si especifica una configuración mediante el uso de la propiedad de su sistema JVM, esto anula los valores que se encuentran en las variables del entorno o cualquier valor cargado desde un perfil en los archivos `config` y `credentials` de AWS compartidos.
- Si especifica una configuración mediante su variable de entorno, esta anulará cualquier valor cargado desde un perfil en los archivos `config` y `credentials` de AWS compartidos.

Cómo establecer propiedades del sistema JVM

Puede definir las propiedades del sistema JVM de varias maneras.

En la línea de comando

Establezca las propiedades del sistema JVM en la línea de comandos al invocar el comando `java` 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 establecer varias propiedades del sistema JVM, especifique el conmutador `-D` varias veces.

Con una variable de entorno

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

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 el tiempo de ejecución

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

```
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: uso AWS SDKs y herramientas

Al desarrollar una aplicación AWS del SDK o utilizar AWS herramientas para utilizarla Servicios de AWS, debe establecer la forma en que se autentica su código o herramienta. AWS Puede configurar el acceso programático a AWS los recursos de diferentes maneras, según el entorno en el que se ejecute el código y el AWS acceso del que disponga.

Las siguientes opciones forman parte de la cadena de proveedores de credenciales. Esto significa que, al configurar `credentials` los archivos AWS `config` y recursos compartidos en consecuencia, el AWS SDK o la herramienta detectarán y utilizarán automáticamente ese método de autenticación.

Elección de un método para autenticar el código de aplicación

Elige un método para autenticar las llamadas realizadas AWS por tu aplicación.

¿Está ejecutando código DENTRO de un Servicio de AWS (como Amazon EC2, Lambda, Amazon ECS, Amazon EKS)? CodeBuild

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

¿Su código se encuentra en una instancia de Amazon Elastic Compute Cloud?

Uso de funciones de IAM para autenticar las aplicaciones desplegadas en Amazon EC2: utilizar roles de IAM para ejecutar de forma segura su aplicación en una instancia de Amazon EC2.

¿Su código está en una AWS Lambda función?

Lambda crea un rol de ejecución con permisos mínimos al crear una función de Lambda. A continuación, el AWS SDK o la herramienta utilizan automáticamente la función de IAM asociada a la Lambda en tiempo de ejecución, a través del entorno de ejecución de Lambda.

¿Su código está en Amazon Elastic Container Service (en Amazon EC2 o en AWS Fargate Amazon ECS)?

Roles de IAM para la tarea. Debe crear un rol de tarea y especificarlo en la definición de tareas de Amazon ECS. A continuación, el SDK o la herramienta de AWS utilizan automáticamente el rol de IAM asignado a la tarea en tiempo de ejecución, a través de los metadatos de Amazon ECS.

¿Su código está en Amazon Elastic Kubernetes Service?

Le recomendamos que utilice Amazon EKS Pod Identities.

Nota: Si cree que los roles de IAM para cuentas de servicio (IRSA) podrían adaptarse mejor a sus necesidades específicas, consulte Comparación de Pod Identity de EKS e IRSA en la Guía del usuario de Amazon EKS.

¿Su código se ejecuta en AWS CodeBuild

Consulte Uso de políticas basadas en la identidad para. CodeBuild

¿Está su código en otro Servicio de AWS?

Consulte la guía dedicada a su Servicio de AWS. Cuando ejecutas código AWS, la cadena de proveedores de credenciales del SDK puede obtener y actualizar automáticamente las credenciales por ti.

¿Está creando aplicaciones móviles o aplicaciones web basadas en clientes?

Si va a crear aplicaciones móviles o aplicaciones web basadas en clientes a las que es necesario acceder AWS, cree su aplicación de manera que solicite credenciales de AWS seguridad temporales 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 lugar de ello, los usuarios de la aplicación pueden iniciar sesión con un proveedor de identidades (IdP) externo bien conocido, como Login with Amazon, Facebook, Google o cualquier otro IdP compatible con OpenID Connect (OIDC). Pueden recibir un token de autenticación y, después, cambiarlo por credenciales de seguridad temporales en AWS ese mapa por un rol de IAM con permisos para usar los recursos de su empresa. Cuenta de AWS

Para aprender a configurar esto para su SDK o herramienta, consulte Asumir un rol con identidad web u OpenID Connect para autenticar y herramientas AWS SDKs.

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 usuario de IAM.

¿Está desarrollando y ejecutando el código LOCALMENTE?

Lo recomendamosUso de credenciales de consola para autenticar AWS SDKs y herramientas.

Tras un flujo de autenticación rápido basado en el navegador, genera AWS automáticamente credenciales temporales que funcionan en todas las herramientas de desarrollo locales, como la AWS CLI y Herramientas de AWS para PowerShell . AWS SDKs

Si usa Identity Center para AWS acceder a la cuenta

Utilice el Centro de identidades de IAM para autenticar el AWS SDK y las herramientas si ya tiene acceso a AWS las cuentas que and/or necesita para gestionar el acceso de sus empleados. Como práctica recomendada de seguridad, te recomendamos que utilices AWS Organizations el IAM Identity Center para gestionar el acceso a todas tus cuentas. AWS 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 en cuentas. AWS Para comprobar si su región es compatible con el Centro de Identidad de IAM, consulte los puntos de enlace y las cuotas del Centro de Identidad de Uso del Centro de identidades de IAM para autenticar el AWS SDK y las herramientas IAM en la Referencia general de Amazon Web Services.

Si busca otras formas de autenticarse

Cree un usuario de IAM con menos privilegios con permisos para `sts:AssumeRole` desempeñar su función de destino. A continuación, configure su perfil para que asuma un rol mediante una `source_profile` configuración para ese usuario.

También puede usar credenciales de IAM temporales a través de variables de entorno o del archivo de AWS credenciales compartido. Consulte Uso de credenciales de corta duración para autenticar AWS SDKs y utilizar herramientas.

Nota: Solo en entornos aislados o de aprendizaje, puede considerar la posibilidad de utilizar credenciales de larga duración para AWS SDKs autenticar y utilizar herramientas.

¿Este código se ejecuta en las instalaciones o en una máquina virtual híbrida/bajo demanda (como un servidor que lee o escribe en Amazon S3, o Jenkins que implementa en la nube)?

¿Utiliza certificados de cliente X.509?

Sí, consulte Uso de funciones de IAM en cualquier lugar para AWS SDKs autenticar y utilizar herramientas. Puede usar IAM Roles Anywhere para obtener credenciales de seguridad temporales en IAM para cargas de trabajo como servidores, contenedores y aplicaciones que se ejecutan fuera de ellos. AWS Para utilizar IAM Roles Anywhere, sus cargas de trabajo deben utilizar certificados X.509.

¿Puede el entorno conectarse de forma segura a un proveedor de identidad federado (como Microsoft Entra u Okta) para solicitar credenciales temporales? AWS

Sí: utilice Proveedor de credenciales de proceso

Se utiliza Proveedor de credenciales de proceso para recuperar las credenciales automáticamente en tiempo de ejecución. Estos sistemas pueden utilizar una herramienta auxiliar o un complemento para obtener las credenciales y pueden asumir un rol de IAM entre bastidores al utilizar `sts:AssumeRole`.

No: utilice credenciales temporales inyectadas mediante AWS Secrets Manager

Utilice credenciales temporales inyectadas mediante AWS Secrets Manager. Para ver las opciones para obtener claves de acceso de corta duración, consulte Solicitud de credenciales de seguridad temporales en la Guía del usuario de IAM. Para ver las opciones de almacenamiento de estas credenciales temporales, consulte AWS claves de acceso.

Puede usar estas credenciales para recuperar de forma segura permisos de aplicaciones más amplios desde Secrets Manager, donde se pueden almacenar los secretos de producción o credenciales basadas en roles de larga duración.

¿Estás utilizando una herramienta de terceros que no está incluida AWS?

Utilice la documentación redactada por su proveedor externo para obtener la mejor orientación sobre la obtención de credenciales.

Si un tercero no ha proporcionado la documentación, ¿puede inyectar credenciales temporales de forma segura?

Sí: utilice variables de entorno y AWS STS credenciales temporales.

No: utilice claves de acceso estáticas almacenadas en el administrador de secretos cifrados (último recurso).

Métodos de autenticación

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

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

- Uso de funciones de IAM para autenticar las aplicaciones desplegadas en Amazon EC2: utilizar roles de IAM para ejecutar de forma segura su aplicación en una instancia de Amazon EC2.
- Puede interactuar mediante programación mediante el IAM Identity Center de las siguientes maneras: AWS
  - Se utiliza 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 va a crear aplicaciones móviles o aplicaciones web basadas en clientes a las que es necesario acceder AWS, cree su aplicación de manera que solicite credenciales de AWS seguridad temporales 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 lugar de ello, los usuarios de la aplicación pueden iniciar sesión con un proveedor de identidades (IdP) externo bien conocido, como Login with

Amazon, Facebook, Google o cualquier otro IdP compatible con OpenID Connect (OIDC). Pueden recibir un token de autenticación y, después, cambiarlo por credenciales de seguridad temporales en AWS ese mapa por un rol de IAM con permisos para usar los recursos de su empresa. Cuenta de AWS

Para aprender a configurar esto para su SDK o herramienta, consulte Asumir un rol con identidad web u OpenID Connect para autenticar y herramientas AWS SDKs.

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 usuario de IAM.

Métodos de autenticación para el código que se ejecuta de forma local (no interna en AWS)

- Uso de credenciales de consola para autenticar AWS SDKs y herramientas— Esta función funciona tanto con la interfaz de línea de AWS comandos como con las herramientas PowerShell y le proporciona credenciales actualizables que funcionan en todas las herramientas de desarrollo local, como la AWS CLI, las herramientas para PowerShell y. AWS
- Uso del Centro de identidades de IAM para autenticar el AWS SDK y las herramientas— Como práctica recomendada de seguridad, le recomendamos que la utilice AWS Organizations junto con IAM Identity Center para gestionar el acceso en todas sus instalaciones. Cuentas de AWS Puede crear usuarios en Microsoft Active Directory AWS IAM Identity Center, usar un proveedor de identidades (IdP) de SAML 2.0 o federar individualmente su IdP a. Cuentas de AWS Para comprobar si su región es compatible con el IAM Identity Center, consulte los puntos de conexión de AWS IAM Identity Center y las cuotas en Referencia general de Amazon Web Services.
- Uso de funciones de IAM en cualquier lugar para AWS SDKs autenticar y utilizar herramientas— Puede utilizar IAM Roles Anywhere para obtener credenciales de seguridad temporales en IAM para cargas de trabajo como servidores, contenedores y aplicaciones que se ejecutan fuera de ellas. AWS Para utilizar IAM Roles Anywhere, sus cargas de trabajo deben utilizar certificados X.509.
- Asumir un rol con AWS credenciales para autenticarse AWS SDKs y herramientas— Puedes asumir una función de IAM para acceder temporalmente a AWS recursos a los que, de otro modo, no tendrías acceso.
- Uso de claves de AWS acceso para autenticar AWS SDKs y herramientas— Otras opciones que podrían resultar menos prácticas o que podrían aumentar el riesgo de seguridad de sus AWS recursos.

Más información sobre la administración de acceso

La guía del usuario de IAM contiene la siguiente información sobre el control seguro del acceso a AWS los recursos:

- Identidades de IAM (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 al desarrollar aplicaciones AWS de acuerdo con el modelo de responsabilidad compartida.

Referencia general de Amazon Web Services tiene los conceptos básicos sobre lo siguiente:

- Comprender y obtener sus credenciales de AWS: opciones de claves de acceso y prácticas de gestión tanto para el acceso por consola como programático.

Complemento de propagación de identidades de confianza (TIP) de IAM Identity Center para acceder a Servicios de AWS

- Uso del complemento TIP para acceder Servicios de AWS— Si está creando una aplicación para Amazon Q Business u otro servicio que permita la propagación de identidades de forma fiable y utiliza el AWS SDK para Java o el AWS SDK para JavaScript, puede utilizar el complemento TIP para disfrutar de una experiencia de autorización simplificada.

ID de creador de AWS

El tuyo ID de creador de AWS complementa cualquier otra que ya Cuentas de AWS tengas o quieras crear. Si bien a Cuenta de AWS actúa como contenedor de los AWS recursos que usted crea y proporciona un límite de seguridad para esos recursos, usted lo ID de creador de AWS representa como individuo. Puedes iniciar sesión con tu cuenta ID de creador de AWS para acceder a herramientas y servicios para desarrolladores, como Amazon Q y Amazon CodeCatalyst.

- Inicia ID de creador de AWS sesión con la Guía del AWS Sign-In usuario: aprende a crear y usar una ID de creador de AWS y descubre qué proporciona el Builder ID.
- CodeCatalystconceptos: ID de creador de AWS en la Guía del CodeCatalyst usuario de Amazon: aprenda cómo se CodeCatalyst usa un ID de creador de AWS.

Uso de credenciales de consola para autenticar AWS SDKs y herramientas

El uso de credenciales de consola es el método recomendado para proporcionar AWS credenciales al desarrollar una AWS aplicación en el entorno local o en otros entornos de servicios no AWS informáticos. Si estás desarrollando en un AWS recurso, como Amazon Elastic Compute Cloud (Amazon EC2) AWS CloudShell o, te recomendamos que obtengas las credenciales de ese servicio.

También puede autenticarse a través del Centro de Identidad de IAM. Uso del Centro de identidades de IAM para autenticar el AWS SDK y las herramientas Esta opción es una forma habitual de que las organizaciones administren el acceso de sus empleados y requiere que Identity Center esté habilitado.

¿Cómo funciona?

El inicio de sesión para el desarrollo AWS local con las credenciales de la consola le permite usar las credenciales de inicio de sesión de AWS Management Console existentes para el acceso programático a los AWS servicios. Tras un flujo de autenticación basado en un navegador, AWS genera credenciales temporales que funcionan en todas las herramientas de desarrollo locales, como la AWS CLI, Tools for PowerShell y. AWS SDKs Esta función simplifica el proceso de configuración y administración de las credenciales de AWS CLI, especialmente si prefiere la autenticación interactiva en lugar de administrar las claves de acceso a largo plazo.

Con este proceso, puede autenticarse con sus credenciales raíz creadas durante la configuración inicial de la cuenta, con los usuarios de IAM o con una identidad federada de su proveedor de identidad.

Si las utilizas SDKs para el desarrollo, los clientes del SDK utilizarán las credenciales temporales a través del. AWS SDKs y Herramientas: proveedores de credenciales estandarizadas También puede configurar elProveedor de credenciales de inicio de sesión.

Tanto la AWS CLI como las herramientas admiten la autenticación mediante el comando login para PowerShell:

- Inicie sesión para el desarrollo AWS local con las credenciales de la consola
- Inicie sesión con las credenciales de la consola que aparecen en la guía Herramientas de AWS para PowerShell del usuario

Uso del Centro de identidades de IAM para autenticar el AWS SDK y las herramientas

AWS IAM Identity Center se puede utilizar para proporcionar AWS credenciales al desarrollar una AWS aplicación en entornos de servicios no AWS informáticos. 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.

Utilice la autenticación del IAM Identity Center si ya utiliza el Identity Center para acceder a la AWS cuenta o si necesita gestionar el acceso de una organización.

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 SDK/tool autenticación.
- 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 mediante 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.

Requisitos previos

Antes de comenzar este procedimiento, debe haber completado lo siguiente:

- Si no tienes una Cuenta de AWS, regístrate para obtener una Cuenta de AWS.
- Si aún no ha activado el IAM Identity Center, active el IAM Identity Center según las instrucciones de la Guía del usuario de AWS IAM Identity Center

Configuración del acceso mediante programación mediante el IAM Identity Center

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

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

No he establecido el acceso a través de IAM Identity Center

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

   Siga el mismo procedimiento de configuración del acceso de los usuarios con el directorio predeterminado de IAM Identity Center, pero esta vez:

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

   Cuando haya terminado, tendrá lo siguiente:

   - Un grupo `Dev team`.
   - Un conjunto de permisos `PowerUserAccess` adjunto al grupo `Dev team`.
   - Su usuario agregado al grupo `Dev team`.
3. Salga del portal e inicie sesión de nuevo para ver sus opciones Cuentas de AWS y para `Administrator` o`PowerUserAccess`. Seleccione `PowerUserAccess` cuando trabaje con el SDK/las herramientas.

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 su administrador de la nube le ha concedido permisos `PowerUserAccess` (desarrollador), verá las Cuentas de AWS a las que tiene acceso y su 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: Configurar SDKs y usar las herramientas para usar el IAM Identity Center

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

   a. Consulte Instalar o actualizar 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. Aparece los roles para el usuario. Elija Claves de acceso para obtener

credenciales de acceso a la línea de comandos o mediante programación para el conjunto de permisos apropiado. 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 `Issuer URL` y `SSO Region` que necesita para el próximo paso. Nota: Se puede usar `SSO Start URL` indistintamente con `Issuer URL`.

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.

   a. En la petición `SSO Start URL`, introduzca el valor que obtuvo para `Issuer URL`.

   b. Para el nombre del perfil CLI, le recomendamos que lo introduzca *`default`* al empezar. Para obtener información sobre cómo configurar perfiles no predeterminados (con nombre) y su variable de entorno asociada, consulte Profiles.

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

   b. Si anteriormente configuró el acceso a AWS, revise el AWS `credentials` archivo compartido para ver si existe 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 Comprender la cadena de proveedores de credenciales.

Para obtener información detallada sobre cómo las herramientas SDKs y herramientas utilizan y actualizan las credenciales con esta configuración, consulteCómo se resuelve la autenticación de IAM Identity Center AWS SDKs y sus herramientas.

Para configurar los ajustes del proveedor de IAM Identity Center directamente en el archivo `config` compartido, consulte Proveedor de credenciales del IAM Identity Center en esta guía.

Actualización de las sesiones de acceso al portal

El acceso eventualmente caducará y los SDK o las herramientas detectarán un error de autenticación. El momento en que se produce este vencimiento depende de la duración de las sesiones configuradas. 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 la 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: configurar la duración de las sesiones del portal de acceso de AWS de sus usuarios
- Duración de la sesión establecida por permisos: establecer la duración de la sesión

Cómo se resuelve la autenticación de IAM Identity Center AWS SDKs y sus herramientas

Términos relevantes del IAM Identity Center

Los siguientes términos le ayudan a entender el proceso y la configuración subyacentes en AWS IAM Identity Center. La documentación del AWS SDK APIs utiliza nombres diferentes a los de IAM Identity Center 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 IAM Identity Center	Nombre de la API del SDK	Description (Descripción)
Centro de identidades	`sso`	Aunque se haya cambiado el nombre de AWS Single Sign-On, los espacios de nombres

Nombre del IAM Identity Center	Nombre de la API del SDK	Description (Descripción)
		de las `sso` API mantendrá n su nombre original por motivos de compatibilidad con versiones anteriores. Para más información, consulte IAM Identity Center renam en la Guía del usuario de AWS IAM Identity Center .
Consola del IAM Identity Center Consola administrativa		La consola que se utiliza para configurar el inicio de sesión único.
AWS URL del portal de acceso		Una URL exclusiva de su cuenta del IAM Identity Center, como `https://xxx.awsapps.com/start` . Inicie sesión en este portal con sus credenciales de inicio de sesión del IAM Identity Center.
Sesión del Portal de Acceso del IAM Identity Center	Sesión de autenticación	Proporciona un token de acceso al portador al intermedi ario.
Sesión del conjunto de permisos		La sesión de IAM que el SDK usa internamente para realizar las Servicio de AWS llamadas. En las discusiones informale s, es posible que vea que esto se denomina incorrectamente “sesión de roles”.

Nombre del IAM Identity Center	Nombre de la API del SDK	Description (Descripción)
Credenciales de configuración de permisos	AWS credenciales credenciales de sigv4	Las credenciales que el SDK utiliza realmente para la mayoría de las Servicio de AWS llamadas (específi camente, todas las Servicio de AWS llamadas sigv4). 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 Servicios de AWS son sigv4 APIs, 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 Servicio de AWS llamadas mediante el código de la aplicación. AWS IAM Identity Center

Inicie una sesión en el portal de AWS acceso

- Inicie el proceso iniciando sesión con sus credenciales.
  - Utilice el `aws sso login` comando de 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. AWS CLI También actualiza un archivo JSON de caché de SSO con un nuevo token de acceso y un token de actualización y lo pone a disposición para su uso. SDKs
- Si ya tienes 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 AWS acceso de los usuarios en la Guía del AWS IAM Identity Center 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 Servicio de AWS

SDKs proporcionan acceso Servicios de AWS cuando se crea una instancia de un objeto de cliente por servicio. Cuando el perfil seleccionado del AWS `config` archivo compartido está configurado para la resolución de credenciales del Centro de Identidad de IAM, el Centro de Identidad de IAM 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 sigv4 APIs 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 AWS acceso al SDK al asumir una función 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 Definir la duración de la sesión en la Guía del usuario de AWS IAM Identity Center .
- Tenga en cuenta que las credenciales del conjunto de permisos también se denominan credenciales y AWS credenciales sigv4 en la mayoría de la documentación de la API AWS del SDK.

Las credenciales del conjunto de permisos se devuelven de una llamada a la API getRoleCredentialsdel Centro de Identidad de IAM al SDK. El objeto de cliente del SDK utiliza esa supuesta función de IAM para realizar llamadas al Servicio de AWS, por ejemplo, pedir a Amazon S3 que incluya los buckets de 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 Configuración del proveedor de token de SSO, el token de acceso por hora obtenido del IAM Identity Center 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 IAM Identity Center compara el token de actualización con la duración de la sesión del portal de acceso al IAM Identity Center. Si el token de actualización no ha caducado, el IAM Identity Center 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 IAM Identity Center 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 IAM Identity Center. 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 IAM Identity Center 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 IAM Identity Center. Este token se concede si la sesión del portal de acceso al IAM Identity Center no ha caducado.

Uso de funciones de IAM en cualquier lugar para AWS SDKs autenticar y utilizar herramientas

Puede utilizar IAM Roles Anywhere para obtener credenciales de seguridad temporales en IAM para cargas de trabajo como servidores, contenedores y aplicaciones que se ejecutan fuera de ellas.

AWS Para utilizar IAM Roles Anywhere, 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 IAM Roles Anywhere como su proveedor de credenciales.

Paso 1: Configurar IAM Roles Anywhere

IAM Roles Anywhere proporciona 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 IAM Roles Anywhere.

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

> **Note**
>
> Un perfil en la IAM Roles Anywhere User Guide hace referencia a un concepto exclusivo del servicio de IAM Roles Anywhere. No está relacionado con los perfiles del archivo compartido. AWS `config`

Paso 2: Utilice IAM Roles Anywhere

Para obtener credenciales de seguridad temporales de IAM Roles Anywhere, utilice la herramienta ayudante de credenciales de IAM Roles Anywhere. La herramienta de credenciales implementa el proceso de firma de IAM Roles Anywhere.

Para obtener instrucciones sobre cómo descargar la herramienta auxiliar de credenciales, consulte Obtener credenciales de seguridad temporales de AWS Identity and Access Management Roles Anywhere en la guía del usuario de IAM Roles Anywhere.

Para utilizar las credenciales de seguridad temporales de IAM Roles Anywhere AWS SDKs y AWS CLI, puede `credential_process` configurar los ajustes del archivo compartido. AWS `config` SDKs Y son AWS CLI 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 guion, pero el nombre de la configuración contiene un guion 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 la Consola de administración de AWS. 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 de ayuda, consulte IAM Roles Anywhere Credential Helper 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 con AWS credenciales para autenticarse AWS SDKs y herramientas

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 una Servicio de AWS u otra Cuenta de AWS. Para más información acerca de los roles de IAM, consulte Uso de 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.

> **Note**
>
> Se AWS recomienda utilizar puntos de enlace regionales siempre que sea posible y configurar los suyosRegión de AWS.

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. Por lo general, este tipo de asumir un rol se utiliza cuando se tienen credenciales de AWS para una cuenta, pero la aplicación necesita acceder a los recursos de otra cuenta.

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.

Cuando lo especificas en un perfil, el SDK o la herramienta realiza automáticamente la llamada a la AWS STS `AssumeRole`API correspondiente por ti. Para recuperar y usar credenciales temporales asumiendo un rol, especifique los siguientes valores de configuración en el AWS `config` archivo compartido. 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: `credential_source` o `source_profile`
- (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
credential_source = Ec2InstanceMetadata
```

```
[profile-with-user-that-can-assume-role]
aws_access_key_id=AKIAIOSFODNN7EXAMPLE
aws_secret_access_key=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
aws_session_token=IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZVERYLONGSTRI

[profile dev]
region = us-east-1
output = json
role_arn = arn:aws:iam::123456789012:role/my-role-name
source_profile = profile-with-user-that-can-assume-role
role_session_name = my_session
```

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

Asumir un rol con identidad web u OpenID Connect para autenticar y herramientas AWS SDKs

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 proveedor de identidad web o una federación OpenID Connect(OIDC) o SAML. Para obtener más información sobre los roles de IAM, consulte Métodos para asumir un rol de IAM en la Guía del usuario de IAM.

Una vez configurada la función de IAM en tu SDK, si esa función está configurada para confiar en tu proveedor de identidades, puedes configurar aún más tu SDK para que asuma esa función a fin de obtener AWS credenciales temporales.

> **Note**
>
> Se recomienda utilizar puntos de enlace regionales siempre que sea posible y configurar los suyos. AWS Región de AWS

Cómo federar con identidad web u OpenID Connect

Puedes usar los JSON Web Tokens (JWTs) de proveedores de identidad públicos, como Login With Amazon, Facebook o Google, para obtener AWS credenciales temporales`AssumeRoleWithWebIdentity`. Según cómo se usen, JWTs pueden denominarse tokens de ID o tokens de acceso. También puede utilizar proveedores de identidad JWTs emitidos por proveedores de identidad (IdPs) que sean compatibles con el protocolo de detección de la OIDC, como EntraId o. PingFederate

Si utiliza Amazon Elastic Kubernetes Service, esta característica permite especificar diferentes roles de IAM para cada uno de sus cuentas de servicio en un clúster de Amazon EKS. Esta función de Kubernetes se distribuye JWTs a tus pods, que luego este proveedor de credenciales los utiliza para obtener credenciales temporales. AWS 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

Para configurar la federación con un IdP externo, utilice un proveedor de identidades de IAM para informar AWS sobre el IdP externo y su configuración. Esto establece la confianza entre su IdP Cuenta de AWS y el externo. Antes de configurar el SDK para usar el JSON Web Token (JWT) 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 o de 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 web JSON (JWT) AWS STS para la autenticación.

Cuando lo especificas en un perfil, el SDK o la herramienta realiza automáticamente la llamada a la AWS STS `AssumeRoleWithWebIdentity`API correspondiente. Para recuperar y usar credenciales temporales mediante la federación de identidades web, especifique los siguientes valores de configuración en el AWS `config` archivo compartido. 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 `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 herramientas SDKs 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 AWS SDKs, 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.

Uso de claves de AWS acceso para autenticar AWS SDKs y herramientas

El uso de claves de AWS acceso es una opción de autenticación cuando se utilizan AWS SDKs herramientas.

Use credenciales a corto plazo

Recomendamos configurar su SDK o herramienta para utilizar Uso del Centro de identidades de IAM para autenticar el AWS SDK y las herramientas para usar opciones de duración de sesión ampliada.

Sin embargo, para configurar directamente las credenciales temporales del SDK o de la herramienta, consulte Uso de credenciales a corto plazo para autenticar AWS SDKs y herramientas.

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 todas partes 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 AWS instalaciones. 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 AWS Con uno de estos enfoques, puede 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. Esto es diferente al de un usuario de IAM, que es una credencial de larga duración que se puede compartir y que podría aumentar el riesgo de seguridad de sus recursos de AWS .

Cree 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 estés aprendiendo, pero te recomendamos que evites 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.
- Ejecute scripts, trabajos y otros procesos automatizados programados que no admitan un proceso de inicio de sesión asistido por una persona como parte de su aprendizaje.

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.

Asegurar claves de acceso para un usuario de IAM

Debe rotar con regularidad las claves de acceso de usuario de IAM. Siga las instrucciones en Rotating access keys en la Guía de usuario de IAM. Si cree que ha compartido accidentalmente sus claves de acceso de usuario de IAM, rote las claves de acceso.

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 ayudarte a no enviar información confidencial a un repositorio de Git de forma inadvertida. Para obtener más información acerca de los usuarios de IAM, consulte Identidades de IAM (usuarios, grupos y funciones) en la Guía de usuario de IAM.

Para configurar un usuario de IAM para empezar, consulte Uso de credenciales a largo plazo para autenticar AWS SDKs y herramientas.

Uso de credenciales a corto plazo para autenticar AWS SDKs y herramientas

Recomendamos configurar el AWS SDK o la herramienta para utilizarlos Uso del Centro de identidades de IAM para autenticar el AWS SDK y las herramientas con opciones de duración de sesión prolongada. Sin embargo, puede copiar y usar las credenciales temporales que están disponibles en el portal de AWS 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 administrar manualmente las claves de acceso y un token del archivo de credenciales, recomendamos que la aplicación utilice credenciales temporales enviadas desde:

- Un servicio de AWS cómputo, como ejecutar la aplicación en Amazon Elastic Compute Cloud o en AWS Lambda.
- Otra opción de la cadena de proveedores de credenciales, como Uso del Centro de identidades de IAM para autenticar el AWS SDK y las herramientas.
- O utilícela Proveedor de credenciales de proceso para recuperar credenciales temporales.

Configure un archivo de credenciales con las credenciales de corta duración recuperadas del portal de AWS 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 el SDK o la herramienta utilizan si no se especifica un perfil con nombre específico.
4. Inicie sesión en el portal de AWS acceso.
5. Siga estas instrucciones para actualizar manualmente las credenciales de las credenciales para copiar las credenciales del rol de IAM del portal de AWS acceso.

   a. Para el paso 4 de las instrucciones vinculadas, elija el nombre del rol de IAM que le concede acceso para sus necesidades de desarrollo. Este rol suele tener un nombre similar PowerUserAccessal de Desarrollador.

   b. Para el paso 7 de las instrucciones vinculadas, seleccione la opción Agregar manualmente un perfil al su archivo de credenciales de AWS y copie 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 el SDK cree un cliente de servicio, accederá a estas credenciales temporales y las utilizará para cada solicitud. La configuración del rol de IAM elegido en el paso 5a determina el tiempo de validez de las credenciales temporales. La duración máxima es de doce horas.

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

Uso de credenciales a largo plazo para autenticar AWS SDKs y herramientas

> **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.

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

Advertencias y directrices importantes para las credenciales

Advertencias para las credenciales

- NO use las credenciales raíz de la cuenta para obtener acceso a los recursos de AWS . Estas credenciales proporcionan acceso ilimitado a la cuenta y son difíciles de revocar.
- NO incluya claves de acceso literales ni información sobre credenciales en sus archivos de aplicación. Si lo hace, puede crear un riesgo de exposición accidental de sus credenciales si, por ejemplo, carga el proyecto en un repositorio público.
- NO incluya archivos que contengan credenciales en el área de su proyecto.
- Tenga en cuenta que todas las credenciales almacenadas en el AWS `credentials` archivo compartido se almacenan en texto sin formato.

Guía adicional para administrar las credenciales de forma segura

Para obtener información general sobre cómo administrar las AWS credenciales de forma segura, consulte Prácticas recomendadas para administrar las claves de AWS acceso en el Referencia general de AWS. Además de esa conversación, tenga en cuenta lo siguiente:

- Use roles de IAM para tareas para tareas de Amazon Elastic Container Service (Amazon ECS).

- Use roles de IAM para aplicaciones que se ejecutan en instancias de Amazon EC2.

Requisitos previos: crear una cuenta AWS

Para utilizar un usuario de IAM para acceder a AWS los servicios, necesita una AWS cuenta y AWS unas credenciales.

1. Cree una cuenta.

   Para crear una AWS cuenta, consulte Primeros pasos: ¿es la primera vez AWS que lo usa? en la Guía AWS Account Management 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, como se explica en Crear un usuario administrativo en la Guía del usuario de IAM.

   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 el desarrollo AWS o la ejecución de aplicaciones AWS. Como buena práctica, debe crear usuarios, conjuntos de permisos o roles de servicio que sean adecuados para estas tareas. Para obtener más información, consulte Aplicar permisos de privilegio mínimo en la Guía del usuario de IAM.

Paso 1: Crear el usuario de IAM

- Siga el procedimiento Creación de usuarios de IAM (consola) de la Guía del usuario de IAM para crear su usuario de IAM. Al crear su usuario de IAM:
  - Le recomendamos que seleccione Proporcionar acceso de usuario a Consola de administración de AWS. Esto le permite ver Servicios de AWS relacionados con el código que está ejecutando en un entorno visual, como comprobar los registros de diagnóstico de AWS CloudTrail o cargar archivos en Amazon Simple Storage Service, lo que resulta útil a la hora de depurar el código.
  - Para Establecer permisos - Opciones de permiso, elija Adjuntar políticas directamente para indicar cómo desea asignar permisos a este usuario.

- La mayoría de los tutoriales del SDK “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 de permisos o etiquetas.

Paso 2: Obtener las claves de acceso

1. En el panel de navegación de la consola de IAM, seleccione Usuarios y, a continuación, seleccione el **`User name`** del 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. Para el Paso 1 Crear clave de acceso, elija Interfaz de línea de comandos (CLI) o Código local. Ambas opciones generan el mismo tipo de clave para utilizarla tanto con la AWS CLI como con la SDKs.
4. En el paso 2 de Crear clave de acceso, introduzca una etiqueta opcional y seleccione Siguiente.
5. En el paso 3 de Crear clave de acceso, seleccione Descargar archivo.csv para guardar un archivo `.csv` con la clave de acceso y la clave de acceso secreta de su usuario de IAM. 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. Cree o abra el archivo AWS `credentials` compartido. 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 al 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 funciones de IAM para autenticar las aplicaciones desplegadas en Amazon EC2

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 instancia de Amazon Elastic Compute Cloud.

Para ejecutar la aplicación AWS del SDK en una instancia de Amazon Elastic Compute Cloud, crea una función de IAM y, a continuación, dale a tu EC2 instancia de Amazon acceso a esa función. Para obtener más información, consulte Funciones de IAM para Amazon EC2 en la Guía del EC2 usuario de Amazon.

Creación de un rol de IAM

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

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

1. Inicie sesión en la consola de IAM Consola de administración de AWS y ábrala 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 función de IAM

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

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

Con esta EC2 configuración de IAM y 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 archivo PEM.

Para ello, siga las instrucciones correspondientes a su tipo de instancia: Conexión con la instancia de Linux o Conexión con la instancia de Windows. 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 par de claves, 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 su máquina de desarrollo.
3. (Opcional) Compruebe que la aplicación utilice las credenciales proporcionadas por el rol de IAM.

    a. Inicia sesión en la EC2 consola de Amazon Consola de administración de AWS y ábrela en https://console.aws.amazon.com/ec2/.
    b. Seleccione la instancia.
    c. Elija Acciones, Seguridad y luego, Modificar rol de IAM.
    d. Para el rol de IAM, separe el rol de IAM seleccionando Sin función de IAM.
    e. Elija Actualizar rol de IAM.
    f. Vuelva a ejecutar la aplicación y confirme que devuelve un error de autorización.

Uso del complemento TIP para acceder Servicios de AWS

La propagación fiable de identidades (TIP) es una función AWS IAM Identity Center que permite Servicios de AWS a los administradores conceder permisos en función de los atributos de los usuarios, como las asociaciones de grupos. Con la propagación de identidades fiable, el contexto de identidad se añade a una función de IAM para identificar al usuario que solicita acceso a AWS los recursos. Este contexto se propaga a otros Servicios de AWS.

El contexto de identidad comprende la información que se Servicios de AWS utiliza para tomar decisiones de autorización cuando reciben solicitudes de acceso. Esta información incluye los metadatos que identifican al solicitante (por ejemplo, un usuario del Centro de Identidad de IAM), el lugar Servicio de AWS al que se solicita el acceso (por ejemplo, Amazon Redshift) y el ámbito del acceso (por ejemplo, el acceso de solo lectura). El destinatario Servicio de AWS utiliza este contexto y cualquier permiso asignado al usuario para autorizar el acceso a sus recursos. Para obtener más información, consulte la descripción general de la propagación de identidades confiables de la Guía del AWS IAM Identity Center usuario.

El complemento TIP se puede utilizar con Servicios de AWS este soporte para la propagación de identidades confiable. Como caso de uso de referencia, consulte Configuración de una aplicación de Amazon Q Business mediante AWS IAM Identity Center en la Guía del usuario de Amazon Q Business.

> **Note**
>
> Si utilizas Amazon Q Business, consulta Configuración de una aplicación de Amazon Q Business mediante AWS IAM Identity Center para obtener las instrucciones específicas del servicio.

Requisitos previos para utilizar el complemento de TIP

Los recursos siguientes son necesarios para que el complemento funcione:

1. Debe utilizar el AWS SDK para Java o el AWS SDK para JavaScript.
2. Compruebe que el servicio que está utilizando es compatible con la propagación de identidades de confianza.

   Consulte la columna Permite la propagación de identidades de confianza a través del IAM Identity Center de la tabla de aplicaciones administradas de AWS que se integran con el IAM Identity Center de la Guía del usuario de AWS IAM Identity Center .
3. Puede habilitar IAM Identity Center y utilizarlo únicamente para la propagación de identidades de confianza.

   Consulte los requisitos previos y las consideraciones de la TIP en la Guía del usuario de AWS IAM Identity Center .
4. Debe tener una Identity-Center-integrated solicitud.

   Consulte las aplicaciones administradas de AWS o las aplicaciones administradas por el cliente en la Guía del usuario de AWS IAM Identity Center .
5. Debe configurar un emisor de token de confianza (TTI) y conectar su servicio al IAM Identity Center.

   Consulte los requisitos previos para emisores de tokens de confianza y las tareas para configurar un emisor de token de confianza en la Guía del usuario de AWS IAM Identity Center .

Para usar el complemento de TIP en su código

1. Cree una instancia del complemento de propagación de identidades de confianza.
2. Cree una instancia de cliente de servicio para interactuar con su cliente de servicio Servicio de AWS y personalícelo añadiendo el complemento confiable de propagación de identidades.

El complemento de TIP usa los siguientes parámetros:

- **`webTokenProvider`**: una función que el cliente implementa para obtener un token OpenID de su proveedor de identidad externo.
- **`accessRoleArn`**: el ARN del rol de IAM que debe asumir el complemento con el contexto de identidad del usuario para obtener las credenciales de identidad mejorada.
- **`applicationArn`**: la cadena del identificador único del cliente o de la aplicación. Este valor es un ARN de aplicación que tiene las OAuth concesiones configuradas.
- **`ssoOidcClient`**: (Opcional) Un cliente OIDC de SSO, por ejemplo, `SsoOidcClient`para Java o for JavaScript, con configuraciones definidas `client-sso-oidc`por el cliente. Si no se proporciona, se creará una instancia y se utilizará un cliente OIDC que utilice `applicationRoleArn`.
- **`stsClient`**: (opcional) Un cliente de AWS STS con configuraciones definidas por el cliente, que se utiliza para asumir `accessRoleArn` con el contexto de identidad del usuario. Si no se proporciona, se creará una instancia y se utilizará un AWS STS cliente que `applicationRoleArn` lo utilice.
- **`applicationRoleArn`**: (Opcional) El ARN del rol de IAM que se va a asumir `AssumeRoleWithWebIdentity` para poder iniciar el OIDC AWS STS y los clientes.
  - Si no se proporciona, se deben proporcionar ambos parámetros `ssoOidcClient` y `stsClient`.
  - Si se proporciona, `applicationRoleArn` no puede tener el mismo valor que el parámetro `accessRoleArn`. `applicationRoleArn` se utiliza para crear el stsClient, que se utiliza para asumir accessRole. Si se usa el mismo rol para ambos `applicationRoleaccessRole`, significaría usar un rol para asumir el rol (suposición del rol propio), lo cual no es aconsejable. AWS Consulte el anuncio para obtener más detalles.

Consideraciones para los parámetros **`ssoOidcClient`**, **`stsClient`** y **`applicationRoleArn`**

Al configurar el complemento de TIP, tenga en cuenta los siguientes requisitos de permiso en función de los parámetros que proporcione:

- Si proporciona `ssoOidcClient` y `stsClient`:
  - Las credenciales en `ssoOidcClient` deben tener permiso `oauth:CreateTokenWithIAM` para llamar al centro de identidad y obtener el contexto de usuario específico del centro de identidad.
  - Las credenciales en `stsClient` deben contar con `sts:AssumeRole` y permisos `sts:SetContext` en `accessRole`. `accessRole` también debe configurarse con una relación de confianza con las credenciales activadas en `stsClient`.
- Si proporciona `applicationRoleArn`:
  - `applicationRole` debe tener los permisos `oauth:CreateTokenWithIAM`, `sts:AssumeRole` y `sts:SetContext` necesarios en los recursos requeridos (instancia deIdC, `accessRole`), ya que se utilizará para crear clientes OIDC y STS.
  - `applicationRole`debe tener una relación de confianza con el proveedor de identidad que se utilice para generar el`webToken`, ya que se `webToken` utilizará para asumir el ApplicationRole mediante la AssumeRoleWithWebIdentityllamada del complemento.

Ejemplo de ApplicationRole configuración:

Política de confianza con el proveedor de tokens web:

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

```
                }
            }
        }
    ]
}
```

Política de permisos:

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

Ejemplos de código que utilizan TIP

Los siguientes ejemplos muestran cómo implementar el complemento TIP en su código mediante el AWS SDK para Java o el AWS SDK para JavaScript.

Java

Para usar el complemento TIP en su AWS SDK para Java proyecto, debe declararlo como una dependencia en el `pom.xml` archivo de su proyecto.

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

En su código fuente, incluya la declaración de paquete requerida para `software.amazon.awssdk.trustedidentitypropagation`.

Los siguientes ejemplos muestran dos formas de crear una instancia del complemento de propagación de identidades de confianza y agregarla a un cliente de servicio. Ambos ejemplos utilizan Amazon S3 como servicio y se utilizan `S3AccessGrantsPlugin` para administrar los permisos específicos del usuario, pero se pueden aplicar a cualquiera Servicio de AWS que admita la propagación de identidades confiables (TIP).

> **Note**
>
> Para estos ejemplos, debe configurar los permisos específicos de usuario de S3 Access Grants. Consulte la documentación de concesiones de acceso de S3 para obtener más información.

Opción 1: Cree y transfiera clientes OIDC y STS

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

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

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

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

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

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

Opción 2: Transferir applicationRoleArn y aplazar la creación del cliente al complemento

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

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

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

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

Para obtener información y fuentes adicionales, consulte trusted-identity-propagation-javaen GitHub.

JavaScript

Ejecute el siguiente comando para instalar el paquete de complementos de autenticación TIP en su AWS SDK para JavaScript proyecto:

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

El `package.json` final debería incluir una dependencia similar a la siguiente:

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

En su código fuente, importe la dependencia de `TrustedIdentityPropagationExtension` requerida.

Los siguientes ejemplos muestran dos formas de crear una instancia del complemento de propagación de identidades de confianza y agregarla a un cliente de servicio. Ambos ejemplos utilizan Amazon S3 como servicio y Amazon S3 Access Grants para administrar los permisos específicos de los usuarios, pero se pueden aplicar a cualquiera Servicio de AWS que admita la propagación de identidades confiables (TIP).

> **Note**
>
> Para estos ejemplos, debe configurar los permisos específicos de usuario de concesiones de acceso a Amazon S3; consulte la documentación de concesiones de acceso a Amazon S3 para obtener más información.

Opción 1: Cree y transfiera clientes OIDC y STS

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

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

```
                return 'ID_TOKEN_FROM_YOUR_IDENTITY_PROVIDER';
            },
            ssoOidcClient: customOidcClient,
            stsClient: customStsClient,
            accessRoleArn: accessRoleArn,
            applicationArn: applicationArn,
        }),
    ],
});

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

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

  const credentials = response.Credentials;

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

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

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

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

Opción 2: Transferir applicationRoleArn y aplazar la creación del cliente al complemento

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

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

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

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

  const credentials = response.Credentials;

  const temporaryS3Client = new S3Client({
```

```
    region: "us-east-1",
    credentials: {
      accessKeyId: credentials.AccessKeyId,
      secretAccessKey: credentials.SecretAccessKey,
      sessionToken: credentials.SessionToken,
    },
  });

  const s3Params = {
    Bucket: "BUCKET_NAME",
    Key: "S3_OBJECT_KEY",
  };
  const getObjectCommand = new GetObjectCommand(s3Params);
  const s3Object = await temporaryS3Client.send(getObjectCommand);

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

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

Para obtener información y fuentes adicionales, consulte trusted-identity-propagation-jsen GitHub.

AWS SDKs y referencia de configuración de herramientas

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

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

- AWS SDKs y Herramientas: proveedores de credenciales estandarizadas— Los proveedores de credenciales más comunes están estandarizados en varios SDKs.
- AWS SDKs y funciones estandarizadas de Tools— Características comunes estandarizadas en varios SDKs.

Cómo crear clientes de servicio

Para acceder mediante programación Servicios de AWS, SDKs utilice un cliente class/object para cada una de ellas. Servicio de AWS Por ejemplo, si tu aplicación necesita acceder a Amazon EC2, crea un objeto EC2 cliente de Amazon para interactuar con ese servicio. A continuación, utiliza el cliente de servicio para realizar solicitudes al mismo Servicio de AWS. En la mayoría de los casos SDKs, 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 todos los ámbitos. 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. En el caso del parámetro AWS CLI o Herramientas de AWS para PowerShell, estos parámetros adoptan la forma de parámetros por operación que se introducen en la línea de comandos. En el caso de un SDK, las asignaciones explícitas pueden adoptar la forma de un parámetro que se establece al crear una instancia de un Servicio de AWS cliente o un objeto de configuración o, a veces, al llamar a una API individual.

2. Solo en Java/Kotlin: la propiedad del sistema JVM para 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. El SDK verifica la configuración en el archivo de `credentials` compartido. Si está configurado, el cliente lo usa.
5. El archivo `config` compartido para la configuración. Si la configuración está presente, el SDK la usa.
   - La variable de entorno `AWS_PROFILE` o la propiedad del sistema JVM `aws.profile` se pueden utilizar para personalizar el perfil que carga el SDK.
6. Los valores predeterminados proporcionados por el código de origen del SDK 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, AWS SDK para .NET admite una fuente adicional llamada SDK Store. Para obtener más información sobre los proveedores exclusivos de un SDK o una herramienta, consulta la guía específica del SDK o la herramienta que estés utilizando.

El orden determina qué métodos tienen prioridad y sustituyen a los demás. Por ejemplo, si configuras un perfil en el archivo compartido `config`, solo se encuentra y se usa después de que el SDK o la herramienta comprueben 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`. Por último, una configuración en la operación individual

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

Cómo comprender las páginas de configuración de esta guía

Las páginas de la sección 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 (para Java y Kotlin SDKs) los ajustes de la JVM que se pueden utilizar fuera del código para configurar la función. Cada tema vinculado de cada lista lleva a la página de configuración correspondiente.

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

Cada característica 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 se puede establecer el valor al agregar el ajuste a un archivo de configuración, al establecer una variable de entorno o (solo para Java y Kotlin) al ajustar una propiedad del sistema JVM. Cada configuración muestra todos los métodos admitidos para establecer el valor en un bloque situado por 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 características Compresión de solicitudes.

La información de la configuración del ejemplo `disable_request_compression` documenta lo siguiente:

- Hay tres formas equivalentes de controlar la compresión de las solicitudes fuera del código base. Puede:
  - Establecerlo en su archivo de configuración con `disable_request_compression`

- Establecerlo como una variable de entorno con `AWS_DISABLE_REQUEST_COMPRESSION`
- O bien, si se utiliza el SDK de Java o Kotlin, configúrelo como una propiedad del sistema JVM mediante `aws.disableRequestCompression`

> **Note**
>
> También puede haber una forma de configurar la misma funcionalidad directamente en el código, pero esta referencia no cubre este tema, ya que es exclusiva de cada SDK. Si quiere establecer su configuración en el propio código, consulte la guía específica del SDK o la referencia de la API.

- 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 tabla de Support by AWS SDKs and tools.

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

- `Yes`: la configuración es totalmente compatible con el SDK tal como está escrito.
- `Partial`: algunas de las configuraciones son compatibles o el comportamiento se aparta de la descripción. Para `Partial`, 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 muestran en la siguiente tabla se pueden asignar en el AWS `config` archivo compartido. Son globales y afectan a todos Servicios de AWS. SDKs y las herramientas también pueden admitir configuraciones y variables de entorno únicas. Para ver la configuración y las variables de entorno que solo admiten un SDK o una herramienta individual, consulte esa guía de SDK o de herramientas específica.

Nombre del conjunto	Details
`account_id_endpoint_mode`	Puntos finales basados en cuentas
`api_versions`	Ajustes de configuración general
`auth_scheme_preference`	Esquema de autenticación
`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_host_prefix_injection`	Inyección de prefijos de host

Nombre del conjunto	Details
disable_request_compression	Compresión de solicitudes
duration_seconds	Asumir el rol de proveedor de credenciales
ec2_metadata_service_endpoint	Proveedor de credenciales IMDS
ec2_metadata_service_endpoint_mode	Proveedor de credenciales IMDS
ec2_metadata_v1_disabled	Proveedor de credenciales IMDS
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

Nombre del conjunto	Details	
metadata_service_timeout	Metadatos de EC2 instancias de Amazon	
mfa_serial	Asumir el rol de proveedor de credenciales	
output	Ajustes de configuración general	
parameter_validation	Ajustes de configuración general	
region	Región de AWS	
request_checksum_calculation	Protecciones de integridad de datos para Amazon S3	
request_min_compression_size_bytes	Compresión de solicitudes	
response_checksum_validation	Protecciones de integridad de datos para Amazon S3	
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_express_session_auth	Autenticación de sesión S3 Express One Zone	

Nombre del conjunto	Details
`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
`sigv4a_signing_region_set`	Esquema de autenticación
`source_profile`	Asumir el rol de proveedor de credenciales
`sso_account_id`	Proveedor de credenciales del IAM Identity Center
`sso_region`	Proveedor de credenciales del IAM Identity Center
`sso_registration_scopes`	Proveedor de credenciales del IAM Identity Center
`sso_role_name`	Proveedor de credenciales del IAM Identity Center
`sso_start_url`	Proveedor de credenciales del IAM Identity Center
`sts_regional_endpoints`	AWS STS Puntos de conexión regionales
`use_dualstack_endpoint`	Puntos de conexión de doble pila y FIPS

Nombre del conjunto	Details
`use_fips_endpoint`	Puntos de conexión de doble pila y FIPS
`web_identity_token_file`	Asumir el rol de proveedor de credenciales

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

Los ajustes que se indican en la siguiente tabla se pueden asignar al AWS `credentials` archivo compartido. Son globales y afectan a todos Servicios de AWS. SDKs y las herramientas también pueden admitir configuraciones y variables de entorno únicas. Para ver la configuración y las variables de entorno que solo admiten un SDK o una herramienta individual, consulte esa guía de SDK o de herramientas específica.

Nombre del conjunto	Details
`aws_access_key_id`	AWS claves de acceso
`aws_secret_access_key`	AWS claves de acceso
`aws_session_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. SDKs y las herramientas también pueden admitir configuraciones y variables de entorno únicas. Para ver la configuración y las variables de entorno que solo admiten un SDK o una herramienta individual, consulte esa guía de SDK o de herramientas específica.

Nombre del conjunto	Details
AWS_ACCESS_KEY_ID	AWS claves de acceso
AWS_ACCOUNT_ID	puntos finales basados en cuentas
AWS_ACCOUNT_ID_ENDPOINT_MODE	Puntos finales basados en cuentas
AWS_AUTH_SCHEME_PREFERENCE	Esquema de autenticación
AWS_CA_BUNDLE	Ajustes de configuración general
AWS_CONFIG_FILE	Buscar y cambiar la ubicación de los `credentials` archivos compartidos `config` y las herramientas AWS SDKs
AWS_CONTAINER_AUTHORIZATION_TOKEN	Proveedor de credenciales de contenedor
AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE	Proveedor de credenciales de contenedor
AWS_CONTAINER_CREDENTIALS_FULL_URI	Proveedor de credenciales de contenedor
AWS_CONTAINER_CRED	Proveedor de credenciales de contenedor

Nombre del conjunto	Details	
ENTIALS_R ELATIVE_URI		
AWS_DEFAU LTS_MODE	Valores predeterminados de configuración inteligente	
AWS_DISAB LE_HOST_P REFIX_INJ ECTION	Inyección de prefijos de host	
AWS_DISAB LE_REQUES T_COMPRESSION	Compresión de solicitudes	
AWS_EC2_M ETADATA_D ISABLED	Proveedor de credenciales IMDS	
AWS_EC2_M ETADATA_S ERVICE_EN DPOINT	Proveedor de credenciales IMDS	
AWS_EC2_M ETADATA_S ERVICE_EN DPOINT_MODE	Proveedor de credenciales IMDS	
AWS_EC2_M ETADATA_V 1_DISABLED	Proveedor de credenciales IMDS	
AWS_ENABL E_ENDPOIN T_DISCOVERY	Detección de puntos de conexión	

Nombre del conjunto	Details
AWS_ENDPOINT_URL	Puntos de conexión específicos del servicio
AWS_ENDPOINT_URL_<SERVICE>	Puntos de conexión específicos del servicio
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	Uso de `credentials` archivos `config` y compartidos para configurar AWS SDKs y utilizar herramientas a nivel mundial
AWS_REGION	Región de AWS
AWS_REQUEST_CHECKSUM_CALCULATION	Protecciones de integridad de datos para Amazon S3

Nombre del conjunto	Details
AWS_REQUEST_MIN_COMPRESSION_SIZE_BYTES	Compresión de solicitudes
AWS_RESPONSE_CHECKSUM_VALIDATION	Protecciones de integridad de datos para Amazon S3
AWS_RETRY_MODE	Comportamiento de los reintentos
AWS_ROLE_ARN	Asumir el rol de proveedor de credenciales
AWS_ROLE_SESSION_NAME	Asumir el rol de proveedor de credenciales
AWS_S3_DISABLE_EXPRESS_SESSION_AUTH	Autenticación de sesión S3 Express One Zone
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

Nombre del conjunto	Details
AWS_SHARED_CREDENTIALS_FILE	Buscar y cambiar la ubicación de los `credentials` archivos compartidos `config`AWS SDKs y las herramientas
AWS_SIGV4A_SIGNING_REGION_SET	Esquema de autenticación
AWS_STS_REGIONAL_ENDPOINTS	AWS STS Puntos de conexión regionales
AWS_USE_DUALSTACK_ENDPOINT	Puntos de conexión de doble pila y FIPS
AWS_USE_FIPS_ENDPOINT	Puntos de conexión de doble pila y FIPS
AWS_WEB_IDENTITY_TOKEN_FILE	Asumir el rol de proveedor de credenciales

Lista de propiedades del sistema JVM

Puede utilizar las siguientes propiedades del sistema JVM para AWS SDK para Java y para AWS SDK para Kotlin (orientadas a la JVM). Consulte the section called “Cómo establecer propiedades del sistema JVM” para ver instrucciones sobre cómo configurar las propiedades del sistema JVM.

Nombre del conjunto	Details
aws.accessKeyId	AWS claves de acceso
aws.accountId	puntos finales basados en cuentas

Nombre del conjunto	Details	
`aws.accountIdEndpointMode`	Puntos finales basados en cuentas	
`aws.authSchemePreference`	Esquema de autenticación	
`aws.configFile`	Buscar y cambiar la ubicación de los `credentials` archivos compartidos `config` y las herramientas AWS SDKs	
`aws.defaultsMode`	Valores predeterminados de configuración inteligente	
`aws.disableEc2MetadataV1`	Proveedor de credenciales IMDS	
`aws.disableHostPrefixInjection`	Inyección de prefijos de host	
`aws.disableRequestCompression`	Compresión de solicitudes	
`aws.disableS3ExpressAuth`	Autenticación de sesión S3 Express One Zone	
`aws.ec2MetadataServiceEndpoint`	Proveedor de credenciales IMDS	

Nombre del conjunto	Details	
`aws.ec2MetadataServiceEndpointMode`	Proveedor de credenciales IMDS	
`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`	Uso de `credentials` archivos `config` AND compartidos para configurar AWS SDKs y herramientas de forma global	
`aws.region`	Región de AWS	
`aws.requestChecksumCalculation`	Protecciones de integridad de datos para Amazon S3	
`aws.requestMinCompressionSizeBytes`	Compresión de solicitudes	

Nombre del conjunto	Details	
`aws.responseChecksumValidation`	Protecciones de integridad de datos para Amazon S3	
`aws.retryMode`	Comportamiento de los reintentos	
`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`	Buscar y cambiar la ubicación de los `credentials` archivos compartidos `config`AWS SDKs y las herramientas	
`aws.useDualstackEndpoint`	Puntos de conexión de doble pila y FIPS	
`aws.useFipsEndpoint`	Puntos de conexión de doble pila y FIPS	

Nombre del conjunto	Details
`aws.webId entityTok enFile`	Asumir el rol de proveedor de credenciales
`sdk.ua.appId`	Application ID

AWS SDKs y 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 detalles, consulte su API específica de SDK.

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

Temas

- Comprender la cadena de proveedores de credenciales
- Cadenas de proveedores de credenciales específicas del SDK y de las herramientas
- AWS claves de acceso
- Proveedor de credenciales de inicio de sesión
- Asumir el rol de proveedor de credenciales
- Proveedor de credenciales de contenedor
- Proveedor de credenciales del IAM Identity Center
- Proveedor de credenciales IMDS
- Proveedor de credenciales de proceso

Comprender la cadena de proveedores de credenciales

Todos SDKs tienen una serie de sitios (o fuentes) que consultan para encontrar credenciales válidas que puedan utilizarlas para realizar una solicitud a un proveedor Servicio de AWS. Una vez que se encuentran 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, AWS SDKs siempre se intenta 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 para que el SDK lo haga.

Si bien la cadena distinta que utiliza cada SDK varía, la mayoría de las veces incluye fuentes como las siguientes:

Proveedor de credenciales	Description (Descripción)
AWS claves de acceso	AWS claves de acceso para un usuario de IAM (como`AWS_ACCESS_KEY_ID` , y`AWS_SECRET_ACCESS_KEY`).
Cómo federar con identidad web u OpenID Connect: asumir el rol de proveedor de credenciales	Inicie sesión con un proveedor de identidades (IdP) externo bien conocido, como Login with Amazon, Facebook, Google o cualquier otro IdP compatible con OpenID Connect (OIDC). Asuma los permisos de un rol de IAM mediante un token web JSON (JWT) de AWS Security Token Service ().AWS STS
Proveedor de credenciales de inicio de sesión	Obtenga las credenciales de una sesión de consola nueva o existente en la que haya iniciado sesión.
Proveedor de credenciales del IAM 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 rol de IAM. (Recupere las credenciales temporales para un rol y, a continuación, utilícelas).

Proveedor de credenciales	Description (Descripción)
Proveedor de credenciales de contenedor	Credenciales de Amazon Elastic Kubernetes Service (Amazon EKS) y Amazon Elastic Container Service (Amazon ECS). 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 un origen o proceso externo, incluido IAM Roles Anywhere.
Proveedor de credenciales IMDS	Credenciales del perfil de instancia de Amazon Elastic Compute Cloud (Amazon EC2). Asocie un rol de IAM a cada una de sus instancias de EC2. Las credencia les temporales de ese rol estarán disponibles para el código que se ejecute en la instancia. Las credencia les se entregan a través del servicio de metadatos de Amazon EC2.

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 Uso de `credentials` archivos `config` y compartidos para configurar AWS SDKs y herramientas de forma global. Para obtener más información, consulte Prioridad de los ajustes.

Cadenas de proveedores de credenciales específicas del SDK y de las herramientas

Para ir directamente a los detalles de la cadena de proveedores de credenciales específica de su SDK o herramienta, elija su SDK o herramienta entre las siguientes opciones:

- AWS CLI
- SDK para C++
- SDK para Go
- SDK para Java

- SDK para JavaScript
- SDK para Kotlin
- SDK para .NET
- SDK para PHP
- SDK para Python (Boto3)
- SDK para Ruby
- SDK para Rust
- SDK para Swift
- Herramientas para PowerShell

AWS claves de acceso

> **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.

AWS las claves de acceso de un usuario de IAM se pueden utilizar como AWS credenciales. El AWS SDK utiliza automáticamente estas AWS credenciales para firmar las solicitudes de la API AWS, de modo que sus cargas de trabajo puedan acceder a sus 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 usar credenciales a largo plazo.

> **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 AWS `credentials` archivo compartido es la ubicación recomendada para almacenar la información de las credenciales, ya que se encuentra de forma segura fuera de los directorios de

origen de la aplicación y separado de la configuración específica del SDK del archivo compartido. `config`

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

Configure esta funcionalidad mediante lo siguiente:

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

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

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

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

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

> Especifica un token de AWS sesión que se utiliza 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 Prácticas recomendadas de seguridad en IAM.

Para obtener instrucciones acerca de cómo obtener estos valores, consulte Uso de credenciales a corto plazo para autenticar AWS SDKs y herramientas.

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
```

Support by AWS SDKs and tools

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

SDK	cc e	Notas o más información
AWS CLI v2	Sí	
SDK para C++	Sí	No se admite el archivo compartido `config`.
SDK para Go V2 (1.x)	Sí	
SDK para 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 para Java 2.x	Sí	

SDK	cc e	Notas o más información
SDK para Java 1.x	Sí	
SDK para 3.x JavaScript	Sí	
SDK para 2.x JavaScript	Sí	
SDK para Kotlin	Sí	
SDK para.NET 4.x	Sí	
SDK para .NET 3.x	Sí	
SDK para PHP 3.x	Sí	
SDK para Python (Boto3)	Sí	
SDK para Ruby 3.x	Sí	
SDK para Rust	Sí	
SDK para Swift	Sí	
Herramientas para V5 PowerShell	Sí	
Herramientas para la PowerShell V4	Sí	No se admiten variables de entorno.

Proveedor de credenciales de inicio de sesión

Puede usar sus credenciales de inicio de sesión AWS de Management Console actuales para adquirir credenciales a corto plazo que se pueden usar para el acceso mediante programación. Después de completar el flujo de autenticación basado en el navegador, AWS genera credenciales temporales que funcionan en todas las herramientas de desarrollo locales, como la AWS CLI, AWS Tools for PowerShell y. AWS SDKs

Para generar estas credenciales, ejecute el `aws login` comando en la AWS CLI o el `Invoke-AWSLogin` cmdlet en AWS Tools for. PowerShell Las credenciales a corto plazo resultantes se almacenarán en caché local, donde las podrá reutilizar. AWS SDKs Las credenciales a corto plazo caducan en 15 minutos, pero la CLI las SDKs actualizará automáticamente según sea necesario hasta 12 horas. Cuando el token de actualización caduque, se le pedirá que vuelva a iniciar sesión mediante la CLI o PowerShell.

El comando login actualizará el perfil que especifique con la `login_session` configuración, que almacena la identidad de la sesión de la consola de administración que seleccionó durante el flujo de trabajo de inicio de sesión.

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

De forma predeterminada, las credenciales a corto plazo y el token de actualización se almacenan en un archivo JSON en el `~/.aws/login/cache` directorio en Linux y macOS, o `%USERPROFILE%\.aws\login\cache` en Windows. El nombre del archivo se basa en el nombre de la sesión de inicio de sesión. Puede anular el directorio configurando la variable de `AWS_LOGIN_CACHE_DIRECTORY` entorno.

Configuración del proveedor de inicio de sesión

Configure esta funcionalidad mediante lo siguiente:

`AWS_LOGIN_CACHE_DIRECTORY`: variable de entorno

Directorio alternativo donde la CLI SDKs almacenará las credenciales en caché que se asignan a un perfil de sesión de inicio de sesión.

Valor predeterminado: `~/.aws/login/cache` en Linux y macOS, o `%USERPROFILE%\.aws\login\cache` en Windows.

Support by AWS SDKs and tools

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

SDK	cc e	Notas o más información
AWS CLI v2	Sí	
SDK para C++	Sí	
SDK para Go V2 (1.x)	No	
SDK para Go 1.x (V1)	Sí	
SDK para Java 2.x	Sí	
SDK para Java 1.x	No	
SDK para 3.x JavaScript	Sí	
SDK para 2.x JavaScript	No	
SDK para Kotlin	Sí	
SDK para.NET 4.x	Sí	
SDK para .NET 3.x	Sí	
SDK para PHP 3.x	Sí	
SDK para Python (Boto3)	Sí	Requiere CRT
SDK para Ruby 3.x	Sí	
SDK para Rust	Sí	
Herramientas para V5 PowerShell	Sí	
Herramientas para la PowerShell V4	No	

Asumir el rol de proveedor de credenciales

> **Note**
>
> Para obtener ayuda para comprender el diseño de las páginas de configuración o para interpretar la tabla Support by AWS SDKs and tools que aparece a continuación, consulteCómo comprender las páginas de configuración de esta guía.

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 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 proveedor de identidades web Cuenta de AWS, una federación OIDC o SAML.

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, utilice los siguientes comandos.

Para comenzar a utilizar esta configuración, consulte Asumir un rol con AWS credenciales para autenticarse AWS SDKs y herramientas en esta guía.

Asumir la configuración del proveedor de credenciales de rol

Configure esta funcionalidad mediante lo siguiente:

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

Se utiliza en instancias de Amazon EC2 o en contenedores de Amazon Elastic Container Service para especificar dónde el SDK o la herramienta puede encontrar credenciales que tienen permisos para asumir el rol que especificó con el parámetro `role_arn`.

Valor predeterminado: ninguno.

Valores válidos:

- Entorno: especifica que el SDK o la herramienta va a recuperar las credenciales fuente a partir de las variables de entorno `AWS_ACCESS_KEY_ID` y `AWS_SECRET_ACCESS_KEY`.
- Ec2 InstanceMetadata: especifica que el SDK o la herramienta deben utilizar la función de IAM asociada al perfil de la instancia EC2 para obtener las credenciales de origen.
- EcsContainer— Especifica que el SDK o la herramienta deben utilizar la función de IAM asociada al contenedor Amazon ECS o la función de IAM asociada al contenedor Amazon EKS para obtener las credenciales de origen.

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

Ejemplo de configuración en un archivo `config` para indicar que las credenciales deben proceder de Amazon EC2:

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

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

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 Cómo consultar la configuración de la duración máxima de la sesión para un rol en la Guía del usuario de IAM.

Ejemplo de esta configuración en un archivo `config`:

```
duration_seconds = 43200
```

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

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 sus AWS recursos a un tercero en la Guía del usuario de IAM.

Ejemplo de esta configuración en un archivo `config`:

```
external_id = unique_value_assigned_by_3rd_party
```

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

Especifica la identificación o el número de serie de un dispositivo de autenticación multifactor (MFA) que el usuario debe utilizar al asumir un rol.

Se requiere cuando se asume un rol en el que la política de confianza para ese rol incluye una condición que requiere la autenticación MFA. Para obtener más información sobre la MFA, consulte AWS autenticación multifactor de en IAM en la Guía del usuario de IAM.

Valor predeterminado: ninguno.

Valores válidos: el valor puede ser un número de serie de un dispositivo de hardware (como `GAHT12345678`) o un nombre de recurso de Amazon (ARN) de un dispositivo MFA virtual. El formato del ARN es: `arn:aws:iam::`*`account-id`*`:mfa/`*`mfa-device-name`*

Ejemplo de esta configuración en un archivo `config`:

En este ejemplo, se asume un dispositivo MFA virtual, denominado `MyMFADevice`, que se creó para la cuenta y se habilitó para un usuario.

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

`role_arn`- configuración de AWS **`config`** archivos compartidos, **`AWS_ROLE_ARN`**: variable de entorno, **`aws.roleArn`**- Propiedad del sistema JVM: solo Java/Kotlin

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

Valor predeterminado: ninguno.

Valores válidos: el valor debe ser el ARN de un rol de IAM, con 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 instancia de Amazon EC2 o a una instancia de contenedor de Amazon ECS.
- `web_identity_token_file`: utilizar proveedores de identidades públicos o cualquier proveedor de identidades compatible con OpenID Connect (OIDC) para los usuarios que han sido autenticados en un móvil o una aplicación web.

`role_session_name`- configuración de AWS **`config`** archivos compartidos, **`AWS_ROLE_SESSION_NAME`**: variable de entorno, **`aws.roleSessionName`**- Propiedad del sistema JVM: solo Java/Kotlin

Especifica el nombre que se va a asociar a la sesión de rol. Este nombre aparece en los registros de AWS CloudTrail para las entradas asociadas a esta sesión, que puede resultar útil al realizar auditorías. Para obtener más información, consulte el CloudTrailelemento UserIdentity en la Guía del usuario.AWS CloudTrail

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 la AWS CLI AWS API llama 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`- configuración de AWS **`config`** archivos compartidos

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

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 el SDK encuentra un perfil con credenciales. El encadenamiento de roles limita tu sesión de rol AWS CLI o de AWS API a un máximo de una hora y no se puede aumentar. Para obtener más información, consulte los Términos y conceptos de roles en la Guía del usuario de IAM.

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
role_session_name = ProfileARoleSession

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

En el ejemplo anterior, el perfil `A` indica al SDK o a la herramienta que busque automáticamente las credenciales del perfil `B` vinculado. En este caso, el perfil `B` utiliza la herramienta de ayudante de credenciales proporcionada por Uso de funciones de IAM en cualquier lugar para AWS SDKs autenticar y utilizar herramientas para obtener las credenciales de AWS SDK. Estas credenciales temporales las utiliza el código para acceder a los recursos de AWS . El rol especificado debe tener políticas de permisos de IAM adjuntas que permitan ejecutar el código solicitado, como el comando o el método Servicio de AWS de API. Cada acción que realiza el perfil `A` incluye el nombre de la sesión del rol en CloudTrail los registros.

Para ver un segundo ejemplo de encadenamiento de roles, puede usar la siguiente configuración si tiene una aplicación en una instancia de Amazon Elastic Compute Cloud y desea que esa aplicación asuma otro rol.

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

[profile B]
credential_source=Ec2InstanceMetadata
```

El perfil A utilizará las credenciales de la instancia de Amazon EC2 para asumir el rol especificado y las renovará de manera automática.

`web_identity_token_file`- configuración de AWS **`config`** archivos compartidos, **`AWS_WEB_IDENTITY_TOKEN_FILE`**: variable de entorno, **`aws.webIdentityTokenFile`**- Propiedad del sistema JVM: solo 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. El SDK o la herramienta para desarrolladores carga el contenido de este archivo y lo pasa como argumento `WebIdentityToken` cuando llama a la operación `AssumeRoleWithWebIdentity` en su 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.

Support by AWS SDKs and tools

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

SDK	cc e	Notas o más información
AWS CLI v2	Sí	
SDK para C++	Parci	`credential_source` no admitido. `duration_seconds` no admitido. `mfa_serial` no admitido.
SDK para Go V2 (1.x)	Sí	
SDK para 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 para Java 2.x	Parci	`mfa_serial` no es compatible. `duration_seconds` no es compatible.
SDK para Java 1.x	Parci	`credential_source` no es compatible. `mfa_serial` no es compatible. No se admiten las propiedades del sistema JVM.
SDK para 3.x JavaScript	Sí	
SDK para 2.x JavaScript	Parci	`credential_source` no admitidas.
SDK para Kotlin	Sí	
SDK para .NET 4.x	Sí	
SDK para .NET 3.x	Sí	
SDK para PHP 3.x	Sí	
SDK para Python (Boto3)	Sí	
SDK para Ruby 3.x	Sí	
SDK para Rust	Sí	
SDK para Swift	Sí	

SDK	cc e	Notas o más información
Herramientas para V5 PowerShell	Sí	
Herramientas para V4 PowerShell	Sí	

Proveedor de credenciales de contenedor

> **Note**
>
> Para obtener ayuda para comprender el diseño de las páginas de configuración o para interpretar la tabla Support by AWS SDKs and tools que aparece a continuación, consulteCómo comprender las páginas de configuración de esta guía.

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 (Amazon ECS) y Amazon Elastic Kubernetes Service (Amazon EKS). SDKs intente cargar las credenciales desde el punto de enlace HTTP especificado mediante una solicitud GET.

Si utiliza Amazon ECS, le recomendamos que utilice un rol de IAM 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 Rol de IAM para la tarea en la Guía para desarrolladores de Amazon Elastic Container Service.

Si utiliza Amazon EKS, le recomendamos que utilice Amazon EKS Pod Identity para mejorar el aislamiento, los privilegios mínimos, la auditabilidad, el funcionamiento independiente, la reutilización y la escalabilidad de las credenciales. Tanto el Pod como el rol de IAM están asociados a una cuenta de servicio de Kubernetes para administrar las credenciales de las aplicaciones. Para obtener más información sobre Amazon EKS Pod Identity, consulte Amazon EKS Pod Identities en la Guía del usuario de Amazon EKS.. 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 para obtener las credenciales. SDKs Para obtener información sobre la configuración, consulte Configuración del Amazon EKS Pod Identity Agent en la Guía del usuario de Amazon EKS o Amazon EKS Pod Identity simplifica los permisos de IAM para las aplicaciones en los clústeres de Amazon EKS en el sitio AWS web del blog.

Configure esta funcionalidad mediante lo siguiente:

`AWS_CONTAINER_CREDENTIALS_FULL_URI`: variable de entorno

Especifica el punto de conexión de la URL HTTP completo para que el SDK lo utilice al realizar una solicitud de credenciales. Esto incluye tanto el esquema como el host.

Valor predeterminado: ninguno.

Valores válidos: URI válido.

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 punto de conexión de la URL HTTP relativa para que el SDK lo utilice al realizar una solicitud de credenciales. El valor se añade al nombre de host predeterminado de Amazon ECS de `169.254.170.2`.

Valor predeterminado: ninguno

Valores válidos: URI relativa válida.

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, el SDK configurará el encabezado de autorización de la solicitud HTTP 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
```

Support by AWS SDKs and tools

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

SDK	cc e	Notas o más información
AWS CLI v2	Sí	
SDK para C++	Sí	
SDK para Go V2 (1.x)	Sí	
SDK para Go 1.x (V1)	Sí	
SDK para Java 2.x	Sí	Cuando Lambda SnapStart está activada `AWS_CONTA INER_CREDENTIALS_FULL_URI` y se `AWS_CONTA INER_AUTHORIZATION_TOKEN` utilizan automáticamente para la autenticación.
SDK para Java 1.x	Sí	Cuando Lambda SnapStart está activada `AWS_CONTA INER_CREDENTIALS_FULL_URI` y se `AWS_CONTA INER_AUTHORIZATION_TOKEN` utilizan automáticamente para la autenticación.
SDK para 3.x JavaScript	Sí	
SDK para 2.x JavaScript	Sí	
SDK para Kotlin	Sí	
SDK para .NET 4.x	Sí	Cuando Lambda SnapStart está activada `AWS_CONTA INER_CREDENTIALS_FULL_URI` y se `AWS_CONTA INER_AUTHORIZATION_TOKEN` utilizan automáticamente para la autenticación.
SDK para .NET 3.x	Sí	Cuando Lambda SnapStart está activada `AWS_CONTA INER_CREDENTIALS_FULL_URI` y se `AWS_CONTA INER_AUTHORIZATION_TOKEN` utilizan automáticamente para la autenticación.
SDK para PHP 3.x	Sí	

SDK	cc e	Notas o más información
SDK para Python (Boto3)	Sí	Cuando Lambda SnapStart está activada `AWS_CONTAINER_CREDENTIALS_FULL_URI` y se `AWS_CONTAINER_AUTHORIZATION_TOKEN` utilizan automáticamente para la autenticación.
SDK para Ruby 3.x	Sí	
SDK para Rust	Sí	
SDK para Swift	Sí	
Herramientas para la versión PowerShell 5	Sí	
Herramientas para V4 PowerShell	Sí	

Proveedor de credenciales del IAM Identity Center

> **Note**
>
> Para obtener ayuda para comprender el diseño de las páginas de configuración o para interpretar la tabla Support by AWS SDKs and tools que aparece a continuación, consulteCómo comprender las páginas de configuración de esta guía.

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

> **Note**
>
> En la documentación de la API del AWS SDK, el proveedor de credenciales del IAM Identity Center se denomina proveedor de credenciales de SSO.

Tras activar el Centro de identidades de IAM, debe definir un perfil para su configuración en el archivo compartido. AWS `config` Este perfil se utiliza para conectarse al portal de acceso a IAM Identity Center. Cuando un usuario se autentica correctamente en IAM Identity Center, el portal devuelve las credenciales de corta duración para el rol de IAM asociado a ese usuario. Para saber cómo el SDK obtiene las credenciales temporales de la configuración y las utiliza para las Servicio de AWS solicitudes, consulteCómo se resuelve la autenticación de IAM Identity Center AWS SDKs y sus herramientas.

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

- (Recomendado) Configuración del proveedor de tokens de SSO: duraciones de sesión prolongadas. Incluye soporte para duraciones de sesión personalizadas.
- 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.

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

- AWS IAM Identity Center Guía del usuario
- AWS IAM Identity Center Referencia de la API del portal

Para obtener información detallada sobre cómo las herramientas SDKs y herramientas utilizan y actualizan las credenciales con esta configuración, consulteCómo se resuelve la autenticación de IAM Identity Center AWS SDKs y sus herramientas.

Requisitos previos

Primero debe activar el IAM Identity Center. Para más detalles sobre la activación de la autenticación en el IAM Identity Center, consulte Habilitación AWS IAM Identity Center en la Guía del usuario de AWS IAM Identity Center .

> **Note**
>
> Como alternativa, para conocer todos los requisitos previos y la configuración del archivo `config` compartidos necesaria que se detalla en esta página, consulte las instrucciones de configuración detalladas Uso del Centro de identidades de IAM para autenticar el AWS SDK y las herramientas.

Configuración del proveedor de token de SSO

Cuando utilizas la configuración del proveedor de token de SSO, el AWS SDK o la herramienta actualizan automáticamente la sesión hasta que se prolongue el período de sesión. Para obtener más información sobre la duración y la duración máxima de la sesión, consulte Configurar la duración de la sesión del portal de AWS acceso y de las aplicaciones integradas del IAM Identity Center en la Guía del AWS IAM Identity Center usuario.

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

En el siguiente ejemplo de archivo `config` compartido se configura el SDK o la herramienta mediante un perfil `dev` para que solicite credenciales del IAM Identity Center.

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

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

En los siguientes ejemplos, se muestra cómo se define una sección `sso-session` y asóciela a un perfil. Normalmente, `sso_account_id` se `sso_role_name` debe configurar en la `profile` sección para que el SDK pueda solicitar AWS credenciales. `sso_region`,`sso_start_url`, y `sso_registration_scopes` debe configurarse dentro de la `sso-session` sección.

No obstante, `sso_account_id` y `sso_role_name` no son necesarios para todos los escenarios de configuración de token de SSO. Si su aplicación solo utiliza Servicios de AWS ese soporte de autenticación de portador, no necesitará AWS las credenciales tradicionales. La autenticación de portador es un esquema de autenticación HTTP que utiliza tokens de seguridad denominados tokens de portador. En este escenario, no se necesitan `sso_account_id` ni `sso_role_name`. Consulte la Servicio de AWS guía individual para determinar si el servicio admite la autorización de token al portador.

Los ámbitos de registro se configuran como parte de un `sso-session`. El alcance es un mecanismo de OAuth 2.0 para limitar el acceso de una aplicación a la cuenta de un usuario. El anterior ejemplo establece `sso_registration_scopes` para proporcionar acceso para enumerar cuentas y roles.

En los siguientes ejemplos, se muestra cómo reutilizar la misma configuración `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
```

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 Configuración del proveedor de token de SSO 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`.

Configuración del proveedor de credenciales del IAM Identity Center

Configure esta funcionalidad mediante lo siguiente:

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

La URL que apunta a la URL del portal de acceso o a la URL emisora del IAM Identity Center de su organización. Para obtener más información, consulte Uso del portal de acceso de AWS en la Guía de usuario de AWS IAM Identity Center .

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

- Como alternativa, a partir de la versión 2.22.0 del AWS CLI, puedes utilizar el valor de la URL del AWS emisor.

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

El Región de AWS que contiene el host del portal del Centro de Identidad de IAM; es decir, la región que seleccionó antes de activar el Centro de Identidad de IAM. Es independiente de la AWS región predeterminada y puede ser diferente.

Para obtener una lista completa de ellos 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 del IAM Identity Center, consulte el panel de control y busque la región.

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

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

Para ver la lista de cuentas disponibles, vaya a la consola del IAM Identity Center y abra la página Cuentas de AWS. También puedes ver la lista de cuentas disponibles mediante el método ListAccountsAPI en la Referencia de API del AWS IAM Identity Center portal. Por ejemplo, puedes llamar al AWS CLI método list-accounts.

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

El nombre de un conjunto de permisos aprovisionado como rol de IAM que define los permisos resultantes que tiene el usuario. El rol debe existir en el lugar Cuenta de AWS especificado por`sso_account_id`. Utilice el nombre de la función, no el Nombre de recurso de Amazon (ARN) de la función.

Los conjuntos de permisos tienen adjuntas políticas de IAM y políticas de permisos personalizadas y definen el nivel de acceso que los usuarios tienen a su Cuentas de AWS asignada.

Para ver la lista de conjuntos de permisos disponibles por cada uno Cuenta de AWS, vaya a la consola del IAM Identity Center y abra la 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 método ListAccountRolesAPI en la Referencia de API del AWS IAM Identity Center portal. Por ejemplo, puedes llamar al AWS CLI método list-account-roles.

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

Una lista delimitada por comas de los ámbitos válidos que deben autorizarse para la `sso-session`. Una solicitud puede pedir uno o varios ámbitos y el token de acceso emitido a la solicitud se limita a los ámbitos concedidos. Para recuperar un token de actualización del servicio del IAM Identity Center, se debe conceder un límite mínimo de `sso:account:access`. Para ver una lista de las opciones de ámbito de acceso compatibles, consulte los Ámbitos de acceso en la Guía del usuario de AWS IAM Identity Center .

Estos ámbitos definen los permisos cuya autorización se solicita para el cliente OIDC registrado y los tokens de acceso recuperados por el cliente. Los ámbitos autorizan el acceso a los puntos de conexión autorizados por el token de portador del Centro de identidades de IAM.

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.

Support by AWS SDKs and tools

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

SDK	cc e	Notas o más información
AWS CLI v2	Sí	
SDK para C++	Sí	
SDK para Go V2 (1.x)	Sí	
SDK para 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 para Java 2.x	Sí	Los valores de configuración también se admiten en el archivo `credentials` .
SDK para Java 1.x	No	
SDK para 3.x JavaScript	Sí	
SDK para 2.x JavaScript	Sí	
SDK para Kotlin	Sí	
SDK para .NET 4.x	Sí	
SDK para .NET 3.x	Sí	
SDK para PHP 3.x	Sí	

SDK	cc e	Notas o más información
SDK para Python (Boto3)	Sí	
SDK para Ruby 3.x	Sí	
SDK para Rust	Parci	Solo configuración heredada no actualizable.
SDK para Swift	Sí	
Herramientas para V5 PowerShell	Sí	
Herramientas para la PowerShell V4	Sí	

Proveedor de credenciales IMDS

> **Note**
>
> Para obtener ayuda para comprender el diseño de las páginas de configuración o para interpretar la tabla Support by AWS SDKs and tools que aparece a continuación, consulteCómo comprender las páginas de configuración de esta guía.

El servicio de metadatos de instancias (IMDS) son datos sobre una instancia que se pueden utilizar para configurar o administrar la instancia en ejecución. Para obtener más información, consulte Trabajo con metadatos de la instancia en la Guía del usuario de Amazon EC2. Amazon EC2 proporciona un punto de conexión 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. SDKs Pueden 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 del 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 error HTTP 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 el servicio de metadatos de instancias (IMDS) de Amazon EC2 para obtener credenciales.

Valor predeterminado: `false`.

Valores válidos:

- **`true`**: no utilice el IMDS para obtener credenciales.
- **`false`**: utilice el IMDS para obtener las credenciales.

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

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

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

Valor predeterminado: `false`.

Valores válidos:

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

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

El tipo de punto de conexión. Este valor anula la ubicación predeterminada en la que AWS los SDK y las herramientas buscarán los metadatos de las instancias de Amazon EC2.

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: URI válido.

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

El modo de punto de conexión de IMDS.

Valor predeterminado:`IPv4`.

Valores válidos: `IPv4`, `IPv6`.

> **Note**
>
> El proveedor de credenciales IMDS forma parte del Comprender la cadena de proveedores de credenciales. Sin embargo, el proveedor de credenciales IMDS 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 IMDS en el código. Puede especificar las fuentes de credenciales directamente al crear clientes de servicio.

Seguridad de credenciales IMDS

De forma predeterminada, cuando el AWS SDK no está configurado con credenciales válidas, el SDK intentará utilizar el Amazon EC2 Instance Metadata Service (IMDS) para recuperar las credenciales de un rol. AWS 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 servicio de metadatos de instancias Amazon EC2.

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

Cómo inhabilitar el uso de las credenciales IMDS de Amazon EC2

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.

Support by AWS SDKs and tools

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

SDK	cc e	Notas o más información
AWS CLI v2	Sí	
SDK para C++	Sí	
SDK para Go V2 (1.x)	Sí	
SDK para 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 para Java 2.x	Sí	
SDK para Java 1.x	Parci	Propiedades del sistema JVM: se utiliza `com.amazonaws.sdk.disableEc2MetadataV1` en lugar de `aws.disableEc2MetadataV1`; `aws.ec2MetadataServiceEndpointMode` y `aws.ec2MetadataServiceEndpoint` no se admiten.
SDK para 3.x JavaScript	Sí	
SDK para 2.x JavaScript	Sí	
SDK para Kotlin	Sí	No utiliza la opción IMDSv1 alternativa.
SDK para.NET 4.x	Sí	
SDK para .NET 3.x	Sí	
SDK para PHP 3.x	Sí	
SDK para Python (Boto3)	Sí	

SDK	cc e	Notas o más información
SDK para Ruby 3.x	Sí	
SDK para Rust	Sí	No utiliza la opción IMDSv1 alternativa.
SDK para Swift	Sí	
Herramientas para V5 PowerShell	Sí	Puede deshabilitar la opción IMDSv1 alternativa de forma explícita en el código mediante. `[Amazon.Util.EC2InstanceMetadata]::EC2MetadataV1Disabled = $true`
Herramientas para la versión PowerShell 4	Sí	Puede deshabilitar la opción IMDSv1 alternativa de forma explícita en el código mediante`[Amazon.Util.EC2InstanceMetadata]::EC2MetadataV1Disabled = $true`.

Proveedor de credenciales de proceso

> **Note**
>
> Para obtener ayuda para comprender el diseño de las páginas de configuración o para interpretar la tabla Support by AWS SDKs and tools que aparece a continuación, consulteCómo comprender las páginas de configuración de esta guía.

SDKs proporcionan 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 en las instalaciones o integrarlas con su proveedor de identificación en las instalaciones.

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 Uso de funciones de IAM en cualquier lugar para AWS SDKs autenticar y utilizar herramientas.

> **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 se basa en un recurso AWS informático, utilice otros proveedores de credenciales. Si usa esta opción, debe asegurarse de que el archivo config esté lo más bloqueado posible siguiendo las mejores prácticas de seguridad para su sistema operativo. Confirme que su herramienta de credenciales personalizada no escriba información secreta en ella`StdErr`, ya que AWS CLI puede capturarla SDKs y registrarla, lo que podría exponerla a usuarios no autorizados.

Configure esta funcionalidad mediante lo siguiente:

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

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

> **Note**
>
> El proveedor de credenciales del proceso forma parte del Comprender la 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 una ruta a un programa que el SDK o la herramienta de desarrollo ejecutan en su 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 dedicado:

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

Salida válida del programa de credenciales

El SDK ejecuta el comando tal y como se especifica en el perfil y luego lee datos de la secuencia de salida estándar. El comando que especifique, ya se trate de una secuencia de comandos o de un programa binario, debe generar una salida JSON en STD0UT que se ajuste a 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 clave `Expiration` no está presente en la salida de la herramienta, el SDK da por hecho que las credenciales son credenciales a largo plazo 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**
>
> El SDK no almacena en caché credenciales de procesos externos de la forma que lo hace con las credenciales de asunción 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.

Support by AWS SDKs and tools

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

SDK	cc e	Notas o más información
AWS CLI v2	Sí	
SDK para C++	Sí	
SDK para Go V2 (1.x)	Sí	
SDK para 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 para Java 2.x	Sí	
SDK para Java 1.x	Sí	
SDK para 3.x JavaScript	Sí	
SDK para 2.x JavaScript	Sí	
SDK para Kotlin	Sí	
SDK para.NET 4.x	Sí	
SDK para .NET 3.x	Sí	
SDK para PHP 3.x	Sí	
SDK para Python (Boto3)	Sí	
SDK para Ruby 3.x	Sí	
SDK para Rust	Sí	

SDK	cc e	Notas o más información
SDK para Swift	Sí	
Herramientas para V5 PowerShell	Sí	
Herramientas para la PowerShell V4	Sí	

AWS SDKs y funciones estandarizadas de Tools

Muchas funciones se han estandarizado con valores predeterminados consistentes y para que funcionen de la misma manera en muchas SDKs de ellas. 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. Consulta la API específica del SDK para obtener más información.

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

Temas

- Puntos de conexión basados en cuentas
- Application ID
- Metadatos de la instancia de Amazon EC2
- Puntos de acceso de Amazon S3
- Puntos de acceso multirregión de Amazon S3
- Autenticación de sesión de S3 Express One Zone
- Esquema de autenticación
- Región de AWS
- AWS STS Puntos finales regionales
- Protecciones de integridad de datos para Amazon S3
- Puntos de conexión de doble pila y FIPS

- Detección de puntos de conexión
- Ajustes de configuración general
- Inyección de prefijos de host
- Cliente IMDS
- Comportamiento de los reintentos
- Compresión de solicitudes
- Puntos de conexión específicos del servicio
- Valores predeterminados de configuración inteligente

Puntos de conexión basados en cuentas

> **Note**
>
> Para obtener ayuda para comprender el diseño de las páginas de configuración o para interpretar la tabla Support by AWS SDKs and tools que aparece a continuación, consulteCómo comprender las páginas de configuración de esta guía.

Los puntos de conexión basados en cuentas ayudan a garantizar un alto rendimiento y escalabilidad, ya que utilizan el ID de cuenta de Cuenta de AWS para enrutar las solicitudes para los servicios compatibles con esta característica. Cuando utiliza un SDK de AWS y un servicio que admiten puntos de conexión basados en cuentas, el cliente del SDK crea y utiliza un punto de conexión basado en una cuenta en lugar de un punto de conexión regional. Si el cliente del SDK no ve el ID de la cuenta, este utilizará el punto de conexión regional. Los puntos de enlace basados en cuentas adoptan la forma de`https://<account-id>.ddb.<region>.amazonaws.com`, dónde *`<account-id>`* y dónde *`<region>`* están su Cuenta de AWS ID y. Región de AWS

Configure esta funcionalidad mediante lo siguiente:

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

El Cuenta de AWS ID. Se utiliza para el enrutamiento de puntos de conexión basado en cuentas. Un ID de Cuenta de AWS tiene un formato similar a 111122223333.

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

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

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

Valor predeterminado: `preferred`

Valores válidos:

- **`preferred`**: el punto de conexión debe incluir el ID de cuenta si está disponible.
- **`disabled`**: un punto de conexión resuelto no incluye el ID de cuenta.
- **`required`**: el punto de conexión debe incluir el ID de cuenta. Si el ID de la cuenta no está disponible, el SDK lanza un error.

Support by AWS SDKs and tools

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

SDK	com| e	Publicado en la versión de SDK	Notas o más información
AWS CLI v2	Sí	2.25.0	
AWS CLI v1	Sí	1.38.0	
SDK para C++	No		
SDK para Go V2 (1.x)	Sí	v1.35.0	
SDK para Go 1.x (V1)	No		
SDK para Java 2.x	Sí	v2.28.4	

SDK	com| e	Publicado en la versión de SDK	Notas o más información
SDK para Java 1.x	Sí	v1.12.771	
SDK para 3.x JavaScript	Sí	v3.656.0	
SDK para 2.x JavaScript	No		
SDK para Kotlin	Sí	v1.3.37	
SDK para .NET 4.x	Sí	4.0.0	
SDK para .NET 3.x	No		
SDK para PHP 3.x	Sí	v3.318.0	
SDK para Python (Boto3)	Sí	1.37.0	
SDK para Ruby 3.x	Sí	v1.123.0	
SDK para Rust	Sí	versión-2025-04-24	
SDK para Swift	Sí	1.2.0	
Herramientas para V5 PowerShell	No		
Herramientas para la PowerShell V4	No		

Application ID

> **Note**
>
> Para obtener ayuda para comprender el diseño de las páginas de configuración o para interpretar la tabla Support by AWS SDKs and tools que aparece a continuación, consulteCómo comprender las páginas de configuración de esta guía.

Varias aplicaciones de clientes Cuenta de AWS pueden utilizar una sola aplicación para realizar llamadas 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 SDKs y 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 Panel de AWS Health para identificar de forma exclusiva qué aplicaciones están asociadas a la notificación.

Configure esta funcionalidad mediante lo siguiente:

`sdk_ua_app_id`- configuración de AWS **`config`** archivos compartidos, **`AWS_SDK_UA_APP_ID`**: variable de entorno, **`sdk.ua.appId`**- Propiedad del sistema JVM: solo Java/Kotlin

Esta configuración es una cadena única que se asigna a la aplicación para identificar a cuáles de las aplicaciones de una determinada aplicación Cuenta de AWS realizan llamadas. 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 el intérprete de comandos que se utiliza, realice el escape del valor según corresponda.

Support by AWS SDKs and tools

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

SDK	cc e	Notas o más información
AWS CLI v2	Sí	
SDK para C++	Sí	No se admite el archivo compartido `config`.
SDK para Go V2 (1.x)	Sí	
SDK para Go 1.x (V1)	No	
SDK para Java 2.x	Parci	No se admite la configuración de archivos `config` compartidos ni de variable de entorno.
SDK para Java 1.x	No	
SDK para 3.x JavaScript	Sí	
SDK para 2.x JavaScript	No	
SDK para Kotlin	Sí	La propiedad del sistema JVM es `aws.userAgentAppId` .
SDK para .NET 4.x	Sí	
SDK para .NET 3.x	Sí	
SDK para PHP 3.x	Sí	

SDK	cc e	Notas o más información
SDK para Python (Boto3)	Sí	
SDK para Ruby 3.x	Sí	
SDK para Rust	Sí	
SDK para Swift	Sí	
Herramientas para V5 PowerShell	Sí	
Herramientas para la PowerShell V4	Sí	

Metadatos de la instancia de Amazon EC2

> **Note**
>
> Para obtener ayuda para comprender el diseño de las páginas de configuración o para interpretar la tabla Support by AWS SDKs and tools que aparece a continuación, consulteCómo comprender las páginas de configuración de esta guía.

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

Configure esta funcionalidad mediante lo siguiente:

`metadata_service_num_attempts`- configuración de AWS **`config`** archivos compartidos, **`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`- configuración de AWS **`config`** archivos compartidos, **`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
```

Support by AWS SDKs and tools

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

SDK	cc e	Notas o más información
AWS CLI v2	Sí	

SDK	cc e	Notas o más información
SDK para C++	No	
SDK para Go V2 (1.x)	No	
SDK para Go 1.x (V1)	No	
SDK para Java 2.x	Parci	Solo se admite `AWS_METADATA_SERVICE_TIMEOUT` .
SDK para Java 1.x	Parci	Solo se admite `AWS_METADATA_SERVICE_TIMEOUT` .
SDK para 3.x JavaScript	No	
SDK para 2.x JavaScript	No	
SDK para Kotlin	No	
SDK para .NET 4.x	No	
SDK para .NET 3.x	No	
SDK para PHP 3.x	Sí	
SDK para Python (Boto3)	Sí	
SDK para Ruby 3.x	No	
SDK para Rust	No	
SDK para Swift	No	
Herramientas para V5 PowerShell	No	
Herramientas para V4 PowerShell	No	

Puntos de acceso de Amazon S3

> **Note**
>
> Para obtener ayuda para comprender el diseño de las páginas de configuración o para interpretar la tabla Support by AWS SDKs and tools que aparece a continuación, consulteCómo comprender las páginas de configuración de esta guía.

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, puedes usar el punto de acceso Amazon Resource Names (ARNs) en el campo del bucket para las operaciones de la API en lugar de especificar el nombre del bucket de forma explícita. Se utilizan para operaciones específicas, como el uso de un ARN de punto de acceso `GetObject` para recuperar un objeto de un bucket o el uso del ARN de un punto de acceso `PutObject` para añadir un objeto a un bucket.

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

Configure esta funcionalidad mediante lo siguiente:

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

Esta configuración controla si el SDK usa el ARN del punto de acceso Región de AWS para construir el punto final regional de la solicitud. El SDK valida que el Región de AWS ARN esté servido por la AWS misma partición que la Región de AWS configurada por el cliente 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`**— El SDK usa los ARN Región de AWS al construir el punto final en lugar de los configurados por el cliente. Región de AWS Excepción: si la configuración del cliente Región de

AWS es un FIPS Región de AWS, debe coincidir con los ARN. Región de AWS De lo contrario, se producirá un error.

- **`false`**: el SDK utiliza los datos configurados por el cliente de Región de AWS al construir el punto de conexión.

Support by AWS SDKs and tools

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

SDK	cc e	Notas o más información
AWS CLI v2	Sí	
SDK para C++	Sí	
SDK para Go V2 (1.x)	Sí	
SDK para 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 para Java 2.x	Sí	
SDK para Java 1.x	Sí	No se admite la propiedad del sistema JVM.
SDK para 3.x JavaScript	Sí	
SDK para 2.x JavaScript	Sí	
SDK para Kotlin	Sí	
SDK para .NET 4.x	Sí	
SDK para .NET 3.x	Sí	No sigue la prioridad estándar; el valor del archivo compartido `config` tiene prioridad sobre la variable de entorno.

SDK	cc e	Notas o más información
SDK para PHP 3.x	Sí	
SDK para Python (Boto3)	Sí	
SDK para Ruby 3.x	Sí	
SDK para Rust	No	
SDK para Swift	No	
Herramientas para V5 PowerShell	Sí	No sigue la prioridad estándar; el valor del archivo compartido `config` tiene prioridad sobre la variable de entorno.
Herramientas para V4 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

> **Note**
>
> Para obtener ayuda para comprender el diseño de las páginas de configuración o para interpretar la tabla Support by AWS SDKs and tools que aparece a continuación, consulteCómo comprender las páginas de configuración de esta guía.

Los puntos de acceso multirregión de Amazon S3 proporcionan un punto de conexión global que las aplicaciones pueden utilizar para satisfacer las solicitudes de los buckets de S3 ubicados en varias Regiones de AWS. Puede utilizar puntos de acceso multirregión para crear aplicaciones de multirregiones 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. El SDK obtiene este algoritmo mediante una dependencia del AWS Bibliotecas de Common Runtime (CRT).

Configure esta funcionalidad mediante lo siguiente:

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

Esta configuración controla si el SDK puede intentar 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.

Support by AWS SDKs and tools

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

SDK	cc e	Notas o más información
AWS CLI v2	Sí	

SDK	cc e	Notas o más información
SDK para C++	Sí	
SDK para Go V2 (1.x)	Sí	
SDK para Go 1.x (V1)	No	
SDK para Java 2.x	Sí	
SDK para Java 1.x	No	
SDK para 3.x JavaScript	Sí	
SDK para 2.x JavaScript	No	
SDK para Kotlin	Sí	
SDK para .NET 4.x	Sí	
SDK para .NET 3.x	Sí	
SDK para PHP 3.x	Sí	
SDK para Python (Boto3)	Sí	
SDK para Ruby 3.x	Sí	
SDK para Rust	Sí	
SDK para Swift	No	
Herramientas para V5 PowerShell	Sí	
Herramientas para la PowerShell V4	Sí	

Autenticación de sesión de S3 Express One Zone

> **Note**
>
> Para obtener ayuda para comprender el diseño de las páginas de configuración o para interpretar la tabla Support by AWS SDKs and tools que aparece a continuación, consulteCómo comprender las páginas de configuración de esta guía.

S3 Express One Zone es la clase de almacenamiento de alto rendimiento de Amazon S3 que proporciona una latencia de milisegundos de un solo dígito para datos de acceso frecuente. Cuando utiliza los buckets AWS SDKs y las herramientas de S3 Express One Zone, utilizan automáticamente la autenticación basada en sesiones, que está optimizada para la autorización de solicitudes de datos con baja latencia. Utiliza tokens de sesión únicamente con operaciones zonales (de nivel de objeto) para distribuir la latencia asociada a la autorización entre varias solicitudes de una sesión, lo que reduce la sobrecarga de la autenticación y mejora el rendimiento general de la solicitud.

Los buckets S3 Express One Zone utilizan un formato de denominación específico que incluye el ID de la zona de disponibilidad, por ejemplo `bucket-name--usw2-az1--x-s3`. Cuando el SDK detecta este patrón de nomenclatura, enruta automáticamente las solicitudes a los puntos de conexión de S3 Express One Zone correspondientes y aplica el flujo de autenticación optimizado. La autenticación de sesión crea credenciales temporales específicas de bucket que otorgan acceso de baja latencia a su bucket y el SDK las almacena en caché y actualiza automáticamente. Consulte S3 Express One Zone en la Guía del usuario de Amazon S3 para obtener más información.

De forma predeterminada, la autenticación de sesión está habilitada para buckets de S3 Express One Zone.

Configure esta funcionalidad mediante lo siguiente:

`s3_disable_express_session_auth`- configuración de archivos compartidos AWS **`config`**, **`AWS_S3_DISABLE_EXPRESS_SESSION_AUTH`**: variable de entorno, **`aws.disableS3ExpressAuth`**- Propiedad del sistema JVM: solo Java/Kotlin

Controla si la autenticación de sesión de S3 Express One Zone está deshabilitada. Cuando se configura en `true`, el SDK utiliza la autenticación SigV4 estándar para los buckets de S3 Express One Zone en lugar de la autenticación de sesión.

Valor predeterminado: `false`

Valores válidos:

- **`true`**: deshabilita la autenticación de sesión de S3 Express One Zone.
- **`false`**: habilita la autenticación de sesión de S3 Express One Zone.

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

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

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

```
export AWS_S3_DISABLE_EXPRESS_SESSION_AUTH=true
```

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

```
setx AWS_S3_DISABLE_EXPRESS_SESSION_AUTH true
```

Support by AWS SDKs and tools

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

SDK	compe	Notas o más información
AWS CLI v2	Sí	
AWS CLI v1	Sí	
SDK para C++	Sí	
SDK para Go V2 (1.x)	Sí	
SDK para Go 1.x (V1)	No	Para usar la configuración de archivos compartidos `config`, debe activar la carga desde el archivo de configuración; consulte Sesiones.

SDK	compe	Notas o más información
SDK para Java 2.x	Sí	
SDK para Java 1.x	No	
SDK para JavaScript 3.x	Sí	
SDK para 2.x JavaScript	No	
SDK para Kotlin	Sí	La propiedad del sistema JVM es `aws.s3DisableExpressSessionAuth` .
SDK para .NET 4.x	Sí	
SDK para .NET 3.x	Sí	
SDK para PHP 3.x	Sí	
SDK para Python (Boto3)	Sí	
SDK para Ruby 3.x	Sí	
SDK para Rust	Sí	
SDK para Swift	Sí	
Herramientas para V5 PowerShell	Sí	
Herramientas para la PowerShell V4	Sí	

Esquema de autenticación

> **Note**
>
> Para obtener ayuda para comprender el diseño de las páginas de configuración o para interpretar la tabla Support by AWS SDKs and tools que aparece a continuación, consulteCómo comprender las páginas de configuración de esta guía.

AWS los servicios admiten varios esquemas de autenticación, como la versión de AWS firma 4 (SiGv4) y la versión de AWS firma 4a (SiGv4a). De forma predeterminada, SDKs seleccione los esquemas de autenticación en función de las definiciones del modelo de servicio y priorice los esquemas que ofrezcan la mejor compatibilidad. Sin embargo, se puede configurar el esquema de autenticación preferido para optimizarlo en función de requisitos específicos.

A diferencia de SigV4, las solicitudes firmadas con SigV4a son válidas en varias Regiones de AWS. SigV4a proporciona una mayor disponibilidad mediante la firma de solicitudes entre regiones, lo que permite la conmutación por error automática a las regiones de respaldo en caso de interrupciones regionales. Esto es particularmente beneficioso para servicios globales como AWS Identity and Access Management Amazon CloudFront.

Para obtener más información sobre estos dos esquemas de autenticación, consulte AWS Signature Version 4 para solicitudes de API en la Guía del usuario de IAM.

Configure esta funcionalidad mediante lo siguiente:

`auth_scheme_preference`- configuración de AWS **`config`** archivos compartidos, **`AWS_AUTH_SCHEME_PREFERENCE`**: variable de entorno, **`aws.authSchemePreference`**- Propiedad del sistema JVM: solo Java/Kotlin

Especifica una lista separada por comas de esquemas de autenticación preferidos en orden de prioridad. Cuando un servicio admite varios esquemas de autenticación, el SDK intenta usar los esquemas de esta lista en el orden especificado y vuelve al comportamiento predeterminado si ninguno de los esquemas preferidos está disponible.

Valor predeterminado: ninguno.

Valores válidos: una lista separada por comas de uno o varios de los siguientes valores:

- **`sigv4`**: Signature Version 4 (rendimiento más rápido, región única)

- **`sigv4a`**: Signature Version 4a (disponibilidad mejorada, compatibilidad entre regiones, tiene un rendimiento de firma más lento que SigV4)
- **`httpBearerAuth`**: autenticación mediante token HTTP Bearer

Se ignoran los espacios y los caracteres de tabulación entre los nombres de los esquemas.

Ejemplo de configuración de este valor en el archivo `config` para dar preferencia a SigV4a:

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

`sigv4a_signing_region_set`- configuración de AWS **`config`** archivos compartidos, **`AWS_SIGV4A_SIGNING_REGION_SET`**: variable de entorno

Especifica una lista separada por comas Regiones de AWS para la firma multirregional de SigV4a. Se utiliza como el conjunto de regiones predeterminado para la solicitud si el esquema de autenticación seleccionado es SigV4a.

Valor predeterminado: determinado por la solicitud.

Valores válidos: listas de Regiones de AWS separados por comas. Se ignoran los espacios y los caracteres de tabulación entre las regiones.

Support by AWS SDKs and tools

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

SDK	cc e	Notas o más información
AWS CLI v2	Sí	
SDK para C++	Sí	
SDK para Go V2 (1.x)	Sí	
SDK para Go 1.x (V1)	No	

SDK	cc e	Notas o más información
SDK para Java 2.x	Sí	
SDK para Java 1.x	No	
SDK para 3.x JavaScript	Sí	
SDK para 2.x JavaScript	No	
SDK para Kotlin	Sí	
SDK para .NET 4.x	Sí	
SDK para .NET 3.x	No	
SDK para PHP 3.x	Sí	
SDK para Python (Boto3)	Sí	
SDK para Ruby 3.x	Sí	
SDK para Rust	Sí	
SDK para Swift	Sí	
Herramientas para V5 PowerShell	Sí	
Herramientas para V4 PowerShell	No	

Región de AWS

> **Note**
>
> Para obtener ayuda para comprender el diseño de las páginas de configuración o para interpretar la tabla Support by AWS SDKs and tools que aparece a continuación, consulteCómo comprender las páginas de configuración de esta guía.

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

Con Regiones de AWS, puede acceder a Servicios de AWS esa residencia física 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 de Servicio de AWS las solicitudes están asociadas a una región geográfica en particular. Los recursos que crea en una región no existen en ninguna otra región salvo que utilice explícitamente una característica de replicación ofrecida por un Servicio de AWS. Por ejemplo, Amazon S3 y Amazon EC2 admiten la replicación entre regiones. Algunos servicios, como IAM, no tienen recursos regionales.

El informe Referencia general de AWS contiene la siguiente información:

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

Cómo crear clientes de servicio

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

Si no se especifica explícitamente ninguna región para el cliente en el código en sí, 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 de su SDK (consulte la guía del SDK específica o la base de código de su SDK).

Configure esta funcionalidad mediante lo siguiente:

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

Especifica el valor predeterminado Región de AWS que se utilizará en AWS las solicitudes. Esta región se usa para las solicitudes de servicio del SDK 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, como se muestran en Puntos de conexión de AWS en la referencia general de AWS . Por ejemplo, el valor `us-east-1` establece el punto de conexión en la región Región de AWS Este de EE. UU. (Norte de Virginia).
- `aws-global`especifica el punto de enlace global para los servicios que admiten un punto de enlace global independiente además de los puntos de enlace 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 la guía específica AWS para desarrolladores del SDK.

Support by AWS SDKs and tools

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

SDK	cc e	Notas o más información
AWS CLI v2	Sí	AWS CLI v2 utiliza cualquier valor de entrada `AWS_REGION` antes de cualquier valor de entrada `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.
SDK para C++	Sí	
SDK para Go V2 (1.x)	Sí	
SDK para 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 para Java 2.x	Sí	
SDK para Java 1.x	Sí	
SDK para JavaScript 3.x	Sí	
SDK para 2.x JavaScript	Sí	
SDK para Kotlin	Sí	

SDK	cc e	Notas o más información
SDK para .NET 4.x	Sí	
SDK para .NET 3.x	Sí	
SDK para PHP 3.x	Sí	
SDK para Python (Boto3)	Sí	Este SDK usa una variable de entorno llamada `AWS_DEFAULT_REGION` para este propósito.
SDK para Ruby 3.x	Sí	
SDK para Rust	Sí	
SDK para Swift	Sí	
Herramientas para V5 PowerShell	Sí	
Herramientas para la PowerShell V4	Sí	

AWS STS Puntos finales regionales

> **Note**
>
> Para obtener ayuda para comprender el diseño de las páginas de configuración o para interpretar la tabla Support by AWS SDKs and tools que aparece a continuación, consulteCómo comprender las páginas de configuración de esta guía.

AWS Security Token Service (AWS STS) está disponible como un servicio global y regional. Algunos 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}`). AWS SDKs En las regiones que están habilitadas de forma predeterminada, las solicitudes al punto final AWS STS global se atienden

automáticamente en la misma región en la que se origina la solicitud. En las regiones en las que se ha optado por participar, las solicitudes al punto de enlace AWS STS global las atiende un único destinatario Región de AWS, EE. UU. Este (Norte de Virginia). Para obtener más información sobre los AWS STS puntos finales, consulte los puntos finales en la referencia de la AWS Security Token Service API o Administrar AWS STSRegión de AWS en una de la guía del AWS Identity and Access Management usuario.

Se recomienda utilizar puntos de enlace regionales siempre que sea posible y configurar los suyos. AWS Región de AWS Los clientes de particiones que no sean comerciales deben usar puntos de conexión regionales. No todas las herramientas SDKs y herramientas 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.

> **Note**
>
> AWS ha realizado cambios en el punto final global AWS Security Token Service (AWS STS) (`https://sts.amazonaws.com`) de las regiones habilitadas de forma predeterminada para mejorar su capacidad de recuperación y rendimiento. AWS STS las solicitudes al punto final global se atienden automáticamente al Región de AWS igual que sus cargas de trabajo. Estos cambios no se implementarán en las regiones registradas. Le recomendamos que utilice los puntos de enlace AWS STS regionales adecuados. Para obtener más información, consulte Cambios en los puntos de conexión globales de AWS STS, en la Guía del usuario de AWS Identity and Access Management .

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

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

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

Valor predeterminado: `regional`, consulte las excepciones en la siguiente tabla.

> **Note**
>
> Todas las nuevas versiones principales del SDK que se publiquen después de julio de 2022 se instalarán de forma predeterminada en `regional`. Es posible que las nuevas versiones principales del SDK eliminen esta configuración y este comportamiento de uso de `regional`. 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 AWS STS punto final global,`sts.amazonaws.com`.
- **`regional`**— El SDK o la herramienta siempre utilizan el AWS STS punto final de la región configurada actualmente. Por ejemplo, si el cliente está configurado para usar`us-west-2`, todas las llamadas AWS STS se realizan al punto final regional`sts.us-west-2.amazonaws.com`, en lugar de al `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
```

Support by AWS SDKs and tools

> **Note**
>
> Se AWS recomienda utilizar puntos de conexión regionales siempre que sea posible y configurar los suyosRegión de AWS.

En la siguiente tabla se resume lo siguiente, para su SDK o herramienta:

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

En la tabla también se usan los siguientes valores:

- Punto de conexión global: `https://sts.amazonaws.com`.
- Punto de conexión regional: se basa en la Región de AWS configurada que utiliza la aplicación.
- **`us-east-1`** (Regional): utiliza el punto de conexión de la región `us-east-1`, pero con identificadores de sesión más largos que las solicitudes globales habituales.

SDK	(((:	Valor de configuración predeterminado	El cliente de servicio predeterminado es el punto de conexión de STS	Comportamiento alternativo del cliente de servicio	Notas o más información
AWS CLI v2	N	N/A	Punto de conexión regional	Punto de conexión global	
AWS CLI v1	S	`legacy`	Punto de conexión global	Punto de conexión global	
SDK para C++	N	N/A	Punto de conexión regional	`us-east-1` (Regional)	
SDK para Go V2 (1.x)	N	N/A	Punto de conexión regional	Error de solicitud	
SDK para Go 1.x (V1)	S	`legacy`	Punto de conexión global	Punto de conexión global	Para usar la configuración de archivos compartidos `config`, debe activar la carga desde el archivo de configuración; consulte Sesiones.
SDK para Java 2.x	N	N/A	Punto de conexión regional	Error de solicitud	Si no hay ninguna región configurada, `AssumeRole` y `AssumeRoleWithWebIdentity` utilizarán el punto de conexión de STS global.

SDK	(((: Valor de configuración predeterminado	El cliente de servicio predeterminado es el punto de conexión de STS	Comportamiento alternativo del cliente de servicio	Notas o más información
SDK para Java 1.x	S `legacy`	Punto de conexión global	Punto de conexión global	
SDK para JavaScript 3.x	N N/A	Punto de conexión regional	`us-east-1` (Regional)	
SDK para 2.x JavaScript	S `legacy`	Punto de conexión global	Punto de conexión global	
SDK para Kotlin	N N/A	Punto de conexión regional	Punto de conexión global	
SDK para .NET 4.x	N N/A	Punto de conexión regional	`us-east-1` (Regional)	
SDK para .NET 3.x	S `regional`	Punto de conexión global	Punto de conexión global	
SDK para PHP 3.x	S `regional`	Punto de conexión global	Error de solicitud	

SDK	(((:	Valor de configuración predeterminado	El cliente de servicio predeterminado es el punto de conexión de STS	Comportamiento alternativo del cliente de servicio	Notas o más información
SDK para Python (Boto3)	S	`regional`	Punto de conexión global	Punto de conexión global	
SDK para Ruby 3.x	S	`regional`	Punto de conexión regional	Error de solicitud	
SDK para Rust	N	N/A	Punto de conexión regional	Error de solicitud	
SDK para Swift	N	N/A	Punto de conexión regional	Error de solicitud	
Herramientas para V5 PowerShell	S	`regional`	Punto de conexión global	Punto de conexión global	
Herramientas para V4 PowerShell	S	`regional`	Punto de conexión global	Punto de conexión global	

Protecciones de integridad de datos para Amazon S3

> **Note**
>
> Para obtener ayuda para comprender el diseño de las páginas de configuración o para interpretar la tabla Support by AWS SDKs and tools que aparece a continuación, consulteCómo comprender las páginas de configuración de esta guía.

Durante algún tiempo, AWS SDKs han admitido las comprobaciones de integridad de los datos al cargar o descargar datos de Amazon Simple Storage Service. Anteriormente, estas comprobaciones eran opcionales. Ahora, hemos activado estas comprobaciones de forma predeterminada, mediante algoritmos basados en CRC, como CRC32 NVME. CRC64 Aunque cada SDK o herramienta tiene un algoritmo predeterminado, se puede elegir un algoritmo diferente. Si se desea, también se puede seguir proporcionando manualmente una suma de comprobación precalculada para las cargas. El comportamiento uniforme en las cargas, las cargas multiparte, las descargas y los modos de cifrado simplifica las comprobaciones de integridad del lado del cliente.

Las versiones más recientes de nuestro AWS SDKs calculan AWS CLI automáticamente una suma de verificación basada en la comprobación de redundancia cíclica (CRC) para cada carga y la envía a Amazon S3. Amazon S3 calcula de forma independiente un valor de suma de comprobación en el servidor y lo valida con el valor proporcionado antes de almacenar el objeto y la suma de comprobación de forma duradera en los metadatos del objeto. Al almacenar la suma de comprobación en los metadatos junto al objeto, cuando este se descarga, la misma suma de comprobación puede devolverse automáticamente y usarse también para validar las descargas. También se puede verificar la suma de comprobación almacenada en los metadatos del objeto en cualquier momento.

Para obtener más información sobre las operaciones de suma de comprobación, las cargas multiparte o la lista de algoritmos de suma de comprobación compatibles, consulte Comprobación de la integridad de los objetos en Amazon S3 en la Guía del usuario de Amazon Simple Storage Service.

Cargas multiparte:

Amazon S3 también proporciona a los desarrolladores sumas de comprobación coherentes de todos los objetos en las cargas de una o varias partes.

Al cargar archivos en varias partes, SDKs calculan las sumas de verificación para cada parte. Amazon S3 utiliza estas sumas de comprobación para verificar la integridad de cada parte a través de la API de `UploadPart`. Además, Amazon S3 valida el tamaño y la suma de comprobación del archivo completo cuando se llama a la API de `CompleteMultipartUpload`.

Si su SDK tiene un Amazon S3 Transfer Manager para facilitar las cargas de varias partes, las sumas de comprobación se validan para las partes mediante el algoritmo predeterminado específico del SDK que se encuentra en la tabla Support by AWS SDKs and tools. Puede optar por utilizar una suma de verificación completa de objetos configurando la configuración `checksum_type` `FULL_OBJECT` o eligiendo utilizar el algoritmo NVME. CRC64

Si se utiliza una versión anterior del SDK o de la AWS CLI:

Si su aplicación utiliza una versión anterior a diciembre de 2024 del SDK o la herramienta, Amazon S3 seguirá calculando una suma de comprobación de CRC64 NVME en los objetos nuevos y la almacenará en los metadatos del objeto para consultarla en el futuro. Más adelante, podrá comparar la CRC almacenada con una CRC calculada por su parte y comprobar que la transmisión de red fue correcta. Además, puede ampliar manualmente la protección de integridad al proporcionar sus propias sumas de verificación precalculadas junto con sus solicitudes `PutObject` o `UploadPart`, que es la técnica estándar para solucionar este problema en las versiones anteriores.

Configure esta funcionalidad mediante lo siguiente:

`request_checksum_calculation`- configuración de archivos compartidos AWS **`config`**, **`AWS_REQUEST_CHECKSUM_CALCULATION`**: variable de entorno, **`aws.requestChecksumCalculation`**- Propiedad del sistema JVM: solo Java/Kotlin

De forma predeterminada, los usuarios tienen la opción de calcular una suma de comprobación de las solicitudes al enviar una solicitud. El usuario puede elegir cualquiera de los algoritmos de suma de comprobación disponibles como parte de la creación de la solicitud. De lo contrario, se utiliza un algoritmo predeterminado específico del SDK. Consulte la tabla Support by AWS SDKs and tools para ver el algoritmo predeterminado de cada SDK o herramienta.

Valor predeterminado: `WHEN_SUPPORTED`

Valores válidos:

- **`WHEN_SUPPORTED`**: la validación de la suma de comprobación se realiza en todas las cargas útiles solicitadas cuando la operación de la API lo admite, como las transferencias de datos a Amazon S3.

- **`WHEN_REQUIRED`**: la validación de la suma de comprobación solo se realiza cuando la operación de la API lo requiere.

`response_checksum_validation`- configuración de AWS **`config`** archivos compartidos, **`AWS_RESPONSE_CHECKSUM_VALIDATION`**: variable de entorno, **`aws.responseChecksumValidation`**- Propiedad del sistema JVM: solo Java/Kotlin

De forma predeterminada, los usuarios pueden optar por validar la suma de comprobación de la respuesta al enviar una solicitud. Se calcula una suma de comprobación para la carga útil de la respuesta y se compara con el encabezado de la respuesta de la suma de comprobación. Si se produce un error en la validación de la suma de comprobación, se genera un error para el usuario cuando se lee la carga útil.

El encabezado de respuesta de la suma de comprobación también indica el algoritmo de la suma de comprobación. El cliente de Amazon S3 intenta validar las sumas de comprobación de las respuestas para todas las operaciones de la API de Amazon S3 que admiten las sumas de comprobación. Sin embargo, si el SDK no ha implementado el algoritmo de suma de comprobación especificado, se omite esta validación.

Valor predeterminado: `WHEN_SUPPORTED`

Valores válidos:

- **`WHEN_SUPPORTED`**: la validación de la suma de comprobación se realiza en todas las cargas útiles de respuesta cuando la operación de la API lo admite, como las transferencias de datos a Amazon S3.
- **`WHEN_REQUIRED`**: la validación de la suma de comprobación solo se realiza cuando la operación de la API lo admite y la persona que llama ha habilitado explícitamente la suma de comprobación para la operación. Por ejemplo, cuando se llama a la API `GetObject` de Amazon S3 y el parámetro `ChecksumMode` se establece en habilitado.

Support by AWS SDKs and tools

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

> **Note**
>
> En la siguiente tabla, “CRT” hace referencia a AWS Bibliotecas de Common Runtime (CRT) y podría ser necesario agregar una dependencia adicional a su proyecto.

SDK	compa e	Algoritmo de suma de comprobación predeterminado	Algoritmos de suma de comprobación admitidos	Notas o más información
AWS CLI v2	Sí	CRC64NVME	CRC64NVME CRC32, CRC32 C, SHA1 SHA256	Para la AWS CLI versión 1, el algoritmo predeterminado y los algoritmos compatibles serán idénticos a los de Python (Boto3).
SDK para C++	Sí	CRC64NVME	CRC64NVME CRC32, CRC32 C, SHA1 SHA256	
SDK para Go V2 (1.x)	Sí	CRC32	CRC64NVME CRC32, CRC32 C, SHA1 SHA256	
SDK para Go 1.x (V1)	No			
SDK para Java 2.x	Sí	CRC32	CRC64NVME (solo mediante CRT), C CRC32, CRC32 SHA1 SHA256	
SDK para Java 1.x	No			

SDK	compa e	Algoritmo de suma de comprobación predeterminado	Algoritmos de suma de comprobación admitidos	Notas o más información
SDK para 3.x JavaScript	Sí	CRC32	CRC32, CRC32 C, SHA1 SHA256	
SDK para JavaScript 2.x	No			
SDK para Kotlin	Sí	CRC32	CRC32, CRC32 C, SHA1 SHA256	
SDK para.NET 4.x	Sí	CRC32	CRC32, CRC32 C, SHA1 SHA256	
SDK para .NET 3.x	Sí	CRC32	CRC32, CRC32 C SHA1, SHA256	
SDK para PHP 3.x	Sí	CRC32	CRC32, CRC32 C (solo a través de CRT), SHA1 SHA256	`awscrt`se requiere la extensión para poder utilizar C. CRC32
SDK para Python (Boto3)	Sí	CRC32	CRC64NVME (solo mediante CRT) CRC32, CRC32 C (solo mediante CRT),, SHA1 SHA256	
SDK para Ruby 3.x	Sí	CRC32	CRC64NVME (solo mediante CRT), CRC32 C (solo mediante CRT) CRC32,, SHA1 SHA256	

SDK	compa e	Algoritmo de suma de comprobación predeterminado	Algoritmos de suma de comprobación admitidos	Notas o más información
SDK para Rust	Sí	CRC32	CRC64NVME, C, CRC32 CRC32 SHA1 SHA256	
SDK para Swift	Sí	CRC32	CRC64NVME CRC32, CRC32 C, SHA1 SHA256	Se requiere la dependenc ia de CRT para todos los algoritmos.
Herramien tas para V5 PowerShell	Sí	CRC32	CRC32, CRC32 C, SHA1 SHA256	
Herramien tas para PowerShell V4	Sí	CRC32	CRC32, CRC32 C SHA1, SHA256	

Puntos de conexión de doble pila y FIPS

> **Note**
>
> Para obtener ayuda para comprender el diseño de las páginas de configuración o para interpretar la tabla Support by AWS SDKs and tools que aparece a continuación, consulteCómo comprender las páginas de configuración de esta guía.

Configure esta funcionalidad mediante lo siguiente:

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

Activa o desactiva si el SDK enviará solicitudes a los puntos de conexión 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`**: el SDK o la herramienta intentarán utilizar puntos de conexión de doble pila para realizar solicitudes de red. Si no existe un punto de conexión de doble pila para el servicio o Región de AWS, la solicitud fallará.
- **`false`**: el SDK o la herramienta no utilizará los puntos de conexión de doble pila para realizar solicitudes de red.

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

Activa o desactiva si el SDK enviará solicitudes a los puntos de conexión compatibles con FIPS. Los estándares federales de procesamiento de la 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 aquellos que deseen hacer negocios con el gobierno federal deben cumplir con las pautas de la FIPS. A diferencia de AWS los terminales estándar, los terminales FIPS utilizan una biblioteca de software TLS validada según la norma FIPS 140. Si esta configuración está habilitada y no existe un punto final FIPS para su Región de AWS servicio, es posible que se produzca un error en la llamada. AWS Puntos de conexión específicos del servicioy la `--endpoint-url` opción de AWS Command Line Interface anular esta configuración.

Para obtener más información sobre otras formas de especificar los puntos de enlace de FIPS Región de AWS, consulte Puntos de enlace de FIPS 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 (IPv4 y IPv6) en la referencia de la API de Amazon EC2.

Valor predeterminado: `false`

Valores válidos:

- **`true`**: el SDK o herramienta enviará solicitudes a los puntos de conexión compatibles con FIPS.

- **`false`**: el SDK o herramienta no enviará solicitudes a los puntos de conexión compatibles con FIPS.

Support by AWS SDKs and tools

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

SDK	cc e	Notas o más información
AWS CLI v2	Sí	
SDK para C++	Sí	
SDK para Go V2 (1.x)	Sí	
SDK para 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 para Java 2.x	Sí	
SDK para Java 1.x	No	
SDK para 3.x JavaScript	Sí	
SDK para 2.x JavaScript	Sí	
SDK para Kotlin	Sí	
SDK para.NET 4.x	Sí	
SDK para .NET 3.x	Sí	
SDK para PHP 3.x	Sí	
SDK para Python (Boto3)	Sí	

SDK	cc e	Notas o más información
SDK para Ruby 3.x	Sí	
SDK para Rust	Sí	
SDK para Swift	Sí	
Herramientas para V5 PowerShell	Sí	
Herramientas para la PowerShell V4	Sí	

Detección de puntos de conexión

> **Note**
>
> Para obtener ayuda para comprender el diseño de las páginas de configuración o para interpretar la tabla Support by AWS SDKs and tools que aparece a continuación, consulteCómo comprender las páginas de configuración de esta guía.

SDKs utilice la detección de puntos finales para acceder a los puntos finales del servicio (URLs para acceder a varios recursos) y, al mismo tiempo, mantener la flexibilidad necesaria 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, el SDK realiza una solicitud `DescribeEndpoints` para recuperar los puntos de conexión disponibles y, a continuación, los utiliza para completar operaciones específicas, como `CreateDatabase` o `CreateTable`.

Configure esta funcionalidad mediante lo siguiente:

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

Activa o desactiva la detección de puntos de conexión para DynamoDB.

La detección de puntos de conexión es obligatoria en Timestream y opcional en Amazon DynamoDB. El valor predeterminado de esta configuración es `true` o `false`, depende de si el servicio requiere la detección de puntos de conexión. Las solicitudes Timestream se establecen de forma predeterminada en `true` y las solicitudes de Amazon DynamoDB se establecen de forma predeterminada en `false`.

Valores válidos:

- **`true`**: el SDK debería intentar detectar automáticamente un punto de conexión para los servicios en los que la detección de puntos de conexión sea opcional.
- **`false`**: el SDK no debería intentar detectar automáticamente un punto de conexión para los servicios en los que la detección de puntos de conexión sea opcional.

Support by AWS SDKs and tools

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

SDK	cc e	Notas o más información
AWS CLI v2	Sí	
SDK para C++	Sí	
SDK para Go V2 (1.x)	Sí	
SDK para 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	cc e	Notas o más información
SDK para Java 2.x	Sí	El SDK para Java 2.x utiliza el nombre `AWS_ENDPO INT_DISCOVERY_ENABLED` de la variable de entorno.
SDK para Java 1.x	Parci	No se admite la propiedad del sistema JVM.
SDK para 3.x JavaScript	Sí	
SDK para 2.x JavaScript	Sí	
SDK para Kotlin	Sí	
SDK para .NET 4.x	Sí	
SDK para .NET 3.x	Sí	
SDK para PHP 3.x	Sí	
SDK para Python (Boto3)	Sí	
SDK para Ruby 3.x	Sí	
SDK para Rust	Parci	Compatible solo con Timestream.
SDK para Swift	No	
Herramientas para V5 PowerShell	Sí	
Herramientas para la PowerShell V4	Sí	

Ajustes de configuración general

> **Note**
>
> Para obtener ayuda para comprender el diseño de las páginas de configuración o para interpretar la tabla Support by AWS SDKs and tools que aparece a continuación, consulteCómo comprender las páginas de configuración de esta guía.

SDKs admiten algunos ajustes generales que configuran los comportamientos generales del SDK.

Configure esta funcionalidad mediante lo siguiente:

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

Algunos AWS servicios mantienen varias versiones de la API para permitir la compatibilidad con versiones anteriores. De forma predeterminada, el SDK y las operaciones de la AWS CLI utilizan la última versión de API disponible. Si deseas solicitar una versión de API específica para utilizarla en tus solicitudes, incluye la configuración de las `api_versions` en tu perfil.

Valor predeterminado: ninguno. (El SDK utiliza la última versión API de forma predeterminada).

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

El ejemplo establece una versión de API específica para dos AWS servicios del `config` archivo. Estas versiones de API se utilizan únicamente para los comandos que se ejecutan bajo el perfil que contiene estos ajustes. Los comandos de cualquier otro servicio utilizan la versión más reciente de la API de ese servicio.

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

`ca_bundle`- configuración de AWS **`config`** archivos compartidos, **`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 SSL/TLS conexiones.

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 los caracteres de escape de las rutas, a continuación se muestra un ejemplo de cómo configurar este valor en el archivo `config` 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`- configuración de AWS **`config`** archivos compartidos

Especifica el formato de los resultados en la AWS CLI AWS SDKs y otras herramientas.

Valor predeterminado: `json`

Valores válidos:

- **`json`**: la salida se formatea como una cadena JSON.
- **`yaml`**: la salida se formatea como una cadena YAML.
- **`yaml-stream`**: la salida se transmite y se formatea como una cadena YAML. La transmisión permite gestionar tipos de datos de gran tamaño de forma más rápida.

- **`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`- configuración de AWS **`config`** archivos compartidos

Especifica si el cliente del SDK o herramienta intenta validar parámetros antes de enviarlos al punto de conexión de servicio de AWS .

Valor predeterminado: `true`

Valores válidos:

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

Support by AWS SDKs and tools

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

SDK	cc e	Notas o más información
AWS CLI v2	Parci	`api_versions` no admitidas.
SDK para C++	Sí	

SDK	cc e	Notas o más información
SDK para Go V2 (1.x)	Parci	Las `api_versions` y la `parameter_validation` no son compatibles.
SDK para 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.
SDK para Java 2.x	No	
SDK para Java 1.x	No	
SDK para 3.x JavaScript	Sí	
SDK para 2.x JavaScript	Sí	
SDK para Kotlin	No	
SDK para .NET 4.x	No	
SDK para .NET 3.x	No	
SDK para PHP 3.x	Sí	
SDK para Python (Boto3)	Sí	
SDK para Ruby 3.x	Sí	
SDK para Rust	No	
SDK para Swift	No	
Herramientas para V5 PowerShell	No	
Herramientas para la PowerShell V4	No	

Inyección de prefijos de host

> **Note**
>
> Para obtener ayuda para comprender el diseño de las páginas de configuración o para interpretar la tabla Support by AWS SDKs and tools que aparece a continuación, consulteCómo comprender las páginas de configuración de esta guía.

La inyección de prefijos de host es una función en la que AWS SDKs se añade automáticamente un prefijo al nombre de host de los puntos finales del servicio para determinadas operaciones de la API. Este prefijo puede ser una cadena estática o un valor dinámico que incluya datos de los parámetros de la solicitud.

Por ejemplo, cuando se utiliza Amazon Simple Storage Service para realizar acciones en objetos o buckets de Amazon S3, el SDK reemplaza el nombre y el Cuenta de AWS ID del bucket en el punto final de la API.

Si bien este comportamiento es obligatorio para los puntos de enlace de AWS servicio normales, puede causar problemas al usar puntos de enlace personalizados, como puntos de enlace de VPC o herramientas de prueba locales. En estos casos, es posible que tenga que deshabilitar la inyección de prefijos de host.

Configure esta funcionalidad mediante lo siguiente:

`disable_host_prefix_injection`- configuración de archivos compartidos AWS **`config`**, **`AWS_DISABLE_HOST_PREFIX_INJECTION`**: variable de entorno, **`aws.disableHostPrefixInjection`**- Propiedad del sistema JVM: solo Java/Kotlin

Esta configuración controla si el SDK o la herramienta modificarán el nombre de host del punto de conexión mediante la anteposición de un prefijo de host tal como se define en el objeto o la variable de cliente del SDK.

Valor predeterminado: `false`

Valores válidos:

- **`true`**: inhabilitar la inyección de prefijos de host. El SDK no modificará el nombre de host del punto de conexión.

- **`false`**: habilitar la inyección de prefijos de host. El SDK antepondrá el prefijo de host al nombre de host del punto de conexión.

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

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

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

```
export AWS_DISABLE_HOST_PREFIX_INJECTION=true
```

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

```
setx AWS_DISABLE_HOST_PREFIX_INJECTION true
```

Ejemplos de inyección de prefijos de host

La siguiente tabla de ejemplos muestra cómo SDKs modificar el punto final cuando la inyección de prefijos de host está habilitada o deshabilitada.

- Prefijo de host: la plantilla de la cadena de propiedades del prefijo de host establecida en el objeto o variable de cliente del SDK en el código.
- Entradas: entradas adicionales configuradas en el objeto o variable de cliente del SDK en el código.
- Punto de conexión del cliente: punto de conexión derivado del cliente.
- Valor de configuración: valor resuelto para la configuración anterior.
- Punto de conexión resultante: el punto de conexión resultante que el cliente del SDK utiliza para realizar la llamada a la API.

Prefijo de host	Entradas	Punto de conexión del cliente	Valor de configuración	Punto de conexión resultante
«datos».	{}	"https://service.us-west-2.	false	"https://data.service.us-we

Prefijo de host	Entradas	Punto de conexión del cliente	Valor de configuración	Punto de conexión resultante
		amazonaws.com"		st-2.amazonaws.com"
«{Cubo} - {AccountId}.»	Cubeta: «aman-s3-demo-bucket1"," 123456789012" AccountId	"https://service.us-west-2.amazonaws.com"	false	"https://amzn-s3-demo-bucket1-123456789012.service.us-west-2.amazonaws.com"
«datos».	{}	"https://override.us-west-2.amazonaws.com"(como punto final de anulación)	true	"https://override.us-west-2.amazonaws.com"

Support by AWS SDKs and tools

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

SDK	cc e	Notas o más información
AWS CLI v2	Sí	
SDK para C++	No	No se admite la configuración, pero se puede configurar en código en el cliente mediante: `enableHostPrefixInjection`.

SDK	cc e	Notas o más información
SDK para Go V2 (1.x)	No	Se puede deshabilitar mediante middleware.
SDK para Go 1.x (V1)	No	
SDK para Java 2.x	No	No se admite la configuración, pero se puede configurar en código en el cliente mediante: `SdkAdvancedClientOption.DISABLE_HOST_PREFIX_INJECTION`.
SDK para Java 1.x	No	No se admite la configuración, pero se puede configurar en código en el cliente mediante: `withDisableHostPrefixInjection`.
SDK para 3.x JavaScript	No	No se admite la configuración, pero se puede configurar en código en el cliente mediante: `disableHostPrefix`.
SDK para 2.x JavaScript	No	No se admite la configuración, pero se puede configurar en código en el cliente mediante: `hostPrefixEnabled`.
SDK para Kotlin	No	
SDK para .NET 4.x	No	No se admite la configuración, pero se puede configurar en código en el cliente mediante: `DisableHostPrefixInjection`.
SDK para .NET 3.x	No	No se admite la configuración, pero se puede configurar en código en el cliente mediante: `DisableHostPrefixInjection`.
SDK para PHP 3.x	No	No se admite la configuración, pero se puede configurar en código en el cliente mediante: `disable_host_prefix_injection`.
SDK para Python (Boto3)	Sí	Se puede configurar en código en el cliente mediante: `inject_host_prefix`.

SDK	cc e	Notas o más información
SDK para Ruby 3.x	No	No se admite la configuración, pero se puede configurar en código en el cliente mediante: `disable_host_prefix_injection`.
SDK para Rust	No	
SDK para Swift	No	
Herramientas para V5 PowerShell	No	No se admite la configuración, pero se puede incluir en cmdlets específicos mediante el parámetro `-ClientConfig @{DisableHostPrefixInjection = $true}`.
Herramientas para la PowerShell V4	No	No se admite la configuración, pero se puede incluir en cmdlets específicos mediante el parámetro `-ClientConfig @{DisableHostPrefixInjection = $true}`.

Cliente IMDS

> **Note**
>
> Para obtener ayuda para comprender el diseño de las páginas de configuración o para interpretar la tabla Support by AWS SDKs and tools que aparece a continuación, consulteCómo comprender las páginas de configuración de esta guía.

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

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 tipo de punto de conexión.

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: URI válido.

La mayoría SDKs de las opciones son compatibles con las siguientes opciones. Consulte la base de códigos específica del SDK para obtener más información.

`endpoint_mode`: miembro del objeto de configuración del cliente

El modo de punto de conexión de IMDS.

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 I/O flujo válido, como 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 según el SDK.

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

Support by AWS SDKs and tools

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

SDK	cc e	Notas o más información
AWS CLI v2	Sí	
SDK para C++	No	
SDK para Go V2 (1.x)	Sí	

SDK	cc e	Notas o más información
SDK para Go 1.x (V1)	Sí	
SDK para Java 2.x	Sí	
SDK para Java 1.x	Sí	
SDK para 3.x JavaScript	Sí	
SDK para 2.x JavaScript	Sí	
SDK para Kotlin	No	
SDK para.NET 4.x	Sí	
SDK para .NET 3.x	Sí	
SDK para PHP 3.x	Sí	
SDK para Python (Boto3)	Sí	
SDK para Ruby 3.x	Sí	
SDK para Rust	Sí	
SDK para Swift	Sí	
Herramientas para V5 PowerShell	Sí	
Herramientas para la PowerShell V4	Sí	

Comportamiento de los reintentos

> **Note**
>
> Para obtener ayuda para comprender el diseño de las páginas de configuración o para interpretar la tabla Support by AWS SDKs and tools que aparece a continuación, consulteCómo comprender las páginas de configuración de esta guía.

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`- configuración de AWS **`config`** archivos compartidos, **`AWS_RETRY_MODE`**: variable de entorno, **`aws.retryMode`**- Propiedad del sistema JVM: solo Java/Kotlin

Especifica cómo el SDK o la herramienta para desarrolladores intentan los reintentos.

Valor predeterminado: este valor es específico del SDK. Consulte la guía de SDK específica o la base de código del SDK para ver su `retry_mode` predeterminado.

Valores válidos:

- `standard`— (Recomendado) El conjunto recomendado de reglas de reintento en todas partes. AWS SDKs Este modo incluye un conjunto estándar de errores que se reintentan y ajusta automáticamente el número de reintentos para maximizar la disponibilidad y la estabilidad. Este modo es seguro para su uso en aplicaciones de varios inquilinos. 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 Cómo elegir entre los modos de reintento `standard` y `adaptive`. Este modo es experimental y podría cambiar su comportamiento en el futuro.
- `legacy`: (no recomendado) específico para su SDK (consulte su guía de SDK específica o la base de código de su SDK).

`max_attempts`- configuración de AWS **`config`** archivos compartidos, **`AWS_MAX_ATTEMPTS`**: variable de entorno, **`aws.maxAttempts`**- Propiedad del sistema JVM: solo 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 el `retry_mode` es `legacy`: usa un valor predeterminado específico de su SDK (consulte su guía específica del SDK o la base de código de su SDK para ver el valor predeterminado de `max_attempts`).
- 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.

Cómo elegir entre los modos de reintento **`standard`** y **`adaptive`**

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

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

Standard	Flexible
Aplicación y casos de uso: todos	Aplicación y casos de uso: 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.

Standard	Flexible
Admite la interrupción de circuitos para evitar que el SDK vuelva a intentarlo durante las interrupciones.	Admite la interrupción de circuitos para evitar que el SDK vuelva a intentarlo durante las interrupciones.
Utiliza un retroceso exponencial fluctuante en caso de fallo.	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 limitar o retrasar el intento de solicitud inicial.

Si opta por utilizar el modo `adaptive`, 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 de ellos. Servicio de AWS Servicios de AWS pueden tener dimensiones adicionales que utilizan para limitar las solicitudes. Usemos el servicio Amazon DynamoDB como ejemplo. DynamoDB Región de AWS usa plus la tabla a la que se accede para acelerar las solicitudes. Esto significa que una tabla a la que está accediendo su código podría estar más limitada que otras. Si el código utilizaba el mismo cliente para acceder a todas las tablas y las solicitudes a una de esas tablas están limitadas, el modo de reintento adaptativo reducirá la tasa de solicitudes de todas las tablas. El código debe diseñarse para tener un cliente por par. Region-and-table Si experimentas una latencia inesperada al usar el `adaptive` modo, consulta la guía de AWS documentación específica del servicio que estés utilizando.

Detalles de implementación del modo de reintento

AWS SDKs Utiliza grupos 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 SDK utiliza dos grupos de buckets de tokens: un bucket de token de reintentos y un bucket de token de tasa de solicitud.

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

cuando las solicitudes se realizan correctamente, se devuelven al bucket. Si el bucket está vacío cuando se realiza un reintento, el SDK no volverá a intentar la solicitud.

- El bucket de tokens de tasa de solicitudes solo se usa en el modo de reintento `adaptive` para determinar la velocidad a la que se envían las solicitudes. Los tokens se adquieren del bucket 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 bucket de tokens de tasa de solicitud. Si un token no está disponible, esperará a que uno esté disponible. Es posible que el SDK tenga opciones de configuración disponibles para rechazar la solicitud en lugar de esperar. Los tokens del bucket se rellenan a un ritmo que se determina de forma dinámica, en función del número de respuestas de limitación que reciba el cliente.

`SendHTTPRequest`:

Este paso envía la solicitud a. AWS La mayoría AWS SDKs usa una biblioteca HTTP que usa grupos de conexiones para reutilizar una conexión existente al realizar una solicitud HTTP. 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 agregan al bucket de tokens si la solicitud se realiza correctamente. Solo en el modo de reintento `adaptive`, la tasa de llenado del bucket de tokens 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 de estado HTTP.
- El código de error devuelto por el servicio.
- Errores de conexión, definidos como cualquier error recibido por el SDK en el que no se reciba una respuesta HTTP del servicio.

Los errores transitorios (códigos de estado HTTP 400, 408, 500, 502, 503 y 504) y los errores de limitación (códigos de estado HTTP 400, 403, 429, 502, 503 y 509) se pueden volver a intentar. El comportamiento de los reintentos del SDK 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 configuración `retry_mode`, a menos que la configuración `max_attempts` lo anule.

`HasRetryQuota`

Este paso adquiere un token del bucket de tokens de reintento. Si el bucket de tokens 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. SDKs Utilizan un retroceso exponencial binario truncado con fluctuación. 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. Consulte la guía o el código fuente específicos del SDK para confirmarlo.

Support by AWS SDKs and tools

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

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

SDK	cc e	Notas o más información
SDK para Kotlin	Sí	
SDK para .NET 4.x	Sí	
SDK para .NET 3.x	Sí	
SDK para PHP 3.x	Sí	
SDK para Python (Boto3)	Sí	
SDK para Ruby 3.x	Sí	
SDK para Rust	Sí	
SDK para Swift	Sí	
Herramientas para V5 PowerShell	Sí	
Herramientas para V4 PowerShell	Sí	

Compresión de solicitudes

> **Note**
>
> Para obtener ayuda para comprender el diseño de las páginas de configuración o para interpretar la tabla Support by AWS SDKs and tools que aparece a continuación, consulteCómo comprender las páginas de configuración de esta guía.

AWS SDKs y las herramientas pueden comprimir automáticamente las cargas útiles al enviar solicitudes al soporte Servicios de AWS que las recibe. 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, el SDK o la

herramienta selecciona un algoritmo de codificación compatible tanto con el servicio como con el SDK. 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 método PutMetricDataAPI.

Configure esta funcionalidad mediante lo siguiente:

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

Activa o desactiva la opción de que el SDK o la herramienta comprima 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`- configuración de AWS **`config`** archivos compartidos, **`AWS_REQUEST_MIN_COMPRESSION_SIZE_BYTES`**: variable de entorno, **`aws.requestMinCompressionSizeBytes`**- Propiedad del sistema JVM: solo Java/Kotlin

Establece el tamaño mínimo en bytes del cuerpo de la solicitud que el SDK o la herramienta 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.

Support by AWS SDKs and tools

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

SDK	cc e	Notas o más información
AWS CLI v2	Sí	
SDK para C++	Sí	
SDK para Go V2 (1.x)	Sí	
SDK para Go 1.x (V1)	No	
SDK para Java 2.x	Sí	
SDK para Java 1.x	No	
SDK para 3.x JavaScript	Sí	
SDK para 2.x JavaScript	No	
SDK para Kotlin	Sí	
SDK para .NET 4.x	Sí	
SDK para .NET 3.x	Sí	
SDK para PHP 3.x	Sí	
SDK para Python (Boto3)	Sí	
SDK para Ruby 3.x	Sí	
SDK para Rust	Sí	
SDK para Swift	No	

SDK	cc e	Notas o más información
Herramientas para V5 PowerShell	Sí	
Herramientas para V4 PowerShell	Sí	

Puntos de conexión específicos del servicio

> **Note**
>
> Para obtener ayuda para comprender el diseño de las páginas de configuración o para interpretar la tabla Support by AWS SDKs and tools que aparece a continuación, consulteCómo comprender las páginas de configuración de esta guía.

La configuración del punto de conexión específico del servicio ofrece la opción de utilizar un punto de conexión de su elección para las solicitudes de la API y de hacer que esa opción persista. Estas configuraciones proporcionan flexibilidad para admitir puntos de conexión locales, puntos de conexión de VPC y entornos de desarrollo de AWS locales de terceros. Se pueden usar diferentes puntos de conexión para los entornos de prueba y producción. Puede especificar una URL de punto de conexión para Servicios de AWS individuales.

Configure esta funcionalidad mediante lo siguiente:

`endpoint_url`- configuración de AWS **`config`** archivos compartidos, **`AWS_ENDPOINT_URL`**: variable de entorno, **`aws.endpointUrl`**- Propiedad del sistema JVM: solo 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 AWS `config` archivo compartido para establecer 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: una URL que incluye el esquema y el host del punto de conexión. La URL puede contener opcionalmente un componente de ruta que contenga uno o más segmentos de ruta.

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

`AWS_ENDPOINT_URL_<SERVICE>`, donde `<SERVICE>` está el 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: una URL que incluye el esquema y el host del punto de conexión. La URL puede contener opcionalmente un componente de ruta que contenga uno o más segmentos de ruta.

`ignore_configured_endpoint_urls`- configuración de AWS **`config`** archivos compartidos, **`AWS_IGNORE_CONFIGURED_ENDPOINT_URLS`**: variable de entorno, **`aws.ignoreConfiguredEndpointUrls`**- Propiedad del sistema JVM: solo 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, siempre tendrá efecto incluir el `--endpoint-url` parámetro de línea de comandos en un AWS CLI comando o pasar la URL de un punto final a un constructor de clientes.

Valor predeterminado: `false`

Valores válidos:

- **`true`**: el SDK o la herramienta no leen ninguna opción de configuración personalizada del archivo compartido `config` ni de las variables de entorno para configurar la URL de un punto de conexión.

- **`false`**: el SDK o la herramienta utilizan todos los puntos de conexión disponibles proporcionados por el usuario desde el archivo compartido `config` o desde las variables de entorno.

Configuración de puntos de conexión mediante variables de entorno

Para dirigir las solicitudes de todos los servicios a una URL de punto de conexión personalizada, establezca la variable de entorno global de `AWS_ENDPOINT_URL`.

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

Para enrutar las solicitudes de una URL de punto final específica Servicio de AWS a una URL de punto final personalizada, usa la variable de `AWS_ENDPOINT_URL_<SERVICE>` entorno. Amazon DynamoDB tiene un `serviceId` de `DynamoDB`. Para este servicio, la variable de entorno de la URL del punto de conexión 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`. El Servicio de AWS identificador se basa en el modelo de la API, sustituyendo todos los espacios `serviceId` por guiones bajos y mayúsculas 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 definición de `services` para configurar una URL de punto de conexión específica que se utilizará para Amazon S3 y un punto de conexión 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`. `Elastic Beanstalk` El Servicio de AWS identificador se basa en el modelo de la API, 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 el perfil tiene credenciales basadas en roles configuradas mediante un parámetro `source_profile` para la funcionalidad de asumir roles de IAM, el SDK solo usa 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 usa el perfil `B` y realiza una llamada en el código a Amazon EC2, el punto de conexión 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. En el caso de las llamadas a la API realizadas a un Servicio de AWS valor determinado, 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.
   - En el caso de AWS CLI, este es el valor que proporciona el parámetro de la línea de `--endpoint-url` comandos. En el caso de un SDK, las asignaciones explícitas pueden adoptar la forma de un parámetro que se establece al crear una instancia de un Servicio de AWS cliente o un 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. En último lugar, se usa cualquier URL de punto final predeterminada para Servicio de AWS el respectivo.

Support by AWS SDKs and tools

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

SDK	cc e	Notas o más información
AWS CLI v2	Sí	
SDK para C++	Sí	

SDK	cc e	Notas o más información
SDK para Go V2 (1.x)	Sí	
SDK para Go 1.x (V1)	No	
SDK para Java 2.x	Sí	
SDK para Java 1.x	No	
SDK para 3.x JavaScript	Sí	
SDK para 2.x JavaScript	No	
SDK para Kotlin	Sí	
SDK para .NET 4.x	Sí	
SDK para .NET 3.x	Sí	
SDK para PHP 3.x	Sí	
SDK para Python (Boto3)	Sí	
SDK para Ruby 3.x	Sí	
SDK para Rust	Sí	
SDK para Swift	Sí	
Herramientas para V5 PowerShell	Sí	
Herramientas para la PowerShell V4	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 el AV **co** ar co o	**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: _F	AWS_ENDPOINT_URL_ALEXA_FOR_BUSINESS
amp	ar	AWS_ENDPOINT_URL_AMP
Amplify	ar	AWS_ENDPOINT_URL_AMPLIFY
AmplifyBackend	ar cl	AWS_ENDPOINT_URL_AMPLIFYBACKEND
AmplifyUIBuilder	ar bu	AWS_ENDPOINT_URL_AMPLIFYUIBUILDER
API Gateway	ap ay	AWS_ENDPOINT_URL_API_GATEWAY

serviceId	Cl id ac de se pa el AV **co** ar co o	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
ApiGatewayManagement ntApi	ap yr nt	AWS_ENDPOINT_URL_APIGATEWAYMANAGEMENTAPI
ApiGatewayV2	ap yv	AWS_ENDPOINT_URL_APIGATEWAYV2
AppConfig	ap	AWS_ENDPOINT_URL_APPCONFIG
AppConfigData	ap da	AWS_ENDPOINT_URL_APPCONFIGDATA
AppFabric	ap	AWS_ENDPOINT_URL_APPFABRIC
Appflow	ap	AWS_ENDPOINT_URL_APPFLOW
AppIntegrations	ap at	AWS_ENDPOINT_URL_APPINTEGRATIONS
Application Auto Scaling	ap or ca	AWS_ENDPOINT_URL_APPLICATION_AUTO_SCALING

serviceId	Cl id ac de se pa el A\ **cc** ar cc o	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
Application Insights	ap or ts	AWS_ENDPOINT_URL_APPLICATION_INSIGHTS
ApplicationCostPro filer	ap or f:	AWS_ENDPOINT_URL_APPLICATIONCOSTPROFILER
App Mesh	ap	AWS_ENDPOINT_URL_APP_MESH
AppRunner	ap	AWS_ENDPOINT_URL_APPRUNNER
AppStream	ap	AWS_ENDPOINT_URL_APPSTREAM
AppSync	ap	AWS_ENDPOINT_URL_APPSYNC
ARC Zonal Shift	a: _s	AWS_ENDPOINT_URL_ARC_ZONAL_SHIFT
Artifact	a:	AWS_ENDPOINT_URL_ARTIFACT
Athena	at	AWS_ENDPOINT_URL_ATHENA

serviceId	Cl id ac de se pa el AV **co** ar co o	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
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
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

serviceId	Cl id ac de se pa el AV **co** ar co o	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
Bedrock Agent	be ge	AWS_ENDPOINT_URL_BEDROCK_AGENT
Bedrock Agent Runtime	be ge ir	AWS_ENDPOINT_URL_BEDROCK_AGENT_RUNTIME
Bedrock Runtime	be ur	AWS_ENDPOINT_URL_BEDROCK_RUNTIME
billingconductor	bi no	AWS_ENDPOINT_URL_BILLINGCONDUCTOR
Braket	bi	AWS_ENDPOINT_URL_BRAKET
Budgets	bu	AWS_ENDPOINT_URL_BUDGETS
Cost Explorer	co ox	AWS_ENDPOINT_URL_COST_EXPLORER
chatbot	ch	AWS_ENDPOINT_URL_CHATBOT
Chime	ch	AWS_ENDPOINT_URL_CHIME

serviceId	Cl id ac de se pa el AV **co** ar co o	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
Chime SDK Identity	ch _i	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
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

serviceId	Cl id ac de se pa el AV **co** ar co o	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
Cloud9	cl	AWS_ENDPOINT_URL_CLOUD9
CloudControl	cl rc	AWS_ENDPOINT_URL_CLOUDCONTROL
CloudDirectory	cl cl	AWS_ENDPOINT_URL_CLOUDDIRECTORY
CloudFormation	cl af	AWS_ENDPOINT_URL_CLOUDFORMATION
CloudFront	cl t	AWS_ENDPOINT_URL_CLOUDFRONT
CloudFront KeyValueStore	cl t_ es	AWS_ENDPOINT_URL_CLOUDFRONT_KEYVALUESTORE
CloudHSM	cl	AWS_ENDPOINT_URL_CLOUDHSM
CloudHSM V2	cl v2	AWS_ENDPOINT_URL_CLOUDHSM_V2
CloudSearch	cl ch	AWS_ENDPOINT_URL_CLOUDSEARCH

serviceId	Cl id ac de se pa el AV **co** ar co o	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
CloudSearch Domain	c ch	AWS_ENDPOINT_URL_CLOUDSEARCH_DOMAIN
CloudTrail	c l	AWS_ENDPOINT_URL_CLOUDTRAIL
CloudTrail Data	c l_	AWS_ENDPOINT_URL_CLOUDTRAIL_DATA
CloudWatch	c h	AWS_ENDPOINT_URL_CLOUDWATCH
codeartifact	co ac	AWS_ENDPOINT_URL_CODEARTIFACT
CodeBuild	co	AWS_ENDPOINT_URL_CODEBUILD
CodeCatalyst	co ys	AWS_ENDPOINT_URL_CODECATALYST
CodeCommit	co t	AWS_ENDPOINT_URL_CODECOMMIT

serviceId	Cl id ac de se pa el AV **co** ar co o	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
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
CodeGuruProfiler	co ro	AWS_ENDPOINT_URL_CODEGURUPROFILER
CodePipeline	co ir	AWS_ENDPOINT_URL_CODEPIPELINE
CodeStar	co	AWS_ENDPOINT_URL_CODESTAR
CodeStar connections	co co ns	AWS_ENDPOINT_URL_CODESTAR_CONNECTIONS
codestar notificat ions	co no ic	AWS_ENDPOINT_URL_CODESTAR_NOTIFICATIONS

serviceId	Cl id ac de se pa el AV **co** ar co o	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
Cognito Identity	co de	AWS_ENDPOINT_URL_COGNITO_IDENTITY
Cognito Identity Provider	co de ro	AWS_ENDPOINT_URL_COGNITO_IDENTITY_PROVIDER
Cognito Sync	co yn	AWS_ENDPOINT_URL_COGNITO_SYNC
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

serviceId	Cl id ac de se pa el AV **co** ar co o	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
Connect Contact Lens	co on ns	AWS_ENDPOINT_URL_CONNECT_CONTACT_LENS
ConnectCampaigns	co mp	AWS_ENDPOINT_URL_CONNECTCAMPAIGNS
ConnectCases	co se	AWS_ENDPOINT_URL_CONNECTCASES
ConnectParticipant	co rt	AWS_ENDPOINT_URL_CONNECTPARTICIPANT
ControlTower	co we	AWS_ENDPOINT_URL_CONTROLTOWER
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

serviceId	Cl id ac de se pa el AV **co** ar co o	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
Customer Profiles	cu p:	AWS_ENDPOINT_URL_CUSTOMER_PROFILES
DataBrew	da	AWS_ENDPOINT_URL_DATABREW
DataExchange	da ng	AWS_ENDPOINT_URL_DATAEXCHANGE
Data Pipeline	da l:	AWS_ENDPOINT_URL_DATA_PIPELINE
DataSync	da	AWS_ENDPOINT_URL_DATASYNC
DataZone	da	AWS_ENDPOINT_URL_DATAZONE
DAX	da	AWS_ENDPOINT_URL_DAX
Detective	de	AWS_ENDPOINT_URL_DETECTIVE
Device Farm	de rr	AWS_ENDPOINT_URL_DEVICE_FARM
DevOps Guru	de ru	AWS_ENDPOINT_URL_DEVOPS_GURU

serviceId	Cl id ac de se pa el AV **ce** ar co o	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
Direct Connect	d: nı	AWS_ENDPOINT_URL_DIRECT_CONNECT
Application Discovery Service	aı or e: ce	AWS_ENDPOINT_URL_APPLICATION_DISCOVERY_SERVICE
DLM	dI	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	d:	AWS_ENDPOINT_URL_DRS
Directory Service	d: _s	AWS_ENDPOINT_URL_DIRECTORY_SERVICE
DynamoDB	dy	AWS_ENDPOINT_URL_DYNAMODB

serviceId	Cl id ac de se pa el AV **co** ar co o	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
DynamoDB Streams	dy st	AWS_ENDPOINT_URL_DYNAMODB_STREAMS
EBS	eb	AWS_ENDPOINT_URL_EBS
EC2	ec	AWS_ENDPOINT_URL_EC2
EC2 Instance Connect	ec nc ct	AWS_ENDPOINT_URL_EC2_INSTANCE_CONNECT
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

serviceId	Cl id ac de se pa el AV **co** ar co o	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
ElastiCache	el he	AWS_ENDPOINT_URL_ELASTICACHE
Elastic Beanstalk	el ea	AWS_ENDPOINT_URL_ELASTIC_BEANSTALK
Elastic Transcoder	el ra	AWS_ENDPOINT_URL_ELASTIC_TRANSCODER
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

serviceId	Cl id ac de se pa el AV **co** ar co o	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
EntityResolution	er o:	AWS_ENDPOINT_URL_ENTITYRESOLUTION
Elasticsearch Service	e: a: ic	AWS_ENDPOINT_URL_ELASTICSEARCH_SERVICE
EventBridge	ev ge	AWS_ENDPOINT_URL_EVENTBRIDGE
Evidently	ev	AWS_ENDPOINT_URL_EVIDENTLY
finspace	fi	AWS_ENDPOINT_URL_FINSPACE
finspace data	fi da	AWS_ENDPOINT_URL_FINSPACE_DATA
Firehose	fi	AWS_ENDPOINT_URL_FIREHOSE
fis	fi	AWS_ENDPOINT_URL_FIS
FMS	fr	AWS_ENDPOINT_URL_FMS
forecast	fo	AWS_ENDPOINT_URL_FORECAST

serviceId	Cl id ac de se pa el AV **co** ar co o	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
forecastquery	fo ue	AWS_ENDPOINT_URL_FORECASTQUERY
FraudDetector	fr ct	AWS_ENDPOINT_URL_FRAUDDETECTOR
FreeTier	fr	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

serviceId	Cl id ac de se pa el AV **co** ar co o	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
GreengrassV2	g: sv	AWS_ENDPOINT_URL_GREENGRASSV2
GroundStation	g: 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

serviceId	Cl id ac de se pa el A\ **co** ar co o	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
ImportExport	ir o:	AWS_ENDPOINT_URL_IMPORTEXPORT
Inspector	ir	AWS_ENDPOINT_URL_INSPECTOR
Inspector Scan	ir _s	AWS_ENDPOINT_URL_INSPECTOR_SCAN
Inspector2	ir 2	AWS_ENDPOINT_URL_INSPECTOR2
InternetMonitor	ir or	AWS_ENDPOINT_URL_INTERNETMONITOR
IoT	io	AWS_ENDPOINT_URL_IOT
IoT Data Plane	io p:	AWS_ENDPOINT_URL_IOT_DATA_PLANE
IoT Jobs Data Plane	io da e	AWS_ENDPOINT_URL_IOT_JOBS_DATA_PLANE

serviceId	Cl id ac de se pa el AV **co** ar co o	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
IoT 1Click Devices Service	ic k_ _s	AWS_ENDPOINT_URL_IOT_1CLICK_DEVICES_ SERVICE
IoT 1Click Projects	ic k_ s	AWS_ENDPOINT_URL_IOT_1CLICK_PROJECTS
IoTAnalytics	ic ic	AWS_ENDPOINT_URL_IOTANALYTICS
IotDeviceAdvisor	ic ac	AWS_ENDPOINT_URL_IOTDEVICEADVISOR
IoT Events	ic s	AWS_ENDPOINT_URL_IOT_EVENTS
IoT Events Data	ic s_	AWS_ENDPOINT_URL_IOT_EVENTS_DATA
IoTFleetHub	ic ul	AWS_ENDPOINT_URL_IOTFLEETHUB
IoTFleetWise	ic is	AWS_ENDPOINT_URL_IOTFLEETWISE

serviceId	Cl id ac de se pa el AV **co** ar co o	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
IoTSecureTunneling	io tu	AWS_ENDPOINT_URL_IOTSECURETUNNELING
IoTSiteWise	io se	AWS_ENDPOINT_URL_IOTSITEWISE
IoTThingsGraph	io gr	AWS_ENDPOINT_URL_IOTTHINGSGRAPH
IoTTwinMaker	io ke	AWS_ENDPOINT_URL_IOTTWINMAKER
IoT Wireless	io 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

serviceId	Cl id ac de se pa el AV **co** ar co o	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
KafkaConnect	ka ec	AWS_ENDPOINT_URL_KAFKACONNECT
kendra	ke	AWS_ENDPOINT_URL_KENDRA
Kendra Ranking	ke nl	AWS_ENDPOINT_URL_KENDRA_RANKING
Keyspaces	ke	AWS_ENDPOINT_URL_KEYSPACES
Kinesis	ki	AWS_ENDPOINT_URL_KINESIS
Kinesis Video Archived Media	ki id iv a	AWS_ENDPOINT_URL_KINESIS_VIDEO_ARCHIVED_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

serviceId	Cl id ac de se pa el AV **co** ar co o	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
Kinesis Video WebRTC Storage	k: ic t e	AWS_ENDPOINT_URL_KINESIS_VIDEO_WEBRT C_STORAGE
Kinesis Analytics	k: na	AWS_ENDPOINT_URL_KINESIS_ANALYTICS
Kinesis Analytics V2	k: na v2	AWS_ENDPOINT_URL_KINESIS_ANALYTICS_V2
Kinesis Video	k: ic	AWS_ENDPOINT_URL_KINESIS_VIDEO
KMS	kr	AWS_ENDPOINT_URL_KMS
LakeFormation	la t:	AWS_ENDPOINT_URL_LAKEFORMATION
Lambda	la	AWS_ENDPOINT_URL_LAMBDA
Launch Wizard	la za	AWS_ENDPOINT_URL_LAUNCH_WIZARD

serviceId	Cl id ac de se pa el AV **co** ar co o	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
Lex Model Building Service	le _l _s	AWS_ENDPOINT_URL_LEX_MODEL_BUILDING_ SERVICE
Lex Runtime Service	le me e	AWS_ENDPOINT_URL_LEX_RUNTIME_SERVICE
Lex Models V2	le s_	AWS_ENDPOINT_URL_LEX_MODELS_V2
Lex Runtime V2	le me	AWS_ENDPOINT_URL_LEX_RUNTIME_V2
License Manager	li ar	AWS_ENDPOINT_URL_LICENSE_MANAGER
License Manager Linux Subscriptions	li ar nu r:	AWS_ENDPOINT_URL_LICENSE_MANAGER_LIN UX_SUBSCRIPTIONS

serviceId	Cl id ac de se pa el AV **co** ar co o	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
License Manager User Subscriptions	l: ar e: ip	AWS_ENDPOINT_URL_LICENSE_MANAGER_USER_SUBSCRIPTIONS
Lightsail	l:	AWS_ENDPOINT_URL_LIGHTSAIL
Location	lo	AWS_ENDPOINT_URL_LOCATION
CloudWatch Logs	c: h_	AWS_ENDPOINT_URL_CLOUDWATCH_LOGS
LookoutEquipment	lo u:	AWS_ENDPOINT_URL_LOOKOUTEQUIPMENT
LookoutMetrics	lo t:	AWS_ENDPOINT_URL_LOOKOUTMETRICS
LookoutVision	lo s:	AWS_ENDPOINT_URL_LOOKOUTVISION
m2	m:	AWS_ENDPOINT_URL_M2

serviceId	Cl id ac de se pa el A\ **ce** ar cc o	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
Machine Learning	ma ea	AWS_ENDPOINT_URL_MACHINE_LEARNING
Macie2	ma	AWS_ENDPOINT_URL_MACIE2
ManagedBlockchain	ma oc	AWS_ENDPOINT_URL_MANAGEDBLOCKCHAIN
ManagedBlockchain Query	ma oc qu	AWS_ENDPOINT_URL_MANAGEDBLOCKCHAIN_QUERY
Marketplace Agreement	ma ce er	AWS_ENDPOINT_URL_MARKETPLACE_AGREEMENT
Marketplace Catalog	ma ce g	AWS_ENDPOINT_URL_MARKETPLACE_CATALOG
Marketplace Deployment	ma ce me	AWS_ENDPOINT_URL_MARKETPLACE_DEPLOYMENT

serviceId	Cl id ac de se pa el AV **co** ar co o	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
Marketplace Entitleme nt Service	ma ce er v:	AWS_ENDPOINT_URL_MARKETPLACE_ENTITLE MENT_SERVICE
Marketplace Commerce Analytics	ma ce ce ic	AWS_ENDPOINT_URL_MARKETPLACE_COMMERC E_ANALYTICS
MediaConnect	me ec	AWS_ENDPOINT_URL_MEDIACONNECT
MediaConvert	me e:	AWS_ENDPOINT_URL_MEDIACONVERT
MediaLive	me	AWS_ENDPOINT_URL_MEDIALIVE
MediaPackage	me ag	AWS_ENDPOINT_URL_MEDIAPACKAGE
MediaPackage Vod	me ag	AWS_ENDPOINT_URL_MEDIAPACKAGE_VOD

serviceId	Cl id ac de se pa el AV **co** ar co o	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
MediaPackageV2	me ac	AWS_ENDPOINT_URL_MEDIAPACKAGEV2
MediaStore	me e	AWS_ENDPOINT_URL_MEDIASTORE
MediaStore Data	me e_	AWS_ENDPOINT_URL_MEDIASTORE_DATA
MediaTailor	me o:	AWS_ENDPOINT_URL_MEDIATAILOR
Medical Imaging	me ma	AWS_ENDPOINT_URL_MEDICAL_IMAGING
MemoryDB	me	AWS_ENDPOINT_URL_MEMORYDB
Marketplace Metering	ma ce ng	AWS_ENDPOINT_URL_MARKETPLACE_METERING
Migration Hub	m: _h	AWS_ENDPOINT_URL_MIGRATION_HUB
mgn	mg	AWS_ENDPOINT_URL_MGN

serviceId	Cl id ac de se pa el AV **co** ar co o	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
Migration Hub Refactor Spaces	m: _I ct es	AWS_ENDPOINT_URL_MIGRATION_HUB_REFAC TOR_SPACES
MigrationHub Config	m: hu g	AWS_ENDPOINT_URL_MIGRATIONHUB_CONFIG
MigrationHubOrches trator	m: hu t:	AWS_ENDPOINT_URL_MIGRATIONHUBORCHESTRATOR
MigrationHubStrategy	m: hu gy	AWS_ENDPOINT_URL_MIGRATIONHUBSTRATEGY
Mobile	mo	AWS_ENDPOINT_URL_MOBILE
mq	mo	AWS_ENDPOINT_URL_MQ
MTurk	mt	AWS_ENDPOINT_URL_MTURK
MWAA	mw	AWS_ENDPOINT_URL_MWAA

serviceId	Cl id ac de se pa el AV **co** ar co o	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
Neptune	ne	AWS_ENDPOINT_URL_NEPTUNE
Neptune Graph	ne ra	AWS_ENDPOINT_URL_NEPTUNE_GRAPH
neptunedata	ne ta	AWS_ENDPOINT_URL_NEPTUNEDATA
Network Firewall	ne i:	AWS_ENDPOINT_URL_NETWORK_FIREWALL
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

serviceId	Cl id ac de se pa el A\ **co** ar co o	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
OpenSearchServerless	op hs ss	AWS_ENDPOINT_URL_OPENSEARCHSERVERLESS
OpsWorks	op	AWS_ENDPOINT_URL_OPSWORKS
OpsWorksCM	op m	AWS_ENDPOINT_URL_OPSWORKSCM
Organizations	or ic	AWS_ENDPOINT_URL_ORGANIZATIONS
OSIS	os	AWS_ENDPOINT_URL_OSIS
Outposts	ou	AWS_ENDPOINT_URL_OUTPOSTS
p8data	p8	AWS_ENDPOINT_URL_P8DATA
p8data	p8	AWS_ENDPOINT_URL_P8DATA
Panorama	pa	AWS_ENDPOINT_URL_PANORAMA
Payment Cryptography	pa ry hy	AWS_ENDPOINT_URL_PAYMENT_CRYPTOGRAPHY

serviceId	Cl id ac de se pa el AV **co** ar co o	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
Payment Cryptography Data	pa ry hy	AWS_ENDPOINT_URL_PAYMENT_CRYPTOGRAPHY_DATA
Pca Connector Ad	pc ct	AWS_ENDPOINT_URL_PCA_CONNECTOR_AD
Personalize	pe ze	AWS_ENDPOINT_URL_PERSONALIZE
Personalize Events	pe ze	AWS_ENDPOINT_URL_PERSONALIZE_EVENTS
Personalize Runtime	pe ze e	AWS_ENDPOINT_URL_PERSONALIZE_RUNTIME
PI	pi	AWS_ENDPOINT_URL_PI
Pinpoint	pi	AWS_ENDPOINT_URL_PINPOINT
Pinpoint Email	pi er	AWS_ENDPOINT_URL_PINPOINT_EMAIL

serviceId	Cl id ac de se pa el A\ **co** ar cc o	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
Pinpoint SMS Voice	p: sr	AWS_ENDPOINT_URL_PINPOINT_SMS_VOICE
Pinpoint SMS Voice V2	p: sr _\	AWS_ENDPOINT_URL_PINPOINT_SMS_VOICE_V2
Pipes	p:	AWS_ENDPOINT_URL_PIPES
Polly	p(	AWS_ENDPOINT_URL_POLLY
Pricing	p:	AWS_ENDPOINT_URL_PRICING
PrivateNetworks	p: t\	AWS_ENDPOINT_URL_PRIVATENETWORKS
Proton	p:	AWS_ENDPOINT_URL_PROTON
QBusiness	ql	AWS_ENDPOINT_URL_QBUSINESS
QConnect	q(	AWS_ENDPOINT_URL_QCONNECT
QLDB	q]	AWS_ENDPOINT_URL_QLDB

serviceId	Cl id ac de se pa el A\ **co** ar co o	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
QLDB Session	q ic	AWS_ENDPOINT_URL_QLDB_SESSION
QuickSight	qu t	AWS_ENDPOINT_URL_QUICKSIGHT
RAM	ra	AWS_ENDPOINT_URL_RAM
rbin	rb	AWS_ENDPOINT_URL_RBIN
RDS	rd	AWS_ENDPOINT_URL_RDS
RDS Data	rd	AWS_ENDPOINT_URL_RDS_DATA
Redshift	re	AWS_ENDPOINT_URL_REDSHIFT
Redshift Data	re da	AWS_ENDPOINT_URL_REDSHIFT_DATA
Redshift Serverless	re se s	AWS_ENDPOINT_URL_REDSHIFT_SERVERLESS
Rekognition	re or	AWS_ENDPOINT_URL_REKOGNITION

serviceId	Cl id ac de se pa el A\ **co** ar co o	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
repostspace	re ce	AWS_ENDPOINT_URL_REPOSTSPACE
resiliencehub	re eh	AWS_ENDPOINT_URL_RESILIENCEHUB
Resource Explorer 2	re ex 2	AWS_ENDPOINT_URL_RESOURCE_EXPLORER_2
Resource Groups	re gr	AWS_ENDPOINT_URL_RESOURCE_GROUPS
Resource Groups Tagging API	re gr gg	AWS_ENDPOINT_URL_RESOURCE_GROUPS_TAG GING_API
RoboMaker	ro	AWS_ENDPOINT_URL_ROBOMAKER
RolesAnywhere	ro he	AWS_ENDPOINT_URL_ROLESANYWHERE
Route 53	ro	AWS_ENDPOINT_URL_ROUTE_53

serviceId	Cl id ac de se pa el A\ **co** ar co o	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
Route53 Recovery Cluster	rc ec lu	AWS_ENDPOINT_URL_ROUTE53_RECOVERY_CLUSTER
Route53 Recovery Control Config	rc ec or n1	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

serviceId	Cl id ac de se pa el A\ **co** ar co o	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
S3Outposts	s3 s	AWS_ENDPOINT_URL_S3OUTPOSTS
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 a1	AWS_ENDPOINT_URL_SAGEMAKER_GEOSPATIAL
SageMaker Metrics	sa _r	AWS_ENDPOINT_URL_SAGEMAKER_METRICS

serviceId	Cl id ac de se pa el AV **co** ar co o	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
SageMaker Runtime	sa _:	AWS_ENDPOINT_URL_SAGEMAKER_RUNTIME
savingsplans	sa ar	AWS_ENDPOINT_URL_SAVINGSPLANS
Scheduler	sc	AWS_ENDPOINT_URL_SCHEDULER
schemas	sc	AWS_ENDPOINT_URL_SCHEMAS
SimpleDB	s:	AWS_ENDPOINT_URL_SIMPLEDB
Secrets Manager	se ar	AWS_ENDPOINT_URL_SECRETS_MANAGER
SecurityHub	se ub	AWS_ENDPOINT_URL_SECURITYHUB
SecurityLake	se al	AWS_ENDPOINT_URL_SECURITYLAKE

serviceId	Cl id ac de se pa el AV **co** ar co o	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
ServerlessApplicat ionRepository	se sa io to	AWS_ENDPOINT_URL_SERVERLESSAPPLICATI ONREPOSITORY
Service Quotas	se uo	AWS_ENDPOINT_URL_SERVICE_QUOTAS
Service Catalog	se at	AWS_ENDPOINT_URL_SERVICE_CATALOG
Service Catalog AppRegistry	se at pi	AWS_ENDPOINT_URL_SERVICE_CATALOG_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

serviceId	Cl id ac de se pa el AV **ce** ar co o	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
signer	s:	AWS_ENDPOINT_URL_SIGNER
SimSpaceWeaver	s: ea	AWS_ENDPOINT_URL_SIMSPACEWEAVER
SMS	sr	AWS_ENDPOINT_URL_SMS
Snow Device Management	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

serviceId	Cl id ac de se pa el A\ **co** ar co o	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
SSO	ss	AWS_ENDPOINT_URL_SSO
SSO Admin	ss	AWS_ENDPOINT_URL_SSO_ADMIN
SSO OIDC	ss	AWS_ENDPOINT_URL_SSO_OIDC
SFN	st	AWS_ENDPOINT_URL_SFN
Storage Gateway	st at	AWS_ENDPOINT_URL_STORAGE_GATEWAY
STS	st	AWS_ENDPOINT_URL_STS
SupplyChain	su in	AWS_ENDPOINT_URL_SUPPLYCHAIN
Support	su	AWS_ENDPOINT_URL_SUPPORT
Support App	su pp	AWS_ENDPOINT_URL_SUPPORT_APP
SWF	sw	AWS_ENDPOINT_URL_SWF
synthetics	sy s	AWS_ENDPOINT_URL_SYNTHETICS

serviceId	Cl id ac de se pa el AV **co** ar co o	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
Textract	te	AWS_ENDPOINT_URL_TEXTRACT
Timestream InfluxDB	t: m_ b	AWS_ENDPOINT_URL_TIMESTREAM_INFLUXDB
Timestream Query	t: m_	AWS_ENDPOINT_URL_TIMESTREAM_QUERY
Timestream Write	t: m_	AWS_ENDPOINT_URL_TIMESTREAM_WRITE
tnb	tr	AWS_ENDPOINT_URL_TNB
Transcribe	t: e	AWS_ENDPOINT_URL_TRANSCRIBE
Transfer	t:	AWS_ENDPOINT_URL_TRANSFER
Translate	t:	AWS_ENDPOINT_URL_TRANSLATE
TrustedAdvisor	t: v:	AWS_ENDPOINT_URL_TRUSTEDADVISOR

serviceId	Cl id ac de se pa el AV **ce** ar co o	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
VerifiedPermissions	ve e: s	AWS_ENDPOINT_URL_VERIFIEDPERMISSIONS
Voice ID	vo	AWS_ENDPOINT_URL_VOICE_ID
VPC Lattice	vp ce	AWS_ENDPOINT_URL_VPC_LATTICE
WAF	wa	AWS_ENDPOINT_URL_WAF
WAF Regional	wa na	AWS_ENDPOINT_URL_WAF_REGIONAL
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

serviceId	Cl id ac de se pa el AV **co** ar co o	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
WorkMailMessageFlow	wo es w	AWS_ENDPOINT_URL_WORKMAILMESSAGEFLOW
WorkSpaces	wo s	AWS_ENDPOINT_URL_WORKSPACES
WorkSpaces Thin Client	wo s_ ie	AWS_ENDPOINT_URL_WORKSPACES_THIN_CLIENT
WorkSpaces Web	wo s_	AWS_ENDPOINT_URL_WORKSPACES_WEB
XRay	x:	AWS_ENDPOINT_URL_XRAY

Valores predeterminados de configuración inteligente

> **Note**
>
> Para obtener ayuda para comprender el diseño de las páginas de configuración o para interpretar la tabla Support by AWS SDKs and tools que aparece a continuación, consulte Cómo comprender las páginas de configuración de esta guía.

Con la función de configuración inteligente por defecto, AWS SDKs puede proporcionar valores predeterminados optimizados y predefinidos para otros ajustes de configuración.

Configure esta funcionalidad mediante lo siguiente:

`defaults_mode`- configuración de AWS **`config`** archivos compartidos, **`AWS_DEFAULTS_MODE`**: variable de entorno, **`aws.defaultsMode`**- Propiedad del sistema JVM: solo 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 una configuración AWS del SDK tiene un valor establecido de forma explícita, ese valor siempre tiene prioridad. Si una configuración AWS del SDK no tiene un valor establecido de forma explícita y no `defaults_mode` es igual a la antigua, esta función puede proporcionar diferentes valores predeterminados para diversas configuraciones optimizadas para tu aplicación. La configuración puede incluir lo siguiente: la configuración de comunicación HTTP, el comportamiento de los reintentos, la configuración del punto de conexión regional del servicio y, posiblemente, cualquier configuración relacionada con el SDK. Los clientes que utilizan esta característica pueden obtener nuevos valores predeterminados de configuración adaptados a los escenarios de uso habituales. Si su `defaults_mode` no es igual a su `legacy`, le recomendamos que realice pruebas en la aplicación cuando actualice el SDK, 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 una configuración predeterminada que varía según el SDK y que existía antes de la creación de `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 llaman Servicios de AWS desde el mismo modo Región de AWS.
- `cross-region`— Se basa en el modo estándar e incluye una optimización adaptada a las aplicaciones que llaman a Servicios de AWS 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. El SDK intenta descubrir el tiempo 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 puede consultar los metadatos de la instancia y datos de usuario, lo que puede 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 el SDK intenta volver a intentarlo. Consulte Comportamiento de los reintentos.
- `stsRegionalEndpoints`— Especifica cómo el SDK determina el Servicio de AWS punto final que utiliza para comunicarse con el AWS Security Token Service (AWS STS). Consulte AWS STS Puntos finales regionales.
- `s3UsEast1RegionalEndpoints`— Especifica cómo el SDK determina el punto de enlace del AWS 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 protocolo de enlace TLS desde el momento en que se envía el mensaje CLIENT HELLO hasta el momento en que el cliente y el servidor han negociado completamente los cifrados e intercambiado 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`
`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).

Support by AWS SDKs and tools

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

SDK	compatible	Notas o más información
AWS CLI v2	No	

SDK	compatible	Notas o más información
SDK para C++	Sí	Parámetros no optimizados: `stsRegionalEndpoints` , `s3UsEast1RegionalEndpoints` , `tlsNegotiationTimeoutInMillis` .
SDK para Go V2 (1.x)	Sí	Parámetros no optimizados: `retryMode` , `stsRegionalEndpoints` , `s3UsEast1RegionalEndpoints` .
SDK para Go 1.x (V1)	No	
SDK para Java 2.x	Sí	Parámetros no optimizados: `stsRegionalEndpoints` .
SDK para Java 1.x	No	
SDK para 3.x JavaScript	Sí	Parámetros no optimizados: `stsRegionalEndpoints` , `s3UsEast1RegionalEndpoints` , `tlsNegotiationTimeoutInMillis` . `connectTimeoutInMillis` se llama `connectionTimeout` .
SDK para 2.x JavaScript	No	
SDK para Kotlin	No	

SDK	compatible	Notas o más información
SDK para.NET 4.x	Sí	Parámetros no optimizados: `connectTimeoutInMillis` , `tlsNegotiationTimeoutInMillis` .
SDK para .NET 3.x	Sí	Parámetros no optimizados: `connectTimeoutInMillis` , `tlsNegotiationTimeoutInMillis` .
SDK para PHP 3.x	Sí	Parámetros no optimizados: `tlsNegotiationTimeoutInMillis` .
SDK para Python (Boto3)	Sí	Parámetros no optimizados: `tlsNegotiationTimeoutInMillis` .
SDK para Ruby 3.x	Sí	
SDK para Rust	No	
SDK para Swift	No	
Herramientas para V5 PowerShell	Sí	Parámetros no optimizados: `connectTimeoutInMillis` , `tlsNegotiationTimeoutInMillis` .

SDK	compatible	Notas o más información
Herramientas para la PowerShell V4	Sí	Parámetros no optimizados: `connectTimeoutInMillis` , `tlsNegotiationTimeoutInMillis` .

AWS Bibliotecas de Common Runtime (CRT)

Las bibliotecas AWS Common Runtime (CRT) son una biblioteca base de SDKs. 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 entre todas, lo SDKs que proporciona una mejor reutilización, optimización y precisión del código. Los paquetes son:

- `awslabs/aws-c-auth`: autenticación AWS del lado del cliente (proveedores de credenciales estándar y firma (sigv4))
- `awslabs/aws-c-cal`: tipos primitivos criptográficos, hashes (,, HMAC)MD5, firmantes SHA256, SHA256 AES
- `awslabs/aws-c-common`: Estructuras de datos básicas, tipos threading/synchronization primitivos, gestión 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 de AWS 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 de Amazon de gran ancho de banda EC2
- `awslabs/aws-c-sdkutils`: una biblioteca de utilidades para analizar y administrar perfiles AWS
- `awslabs/aws-checksums`: Multiplataforma, acelerada por hardware CRC32c y CRC32 con la posibilidad de recurrir a implementaciones de software eficientes

- awslabs/aws-lc: Biblioteca criptográfica de uso general mantenida por el equipo de AWS criptografía para sus clientes AWS y para ellos, 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 sitios excepto en Go y Rust. SDKs

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 AWS SDK para C++ o el AWS SDK 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.
Diagrama de dependencias de CRT que muestra cómo las bibliotecas CRT individuales se interrelacionan entre sí.

AWS SDKs Política de mantenimiento de herramientas y herramientas

Descripción general de

Este documento describe la política de mantenimiento de los kits y herramientas de desarrollo de AWS software (SDKs), incluidos los dispositivos móviles y el IoT SDKs, y sus dependencias subyacentes. AWS proporciona periódicamente a las Herramientas AWS SDKs y a las Herramientas actualizaciones que pueden incluir soporte para funciones nuevas o actualizadas AWS APIs, 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 la versión principal del SDK

El ciclo de vida de las versiones principales SDKs y de Tools consta de 5 fases, que se describen a continuación.

- Versión preliminar para desarrolladores (fase 0): durante esta fase, no SDKs se admiten, no se deben utilizar en entornos de producción y están pensadas únicamente para facilitar el acceso anticipado y recibir 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, SDKs 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.
- End-of-Support (Fase 4): cuando un SDK llegue al final del soporte, dejará de recibir 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.
Periodo de la política de mantenimiento

Ciclo de vida de la dependencia

La mayoría AWS SDKs tienen dependencias subyacentes, como tiempos de ejecución de idiomas, sistemas operativos o 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 la documentación del SDKs SDK, en la que se describe la ruta end-of-support y se enlaza con ella.

Para ver la lista de las principales versiones disponibles de AWS SDKs and Tools y en qué punto del ciclo de vida de mantenimiento se encuentran, consulte. Ciclo de vida de la versión

AWS SDKs y ciclo de vida de las versiones de Tools

En la siguiente tabla se muestra la lista de las principales versiones disponibles del kit de desarrollo de AWS software (SDK) y en qué parte del ciclo de vida del mantenimiento se encuentran, junto con los plazos correspondientes. Para obtener información detallada sobre el ciclo de vida de las versiones principales de las herramientas AWS SDKs y sus dependencias subyacentes, consulte. Política de mantenimiento

SDK	Versión principal	Fase actual	Fecha de disponibilidad general	Notas
AWS CLI	1.x	Anuncio de mantenimiento	2 de septiembre de 2013	Consulte el anuncio para conocer los detalles y las fechas
AWS CLI	2.x	Disponibilidad general	2 de octubre de 2020	
SDK para C++	1.x	Disponibilidad general	2/9/2015	
SDK para Go V2	V2 1.x	Disponibilidad general	19/1/2021	
SDK para Go	1.x	Fin-del-soporte	19/11/2015	
SDK para Java	1.x	Fin-del-soporte	25/03/2010	
SDK para Java	2.x	Disponibilidad general	20/11/2018	
SDK para JavaScript	1.x	Fin-del-soporte	6/05/2013	

SDK	Versión principal	Fase actual	Fecha de disponibilidad general	Notas
SDK para JavaScript	2.x	Fin-del-soporte	19/06/2014	
SDK para JavaScript	3.x	Disponibilidad general	15 de diciembre de 2020	
SDK para Kotlin	1.x	Disponibilidad general	27/11/2023	
SDK para .NET	1.x	Fin-del-soporte	11/2009	
SDK para .NET	2.x	Fin-del-soporte	8/11/2013	
SDK para .NET	3.x	Disponibilidad general	28/7/2015	
SDK para .NET	4.x	Disponibilidad general	28 de abril de 2025	
SDK para PHP	2.x	Fin-del-soporte	2/11/2012	
SDK para PHP	3.x	Disponibilidad general	27/05/2015	
SDK para Python (Boto2)	1.x	Fin-del-soporte	13/7/2011	
SDK para Python (Boto3)	1.x	Disponibilidad general	22/06/2015	
SDK para Python (Botocore)	1.x	Disponibilidad general	22/06/2015	
SDK para Ruby	1.x	Fin-del-soporte	14/7/2011	

SDK	Versión principal	Fase actual	Fecha de disponibilidad general	Notas
SDK para Ruby	2.x	Fin-del-soporte	15/02/2015	
SDK para Ruby	3.x	Disponibilidad general	29 de agosto de 2017	
SDK para Rust	1.x	Disponibilidad general	27/11/2023	
SDK para 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	
Herramientas para PowerShell	5.x	Disponibilidad general	23/06/2025	

¿Busca un SDK o una herramienta que no se mencione? El cifrado SDKs, los dispositivos SDKs IoT y los dispositivos móviles SDKs, por ejemplo, no se incluyen en esta guía. Para encontrar documentación sobre estas otras herramientas, consulte Herramientas sobre las que basarse en AWS.

Historial de documentos AWS SDKs y guía de referencia de herramientas

En la siguiente tabla se describen las adiciones y actualizaciones importantes de la Guía de referencia de herramientas AWS SDKs y herramientas. Para obtener notificaciones sobre las actualizaciones de esta documentación, puede suscribirse a la fuente RSS.

Cambio	Descripción	Fecha
Agregar un nuevo ajuste de S3 Express One Zone	Se ha agregado un nuevo ajuste de S3 Express One Zone para deshabilitar la autenticación de sesión.	13 de octubre de 2025
Agregar un nuevo árbol de decisiones de autenticación	Agregar un nuevo árbol de decisiones para facilitar las decisiones de autenticación entre las opciones.	23 de septiembre de 2025
Agregar una nueva caracterí stica de esquema de autentica ción	Agregar una nueva caracterí stica de esquema de autentica ción. Actualizaciones de los puntos finales AWS STS regionales.	18 de agosto de 2025
Agregar una nueva versión de Tools for PowerShell	Se ha añadido la última versión de Tools para dar PowerShell soporte a todas las referencias de configura ción compatibles con AWS SDKs las tablas. Se agregó la característica de inyección de prefijos de host.	23 de junio de 2025

Actualizaciones de títulos de páginas	Más títulos, títulos de tablas, resúmenes y actualizaciones de SEO.	5 de marzo de 2025
Actualizaciones de títulos de páginas	Actualización del contenido para utilizar títulos más descriptivos.	24 de febrero de 2025
Agregar el SDK de Swift a la referencia de configuración	Se ha añadido la compatibilidad con Swift SDK a todas las referencias de configuración: compatibilidad con AWS SDKs tablas.	17 de septiembre de 2024
Propiedades del sistema SDK para Java 1.x	Añada detalles sobre los ajustes de configuración del sistema JVM compatibles antes de la versión AWS SDK para Java 1.x.	30 de mayo de 2024
Actualizaciones de configuraciones	Agregue los ajustes de configuración del sistema JVM.	27 de marzo de 2024
Actualizaciones de tablas de compatibilidad	Actualizaciones de la compatibilidad para el soporte del SDK y actualizaciones de los procedimientos del IAM Identity Center.	20 de febrero de 2024
Actualización de credenciales del contenedor. Actualización del IMDS.	Agregar soporte para Amazon EKS. Se agregó una configuración para deshabilitar IMDSv1 la opción 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	Tablas de compatibilidad para SDK y características de herramientas actualizadas para incluir el SDK para Kotlin, SDK para Rust y Herramientas de AWS para PowerShell.	10 de diciembre de 2023
Actualizaciones de autenticación	Actualizaciones de los métodos de autenticación SDKs y las herramientas compatibles.	1 de julio de 2023
Actualizaciones de las prácticas recomendadas de IAM	Se ha actualizado la guía para implementar las prácticas recomendadas de IAM. Para obtener más información, consulta prácticas recomendadas de seguridad en IAM.	27 de febrero de 2023
Actualizaciones de SSO	Actualizaciones de las credenciales de SSO para la nueva configuración del token de SSO.	19 de noviembre de 2022
Actualizaciones de configuraciones	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 configuraciones	Se ha actualizado la claridad de las credenciales del cliente IMDS y del IMDS. 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 que ahora se hace referencia al AWS SSO 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

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.