Guía de referencia

AWS SDK y herramientas

AWS SDK y herramientas: Guía de referencia

Table of Contents

Guía de referencia de los SDK y herramientas de AWS 1
Recursos para desarrolladores 3
Configuración 4
Archivos compartidos `config` y `credentials` 5
Perfiles 5
Formato del archivo de configuración 7
Formato del archivo de credenciales 10
Ubicación de los archivos compartidos 10
Cambie la ubicación predeterminada de estos archivos 11
Variables de entorno 12
Cómo configurar las variables de entorno 13
Configuración de variables de entorno sin servidor 14
Propiedades del sistema JVM 14
¿Cómo configurar las propiedades del sistema JVM 15
Autenticación y acceso 17
ID de creador de AWS 19
Autenticación del Centro de identidades de IAM 19
Configuración del acceso mediante programación mediante el Centro de identidades de IAM 20
Comprender la autenticación del Centro de identidades de IAM 23
Funciones de IAM en cualquier lugar 27
Paso 1: Configurar las Funciones de IAM en cualquier lugar 28
Paso 2: Utilice las funciones de IAM en cualquier lugar 28
Asumir un rol 29
Asumir un rol de IAM. 30
Federar con identidad web u OpenID Connect 31
AWS claves de acceso 33
Use credenciales a corto plazo. 33
Use credenciales a largo plazo. 33
Credenciales a corto plazo 35
Credenciales a largo plazo 36
Roles de IAM para instancias Amazon EC2 39
Crear un rol de IAM 39
Lanzar una instancia de Amazon EC2 y especificar el rol de IAM 40

Conexión a la instancia EC2 ... 40
Ejecutar la aplicación de muestra en la instancia EC2 ... 41
Referencia de configuración ... 42
Creación de clientes de servicio ... 42
Prioridad de los ajustes ... 42
Lista de ajustes de archivos `Config` ... 44
Lista de ajustes de archivos `Credentials` ... 47
Lista de variables de entorno ... 48
Lista de propiedades del sistema JVM ... 51
Proveedores de credenciales estandarizadas ... 54
Cadena de proveedores de credenciales ... 55
AWS claves de acceso ... 56
Asumir el rol de proveedor ... 59
Proveedor de contenedores ... 66
Proveedor del Centro de identidades de IAM ... 69
Proveedor IMDS ... 76
Proveedor del proceso ... 81
Características estandarizadas ... 85
Metadatos de la instancia de Amazon EC2 ... 85
Puntos de acceso de Amazon S3 ... 88
Puntos de acceso multirregión de Amazon S3 ... 90
Región de AWS ... 92
AWS STS Puntos finales regionalizados ... 95
Puntos de conexión de doble pila y FIPS ... 98
Detección de puntos de conexión ... 100
Configuración general ... 102
Cliente IMDS ... 106
Comportamiento de los reintentos ... 109
Compresión de solicitudes ... 113
Puntos de conexión específicos del servicio ... 115
Valores predeterminados de configuración inteligente ... 166
Tiempo de ejecución común ... 172
Dependencias de CRT ... 173
Mantenimiento y soporte ... 174
Política de mantenimiento ... 174
Información general ... 174

Control de versiones 174
Ciclo de vida de las versiones principales del 175
Ciclo de vida de 176
Métodos de comunicación 176
Matriz de compatibilidad con versiones 177
Kits de herramientas IDE 180
Notificación de telemetría 180
Historial de documentos 182
Glosario de AWS 185
..... clxxxvi

Guía de referencia de los SDK y herramientas de AWS

Muchos SDK y herramientas comparten algunas funciones comunes, ya sea mediante especificaciones de diseño compartidas o mediante una biblioteca compartida.

Esta guía incluye información sobre:

- Configuración – Cómo utilizar los archivos compartidos `config` y `credentials` o las variables de entorno para configurar los SDK y las herramientas de AWS.
- Autenticación y acceso – Debe establecer cómo se autentica el código con AWS cuando desarrolla con Servicios de AWS.
- Referencia de configuración – Referencia para todos los ajustes estandarizados disponibles para la autenticación y la configuración.
- Bibliotecas de Common Runtime (CRT) AWS – Descripción general de las bibliotecas compartidas de AWS Common Runtime (CRT) que están disponibles para casi todos los SDK.
- Política de mantenimiento de SDK y herramientas AWS cubre la política de mantenimiento y el control de versiones para los kits y herramientas de desarrollo de software (SDK) de AWS, incluidos los SDK para dispositivos móviles e Internet de las cosas (IoT), y sus dependencias subyacentes.

Esta guía de referencia sobre los SDK y herramientas de AWS pretende ser una base de información aplicable a varios SDK y 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	Política de mantenimiento de SDK y herramientas AWS
• Guía para desarrolladores de AWS Cloud Development Kit (AWS CDK) • Guía para desarrolladores de AWS Serverless Application Model	Configuración Autenticación y acceso

Si utiliza:	Las secciones relevantes de esta guía para usted son:
• Guía del usuario de AWS Toolkit for Eclipse • Guía del usuario de AWS Toolkit for JetBrains • Guía del usuario de AWS Toolkit for Visual Studio • Guía del usuario de AWS Toolkit for Visual Studio Code	Política de mantenimiento de SDK y herramientas AWS
• Guía del usuario de AWS Command Line Interface • Guía para desarrolladores de AWS SDK for C++ • Guía para desarrolladores de AWS SDK for Go • Guía para desarrolladores de AWS SDK for Java • Guía para desarrolladores de AWS SDK for JavaScript • AWS SDK para Kotlin • Guía para desarrolladores de AWS SDK for .NET • Guía para desarrolladores de AWS SDK for PHP • Introducción a SDK para Python (Boto3) AWS • Guía para desarrolladores de AWS SDK for Ruby • AWS SDK para Rust • AWS SDK para Swift • Guía del usuario de AWS Tools for Windows PowerShell	Configuración Autenticación y acceso Referencia de configuración Bibliotecas de Common Runtime (CRT) AWS Política de mantenimiento de SDK y herramientas AWS

Recursos para desarrolladores

Amazon CodeWhisperer es un servicio con tecnología de Machine Learning (ML) que ayuda a mejorar la productividad de los desarrolladores mediante la generación de recomendaciones de código basadas en los comentarios de código y del código en el entorno de desarrollo integrado (IDE). Para obtener más información sobre los idiomas e IDE compatibles, así como sobre cómo registrarse para obtener una vista previa gratuita, consulte Amazon CodeWhisperer.

Configuración

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

La información incluye los siguientes elementos:

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

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

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

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

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

Temas adicionales en esta sección

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

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

Los `credentials` archivos AWS `config` y compartidos contienen un conjunto de perfiles. Un perfil es un conjunto de valores de configuración a los que se puede hacer referencia desde el SDK o la herramienta mediante el nombre de su perfil. Los valores de configuración se adjuntan a un perfil para configurar algún aspecto del SDK o la herramienta cuando se utiliza ese perfil. Estos archivos se “comparten”, ya que los valores se aplican a cualquier aplicación, proceso o SDK del entorno local de un usuario.

Como regla general, cualquier valor que se pueda colocar en el archivo compartido `credentials` también se puede colocar en el archivo compartido `config`. Lo contrario no es cierto; solo se pueden colocar algunos ajustes en el archivo `credentials`. Sin embargo, como práctica recomendada de seguridad, le recomendamos guardar todos los valores confidenciales, como los ID de clave de acceso y las claves secretas, en un archivo único `credentials`. De esta forma, puede proporcionar permisos independientes para cada archivo, si es necesario.

Tanto los archivos compartidos `config` y `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.

Perfiles

Los ajustes de los archivos compartidos `config` y `credentials` están asociados a un perfil específico. Con varios perfiles, puede crear diferentes configuraciones de ajustes para aplicarlas en diferentes escenarios.

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 nombrado puede tener un grupo diferente de ajustes.

`[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. Por ejemplo, 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.

Establece un perfil con nombre

En lugar de usar el `[default]` perfil, puede establecer un perfil con nombre.

Configure esta funcionalidad mediante una de las siguientes opciones:

`AWS_PROFILE`- variable de entorno

Todos los AWS CLI comandos y códigos del SDK utilizan la configuración de este perfil nombrado.

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

En el caso de los SDK que se ejecutan en una JVM, puedes configurar la propiedad del `aws.profile` sistema. Cuando el SDK crea un cliente de servicio, utiliza la configuración del perfil indicado, a menos que la configuración se anule en el código.

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: **`profile`**

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

La línea de definición de la sección `profile` nombra una agrupación de configuraciones que puede aplicar en diferentes escenarios. `[default]` es el único perfil que no requiere el identificador de sección `profile`. Para conocer mejor los perfiles con nombre, consulte la sección anterior sobre Perfiles.

En el siguiente ejemplo, se muestra un archivo `config` con un perfil `[default]`. Establece la configuración `region`.

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

El siguiente ejemplo muestra un archivo `config` con una línea de definición de sección `profile`. Utiliza el identificador `profile` seguido de un nombre único para el perfil. Todos los ajustes que sigan esta línea, hasta que se encuentre otra definición de sección, se incluirán en este perfil con nombre.

```
[profile developers]
```

```
...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 testers]
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 a fin de resolver AWS las credenciales. AWS IAM Identity Center Para obtener más información sobre la configuración de la autenticación de inicio de sesión único, consulte Autenticación del Centro de identidades de IAM. Un perfil está vinculado a una sección `sso-session` mediante un par clave-valor en el que `sso-session` es la clave y el nombre de la sección `sso-session` es el valor, como `sso-session = <name-of-sso-session-section>`.

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 sección `services` admite personalizaciones de puntos de conexión específicas del servicio y solo está disponible en los SDK y las herramientas que incluyen esta característica. Para ver si esta característica está disponible para tu SDK, consulta Compatibilidad con los AWS SDK 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 de Amazon DynamoDB . La sección `"local-dynamodb"` `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 esa palabra `profile`. Utilice únicamente el nombre exclusivo del perfil entre corchetes.

```
[dev]
...settings...
```

Solo puede almacenar un pequeño subconjunto de ajustes y valores en el archivo `credentials`. Por lo general, solo son aquellos con valores que se considerarían "secretos" o confidenciales, como los identificadores de clave de acceso y las claves secretas. La página de cada configuración de esta guía indica si se puede almacenar en el archivo `credentials` o solo en el archivo `config`.

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

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

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

Los archivos compartidos AWS `config` y `credentials` los archivos son archivos de texto sin formato que residen de forma predeterminada en una carpeta cuyo nombre `.aws` se encuentra en la carpeta «home» de su ordenador.

En Linux y macOS, suele aparecer como `~/.aws`. En Windows, es `%USERPROFILE%\.aws`.

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

Sistema operativo	Ubicación y nombre predeterminados de los archivos
	`%USERPROFILE%\.aws\credentials`

Un `~/` o `~` seguido del separador de ruta por defecto del sistema de archivos al principio de la ruta se resuelve comprobando, por orden,

1. (Todas las plataformas) La variable de entorno `HOME`
2. (Plataformas Windows) La variable de entorno `USERPROFILE`
3. (Plataformas Windows) La variable de entorno `HOMEDRIVE`, precedida de la variable de entorno `HOMEPATH` (por ejemplo, `$HOMEDRIVE$HOMEPATH`)
4. (Opcional según el SDK o la herramienta) Una función o variable de resolución de la ruta de inicio específica del SDK o de la herramienta

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

Cambie la ubicación predeterminada de estos archivos

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
```

Utilice las propiedades del sistema JVM

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:

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

Para obtener instrucciones sobre cómo configurar las propiedades del sistema JVM, consulte. the section called “¿Cómo configurar las propiedades del sistema JVM”

Compatibilidad con variables de entorno

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

Prioridad de las opciones

- Si especifica una configuración mediante su variable de entorno, esta anulará cualquier valor cargado desde un perfil en los archivos AWS `config` y `credentials` compartidos.
- Si especifica una configuración mediante un parámetro en la línea de comando AWS CLI, se invalidará cualquier valor de la variable de entorno correspondiente o un perfil en el 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 el símbolo del sistema de PowerShell, tal y como se muestra en los ejemplos anteriores, el valor se guarda únicamente durante la sesión actual. Para que el valor de la variable de entorno persista en todas las sesiones de PowerShell y del símbolo del sistema, guárdelo mediante la aplicación Sistema en el Panel de control. También puede definir la variable para todas las sesiones de PowerShell futuras añadiéndolo a su perfil de PowerShell. Consulte la documentación de PowerShell para obtener más información sobre el almacenamiento de variables de entorno o su persistencia entre 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 entorno de AWS Lambda en la Guía para desarrolladores de AWS Lambda.

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.

Soporte de propiedades del sistema JVM

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

Prioridad de las opciones

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

¿Cómo configurar las propiedades del sistema JVM

Puede configurar las propiedades del sistema JVM de varias maneras.

En la línea de comandos

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

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

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

Con una variable de entorno

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

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

Linux, macOS, or Unix

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

La configuración de la variable de entorno cambia el valor usado hasta el final de su sesión del intérprete de comandos o hasta que otorgue a la variable un valor diferente. Puede hacer que las

variables persistan en sesiones futuras configurándolas en el script de startup del intérprete de comandos.

Windows Command Prompt

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

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

En tiempo de ejecución

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

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

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

Autenticación y acceso

Debe establecer cómo se autentica el código con AWS cuando desarrolla con Servicios de AWS. Puede configurar el acceso a AWS mediante programación a los recursos de diferentes maneras, según el entorno y el acceso a AWS disponibles.

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

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

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

- Uso de roles de IAM para instancias Amazon EC2 – Utilizar roles de IAM para ejecutar de forma segura su aplicación en una instancia de Amazon EC2.
- Puede interactuar con AWS mediante programación usando el Centro de identidades de IAM de las siguientes maneras:
  - Utilice AWS CloudShell para ejecutar comandos AWS CLI desde la consola.
  - Se utiliza AWS Cloud9 para empezar a programar AWS utilizando un entorno de desarrollo integrado (IDE) con recursos AWS.
  - Para probar el espacio de colaboración basado en la nube para equipos de desarrollo de software, considere el uso de 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 a AWS, cree su aplicación de modo que solicite credenciales de seguridad temporales de AWS 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 para entonces intercambiarlo por credenciales de seguridad temporales en AWS que tienen asignado un rol de IAM con permisos para utilizar los recursos de la Cuenta de AWS.

Para aprender a configurar esto para su SDK o herramienta, consulte Federar con identidad web u OpenID Connect.

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

Más información sobre la administración de acceso

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

- Identidades IAM (usuarios, grupos de usuarios y roles) – Comprender los fundamentos de las identidades en AWS.
- Mejores prácticas 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.

ID de creador de AWS

Su ID de creador de AWS complementa cualquier Cuentas de AWS que ya tenga o que quiera crear. Si bien una Cuenta de AWS actúa como contenedor de los recursos de AWS que usted crea y proporciona un límite de seguridad para esos recursos, su ID de creador de AWS lo representa como individuo. Puede iniciar sesión con su cuenta de ID de creador de AWS para acceder a herramientas y servicios para desarrolladores, como Amazon CodeWhisperer y Amazon CodeCatalyst.

- Inicie sesión con ID de creador de AWS en la Guía del usuario de AWS Sign-In: aprenda a crear y usar un ID de creador de AWS y conozca qué proporciona el Builder ID.
- Autenticación con CodeWhisperer y AWS Toolkit - Builder ID en la guía del usuario de CodeWhisperer: aprenda cómo CodeWhisperer usa un ID de creador de AWS.
- Conceptos de CodeCatalyst - ID de creador de AWS en la guía del usuario de Amazon CodeCatalyst: aprenda cómo CodeCatalyst usa un ID de creador de AWS.

Autenticación del Centro de identidades de IAM

AWS IAM Identity Center es el método recomendado para proporcionar AWS credenciales cuando se desarrolla en un servicio que no esAWS informático. Por ejemplo, sería algo así como su entorno de desarrollo local. Si estás desarrollando en un AWS recurso, como Amazon Elastic Compute Cloud (Amazon EC2) AWS Cloud9o, te recomendamos que obtengas las credenciales de ese servicio.

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

- El portal de AWS acceso es la ubicación web en la que se inicia sesión manualmente en el Centro de identidades de IAM. El formato de la URL es `d-xxxxxxxxxx.awsapps.com/start` o *`your_subdomain`*`.awsapps.com/start`. Al iniciar sesión en el portal de AWS acceso, puede ver Cuentas de AWS los roles que se han configurado para ese usuario. Este procedimiento utiliza el portal de AWS acceso para obtener los valores de configuración que necesita para el proceso de autenticación del SDK o la herramienta.
- AWS CLI Se utiliza para configurar el SDK o la herramienta para que utilice la autenticación del Centro de Identidad de IAM para las llamadas a la API realizadas por el código. Este proceso único actualiza el AWS `config` archivo compartido, que luego es utilizado por el SDK o la herramienta al ejecutar el código.

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

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

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

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

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

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

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

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

   Cuando haya terminado, debería disponer de lo siguiente:

   - Un `Dev team` grupo.
   - Un conjunto de `PowerUserAccess` permisos adjunto al `Dev team` grupo.
   - El usuario se ha añadido al `Dev team` grupo.
3. Salga del portal e inicie sesión de nuevo para ver sus opciones Cuentas de AWS y para `Administrator` o`PowerUserAccess`. Seleccione esta opción `PowerUserAccess` cuando trabaje con su herramienta o SDK.

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

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

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

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

Inicie sesión a AWS través del portal de AWS acceso. Si 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: Configure los SDK y las Herramientas para usar el IAM Identity Center

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

   a. Consulte Instalación o actualización de la versión más reciente de la AWS CLI en la Guía del usuario deAWS 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. De lo contrario, puede encontrar 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, seleccione el conjunto de permisos correspondiente y, a continuación, seleccione la línea de comandos o el enlace de acceso programático. Utilice

el conjunto de permisos `PowerUserAccess` predefinido o el conjunto de permisos que usted o su empleador hayan creado para aplicar permisos de privilegio mínimo para el desarrollo.

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

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

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

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

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

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

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

   b. Si anteriormente configuraste el acceso a AWS, revisa tu AWS `credentials` archivo compartido para ver si hay algunoAWS claves de acceso. Debe eliminar todas las credenciales estáticas antes de que el SDK o la herramienta utilicen las credenciales del IAM Identity Center debido a la precedencia Cadena de proveedores de credenciales.

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

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

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

- Duración de la sesión del IAM Identity Center: Configure 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

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

Comprender la autenticación del Centro de identidades de IAM

Términos relevantes del Centro de identidades de IAM

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

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

Nombre del Centro de identidades de IAM	Nombre de la API del SDK	Descripción
Centro de identidades	`sso`	Aunque se haya cambiado el nombre de inicio de sesión único de AWS, los espacios de nombres de las API de `sso` mantendrán su nombre original por motivos de compatibilidad con versiones anteriores. Para más información, consulte Cambiar el nombre del Centro

Nombre del Centro de identidades de IAM	Nombre de la API del SDK	Descripción
		de identidades de IAM en la Guía del usuario de AWS IAM Identity Center.
Consola del Centro de identidades de IAM Consola administrativa		La consola que se utiliza para configurar el inicio de sesión único.
URL del portal de acceso a AWS		Una URL exclusiva de su cuenta del Centro de identidades de IAM, como `https://xxx.awsapps.com/start`. Inicie sesión en este portal con sus credenciales de inicio de sesión del Centro de identidades de IAM.
Sesión del Portal de Acceso del Centro de identidades de IAM	Sesión de autenticación	Proporciona un token de acceso al portador al intermediario.
Sesión del conjunto de permisos		La sesión de IAM que el SDK usa internamente para realizar las llamadas de Servicio de AWS. En las discusiones informales, es posible que vea que esto se denomina incorrectamente “sesión de roles”.

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

Comprenda la resolución de credenciales del SDK para Servicios de AWS

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

Iniciar una sesión en el portal de acceso a AWS

- Inicie el proceso iniciando sesión con sus credenciales.
  - Use el comando de `aws sso login` en el AWS Command Line Interface (AWS CLI). Esto inicia una nueva sesión en el IAM Identity Center si aún no tiene una sesión activa.
- Al iniciar una nueva sesión, recibirá un token de actualización y un token de acceso del IAM Identity Center. El AWS CLI también actualiza un archivo JSON de la caché del SSO con un nuevo token de acceso y un token de actualización, y lo pone a disposición de los SDK para que lo utilicen.

- Si ya tiene una sesión activa, el AWS CLI comando reutiliza la sesión existente y caducará cuando caduque la sesión existente. Para obtener información sobre cómo establecer la duración de una sesión del IAM Identity Center, consulte Configurar la duración de las sesiones del portal de acceso de los usuarios de AWS en la AWS IAM Identity CenterGuía del usuario.
  - La duración máxima de la sesión se ha ampliado a 90 días para reducir la necesidad de iniciar sesión con frecuencia.

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

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

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

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

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

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

Caducidad y actualización de la sesión

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

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

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

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

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

Funciones de IAM en cualquier lugar

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

clave privada necesarios para configurar Funciones de IAM en cualquier lugar como su proveedor de credenciales.

Paso 1: Configurar las Funciones de IAM en cualquier lugar

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

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

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

Paso 2: Utilice las funciones de IAM en cualquier lugar

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

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

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

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

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

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

Su administrador de la nube debe proporcionarle el certificado y la clave privada. Los tres valores del ARN se pueden copiar de AWS Management Console. El siguiente ejemplo muestra un archivo compartido `config` que configura la recuperación de credenciales temporales de la herramienta auxiliar.

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

Para ver los parámetros opcionales y los detalles adicionales de las herramientas auxiliares, consulte el Ayudante de credenciales de las Funciones de IAM en cualquier lugar en GitHub.

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

Asumir un rol

Para asumir un rol, se utiliza un conjunto de credenciales de seguridad temporales para acceder a los recursos de AWS a los que de otro modo usted no tendría acceso. Las credenciales temporales

incluyen un ID de clave de acceso, una clave de acceso secreta y un token de seguridad. Para obtener más información sobre las solicitudes de la API de AWS Security Token Service (AWS STS), consulte Acciones en la Referencia de la API de AWS Security Token Service.

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

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

Asumir un rol de IAM.

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

Paso 1: Configurar un rol de IAM

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

Paso 2: Configurar el SDK o la herramienta

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

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

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

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

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

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

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

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

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

Federar con identidad web u OpenID Connect

Al crear aplicaciones móviles o aplicaciones web basadas en clientes que requieren acceso a AWS, AWS STS devuelve un conjunto de credenciales de seguridad temporales para los usuarios federados que se autentican a través de un proveedor de identidades (IdP) público. Entre los ejemplos de proveedores de identidad públicos se incluyen Login with Amazon, Facebook, Google o cualquier proveedor de identidad compatible con OpenID Connect (OIDC). Con este método, los usuarios no necesitan su propia identidad AWS ni la de IAM.

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

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

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

Paso 2: Configurar el SDK o la herramienta

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

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

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

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

```
[profile web-identity]
```

```
role_arn=arn:aws:iam::123456789012:role/my-role-name
web_identity_token_file=/path/to/a/token
```

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

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

AWS claves de acceso

Use credenciales a corto plazo.

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

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

Use credenciales a largo plazo.

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

Gestione el acceso en todos Cuentas de AWS

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

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

Creación de usuarios de IAM únicamente para entornos aislados

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

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

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

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

Seguridad para las claves de acceso de los usuarios de IAM

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

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

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

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

Autenticar mediante credenciales a corto plazo

Recomendamos configurar su SDK o herramienta de Autenticación del Centro de identidades de IAM para utilizar para usar opciones de duración de sesión ampliada. Sin embargo, puede copiar y utilizar las credenciales temporales que están disponibles en el portal de acceso de AWS. 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.

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

1. Crear un archivo 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 acceso de AWS.
5. Siga estas instrucciones para Actualizar manualmente las credenciales para copiar el rol de IAM desde el portal de acceso de AWS.

    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 como PowerUserAccess o Developer.

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.

Autenticar mediante 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.

Si utilizas un usuario de IAM para ejecutar tu código, el SDK o la herramienta de tu entorno de desarrollo se autentican utilizando credenciales de usuario de IAM de larga duración en el archivo compartido AWS `credentials`. Revise el tema de las Prácticas recomendadas de seguridad en relación con 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 los 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 las credenciales almacenadas en el archivo compartido de AWS `credentials` se almacenan en texto no cifrado.

Guía adicional para administrar las credenciales de forma segura

Para obtener una descripción general de cómo administrar de forma segura las credenciales de AWS, consulte Prácticas recomendadas para administrar las claves de acceso de AWS en 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 usar un usuario de IAM para acceder a servicios de AWS, necesita una cuenta de AWS y unas credenciales de AWS.

1. Cree una cuenta.

   Para crear una cuenta de AWS, consulte Introducción: ¿es la primera vez que usa AWS? en la Guía de referencia de AWS Account Management.
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 desarrollar AWS o ejecutar 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.
  - Para 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.

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 el SDK de AWS CLI como con los SDK.
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 `credentials` de AWS compartido. Este archivo es `~/.aws/credentials` en sistemas Linux y macOS y `%USERPROFILE%\.aws\credentials` en Windows. Para obtener más información, consulte 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 roles de IAM para instancias Amazon EC2

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

Para una instancia de Amazon Elastic Compute Cloud, cree un rol de IAM y, a continuación, conceda a la instancia de Amazon EC2 acceso a dicho rol. Para obtener más información, consulte Roles de IAM para Amazon EC2 en la Guía del usuario de Amazon EC2 para instancias de Linux o Roles de IAM para Amazon EC2 en la Guía del usuario de Amazon EC2 para instancias de Windows.

Crear un rol de IAM

Cree un rol de IAM que conceda acceso de solo lectura a Amazon S3.

1. Inicie sesión en la AWS Management Console y abra la consola de IAM 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, seleccione Amazon EC2 y luego 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. Recuerde este nombre, ya que lo necesitará cuando lance su instancia de Amazon EC2.

Lanzar una instancia de Amazon EC2 y especificar el rol de IAM

Puede lanzar una instancia de Amazon EC2 con un rol de IAM; mediante la consola de Amazon EC2.

Siga las instrucciones para lanzar una instancia en la Guía del usuario de Amazon EC2 para instancias Linux o en la Guía del usuario de Amazon EC2 para instancias Windows.

Cuando llegue a la página Review Instance Launch (Revisar lanzamiento de instancia), seleccione Edit instance details (Editar detalles de la instancia). En IAM role, elija el rol de IAM que creó anteriormente. Complete el procedimiento siguiendo las instrucciones.

> **Note**
>
> Deberá crear o usar un grupo de seguridad y un par de claves existentes para conectarse a la instancia.

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

Conexión a la instancia EC2

Conéctese a la instancia EC2 para poder transferirle la aplicación de muestra y luego ejecutarla. Necesitará el archivo que contiene la parte privada del par de claves que usó para lanzar la instancia, es decir, el archivo PEM.

Para ello, siga el procedimiento de conexión de la Guía del usuario de instancias de Linux de Amazon EC2 o la Guía del usuario de instancias de Windows de Amazon EC2. Cuando se conecte, hágalo de forma que pueda transferir archivos de su máquina de desarrollo a su instancia.

Si utilizas un kit de herramientas de AWS, 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.

Ejecutar la aplicación de muestra en la instancia EC2

1. Copie los archivos de la aplicación de la unidad local a la instancia.

   Para obtener información sobre cómo compartir archivos a su instancia, consulte la Guía del usuario de Amazon EC2 para instancias de Linux o la Guía del usuario de Amazon EC2 para instancias de Windows.
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. Inicie sesión en la AWS Management Console y abra la consola de Amazon EC2 en https://console.aws.amazon.com/ec2/.
   b. Seleccione la instancia y desasocie el rol de IAM mediante Acciones, Configuración de la instancia. Asociar o reemplazar rol de IAM.
   c. Vuelva a ejecutar la aplicación y confirme que devuelve un error de autorización.

Referencia de configuración

Los SDK proporcionan API específicas para cada idioma. 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, los SDK cuentan con estrategias flexibles que permiten obtener credenciales para utilizarlas en tus solicitudes, mantener la configuración que se utilizará con cada servicio y obtener valores para utilizarlos en la configuración global.

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

- Proveedores de credenciales estandarizadas – Los proveedores de credenciales comunes están estandarizados en varios SDK.
- Características estandarizadas – Funciones comunes estandarizadas en varios SDK.

Creación de clientes de servicio

Para acceder mediante programación Servicios de AWS, los SDK utilizan una clase u objeto de cliente para cada uno. Servicio de AWS Por ejemplo, si su aplicación necesita acceder a Amazon EC2,su aplicación crearía un objeto de cliente de Amazon EC2 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 SDK, un objeto de cliente de servicio es inmutable, por lo que debes crear un cliente nuevo para cada servicio al que realices solicitudes y para realizar solicitudes al mismo servicio con una configuración diferente.

Prioridad de los ajustes

Los ajustes globales configuran las funciones, los proveedores de credenciales y otras funcionalidades compatibles con la mayoría de los SDK y que tienen un amplio impacto en todos los Servicios de AWS. Todos los SDK tienen una serie de lugares (o fuentes) que se 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 AWS CLI o AWS Tools for 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 cliente Servicio de AWS o un objeto de configuración o, a veces, al llamar a una API individual.

2. Solo en Java/Kotlin: la propiedad del sistema JVM correspondiente a este ajuste 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 comprueba la configuración en el `credentials` archivo compartido. Si está configurado, el cliente lo usa.
5. El `config` archivo compartido de la configuración. Si la configuración está presente, el SDK la usa.
   - La variable de `AWS_PROFILE` entorno o la propiedad del sistema `aws.profile` JVM se pueden usar para especificar el perfil que carga el SDK.
6. Los valores predeterminados proporcionados por el propio código fuente del SDK se utilizan en último lugar.

> **Note**
>
> Es posible que algunos SDK y herramientas se muestren en un orden diferente. Además, algunos SDK y herramientas admiten otros métodos de almacenamiento y recuperación de parámetros. Por ejemplo, AWS SDK for .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 AWS CLI o parámetro de API) o en el código anularía todos los demás valores de ese comando.

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

La configuración que se muestra en la siguiente tabla se puede asignar al AWS `config` archivo compartido. Son globales y afectan a todos Servicios de AWS.

Nombre del conjunto	Detalles
`api_versions`	Ajustes de configuración general
`aws_access_key_id`	AWS claves de acceso
`aws_secret_access_key`	AWS claves de acceso
`aws_session_token`	AWS claves de acceso
`ca_bundle`	Ajustes de configuración general
`credential_process`	Proveedor de credenciales de proceso
`credential_source`	Asumir el rol de proveedor de credenciales
`defaults_mode`	Valores predeterminados de configuración inteligente
`disable_request_compression`	Compresión de solicitudes
`duration_seconds`	Asumir el rol de proveedor de credenciales

Nombre del conjunto	Detalles
`ec2_metadata_service_endpoint`	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 la instancia de Amazon EC2
`metadata_service_timeout`	Metadatos de la instancia de Amazon EC2
`mfa_serial`	Asumir el rol de proveedor de credenciales

Nombre del conjunto	Detalles
`output`	Ajustes de configuración general
`parameter_validation`	Ajustes de configuración general
`region`	Región de AWS
`request_min_compression_size_bytes`	Compresión de solicitudes
`retry_mode`	Comportamiento de los reintentos
`role_arn`	Asumir el rol de proveedor de credenciales
`role_session_name`	Asumir el rol de proveedor de credenciales
`s3_disable_multiregion_access_points`	Puntos de acceso multirregión de Amazon S3
`s3_use_arn_region`	Puntos de acceso de Amazon S3
`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

Nombre del conjunto	Detalles	
`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 finales regionalizados	
`use_dualstack_endpoint`	Puntos de conexión de doble pila y FIPS	
`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 muestran en la siguiente tabla se pueden asignar en el archivo compartido AWS `credentials`. Son globales y afectan a todos Servicios de AWS.

Nombre del conjunto	Detalles	
`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 admitidas por la mayoría de los SDK se indican en la siguiente tabla. Son globales y afectan a todos Servicios de AWS.

Nombre del conjunto	Detalles	
`AWS_ACCESS_KEY_ID`	AWS claves de acceso	
`AWS_CA_BUNDLE`	Ajustes de configuración general	
`AWS_CONFIG_FILE`	Ubicación de los archivos compartidos `config` y `credentials`	
`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_CREDENTIALS_RELATIVE_URI`	Proveedor de credenciales de contenedor	
`AWS_DEFAULTS_MODE`	Valores predeterminados de configuración inteligente	

Nombre del conjunto	Detalles
AWS_DISABLE_REQUEST_COMPRESSION	Compresión de solicitudes
AWS_EC2_METADATA_DISABLED	Proveedor de credenciales IMDS
AWS_EC2_METADATA_SERVICE_ENDPOINT	Proveedor de credenciales IMDS
AWS_EC2_METADATA_SERVICE_ENDPOINT_MODE	Proveedor de credenciales IMDS
AWS_EC2_METADATA_V1_DISABLED	Proveedor de credenciales IMDS
AWS_ENABLE_ENDPOINT_DISCOVERY	Detección de puntos de conexión
AWS_ENDPOINT_URL	Puntos de conexión específicos del servicio
AWS_ENDPOINT_URL_<SERVICE>	Puntos de conexión específicos del servicio
AWS_IAM_ROLE_ARN	Asumir el rol de proveedor de credenciales

Nombre del conjunto	Detalles
AWS_IAM_ROLE_SESSION_NAME	Asumir el rol de proveedor de credenciales
AWS_IGNORE_CONFIGURED_ENDPOINT_URLS	Puntos de conexión específicos del servicio
AWS_MAX_ATTEMPTS	Comportamiento de los reintentos
AWS_METADATA_SERVICE_NUM_ATTEMPTS	Metadatos de la instancia de Amazon EC2
AWS_METADATA_SERVICE_TIMEOUT	Metadatos de la instancia de Amazon EC2
AWS_PROFILE	Archivos compartidos `config` y `credentials`
AWS_REGION	Región de AWS
AWS_REQUEST_MIN_COMPRESSION_SIZE_BYTES	Compresión de solicitudes
AWS_RETRY_MODE	Comportamiento de los reintentos
AWS_S3_DISABLE_MULTIREGION_ACCESS_POINTS	Puntos de acceso multirregión de Amazon S3

Nombre del conjunto	Detalles
AWS_S3_USE_ARN_REGION	Puntos de acceso de Amazon S3
AWS_SECRET_ACCESS_KEY	AWS claves de acceso
AWS_SESSION_TOKEN	AWS claves de acceso
AWS_SHARED_CREDENTIALS_FILE	Ubicación de los archivos compartidos `config` y `credentials`
AWS_STS_REGIONAL_ENDPOINTS	AWS STS puntos finales regionalizados
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 for Java y para AWS SDK para Kotlin (dirigidas a la JVM). Consulte the section called “¿Cómo configurar las propiedades del sistema JVM” para obtener instrucciones sobre cómo configurar las propiedades del sistema JVM.

Nombre del conjunto	Detalles
`aws.accessKeyId`	AWS claves de acceso
`aws.configFile`	Ubicación de los archivos compartidos `config` y `credentials`
`aws.defaultsMode`	Valores predeterminados de configuración inteligente
`aws.disableEc2MetadataV1`	Proveedor de credenciales IMDS
`aws.disableRequestCompression`	Compresión de solicitudes
`aws.ec2MetadataServiceEndpoint`	Proveedor de credenciales IMDS
`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

Nombre del conjunto	Detalles
`aws.ignoreConfiguredEndpointUrls`	Puntos de conexión específicos del servicio
`aws.maxAttempts`	Comportamiento de los reintentos
`aws.profile`	Archivos compartidos `config` y `credentials`
`aws.region`	Región de AWS
`aws.requestMinCompressionSizeBytes`	Compresión de solicitudes
`aws.retryMode`	Comportamiento de los reintentos
`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

Nombre del conjunto	Detalles
`aws.sharedCredentialsFile`	Ubicación de los archivos compartidos `config` y `credentials`
`aws.useDualstackEndpoint`	Puntos de conexión de doble pila y FIPS
`aws.useFipsEndpoint`	Puntos de conexión de doble pila y FIPS
`aws.webIdentityTokenFile`	Asumir el rol de proveedor de credenciales

Proveedores de credenciales estandarizadas

Muchos proveedores de credenciales se han estandarizado con valores predeterminados consistentes y para que funcionen de la misma manera en muchos SDK. Esta coherencia aumenta la productividad y la claridad a la hora de codificar en varios SDK. Todos los ajustes se pueden anular en el código. Para obtener más detalles, consulte su API específica de SDK.

> **Important**
>
> No todos los SDK son compatibles con todos los proveedores, ni siquiera con todos los aspectos de un proveedor.

Temas

- Cadena de proveedores de credenciales
- AWS claves de acceso
- 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

Cadena de proveedores de credenciales

Todos los SDK tienen una serie de lugares (o fuentes) que consultan para encontrar credenciales válidas y utilizarlas para realizar una solicitud a un Servicio de AWS. Una vez que se encuentran las credenciales válidas, se detiene la búsqueda. A esta búsqueda sistemática, se le denomina cadena predeterminada de proveedores de credenciales.

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

Proveedor de credenciales	Descripción
AWS claves de acceso	Claves de acceso a AWS para un usuario de IAM (como `AWS_ACCESS_KEY_ID` y `AWS_SECRET_ACCESS_KEY`).
Federar con identidad web u OpenID Connect - Asumir el rol de proveedor de credenciales	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). Asuma los permisos de un rol de IAM mediante un token de identidad web de AWS Security Token Service (AWS STS).
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 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	Descripción
Proveedor de credenciales de proceso	Proveedor de credenciales personalizadas. Obtenga sus credenciales de un origen o proceso externo, incluidas las Funciones de IAM en cualquier lugar.
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 Archivos `config` y `credentials` compartidos. Para obtener más información, consulte Prioridad de los ajustes.

AWS claves de acceso

> **Warning**
>
> Para evitar riesgos de seguridad, no utilice a 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 credenciales AWS . 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 en 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 en 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 en Java/Kotlin

Especifica un token AWS de 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 Autenticar mediante credenciales a corto plazo.

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

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

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

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

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

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

Compatibilidad con los SDK AWS

Los siguientes SDK admiten las características y los ajustes descritos en este tema. Se anotan todas las excepciones parciales. Todos los ajustes de propiedades del sistema JVM son compatibles con AWS SDK for Java y únicamente. AWS SDK para Kotlin

SDK	Ce	Notas o más información
AWS CLI v2	Sí	

SDK	Ce	Notas o más información
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 para Java 1.x	Sí	
SDK para 3.x JavaScript	Sí	
SDK para 2.x JavaScript	Sí	
SDK para Kotlin	Sí	
SDK para .NET 3.x	Sí	No se admiten variables de entorno.
SDK para PHP 3.x	Sí	
SDK para Python (Boto3)	Sí	
SDK para Ruby 3.x	Sí	
SDK para Rust	Sí	
Herramientas para PowerShell	Sí	No se admiten variables de entorno.

Asumir el rol de proveedor de credenciales

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 la función puede ser una u otra Servicio de AWS Cuenta de AWS, un proveedor de identidad web o 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 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 de ECS para obtener las credenciales de origen.

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

Ejemplo de configuración en un archivo de `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 AWS **`config`** archivos compartidos

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

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

Valor predeterminado: 3600 segundos (una hora).

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

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. Para obtener más información sobre MFA, consulte Configuración de acceso a una API protegida por MFA en la Guía de usuario de IAM.

Ejemplo de esta configuración en un archivo `config`:

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

`role_arn`- configuración de AWS **`config`** archivos compartidos, **`AWS_IAM_ROLE_ARN`** - variable de entorno, **`aws.roleArn`**- Propiedad del sistema JVM: solo en 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 archivos compartidos AWS **`config`**, **`AWS_IAM_ROLE_SESSION_NAME`** - variable de entorno, **`aws.roleSessionName`**- Propiedad del sistema JVM: solo en 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.

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 API AWS CLI o llama a la `AssumeRole` operación (o a operaciones como la `AssumeRoleWithWebIdentity` operación) en tu 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 compartidos `config` y `credentials`.

Si especifica un perfil que también sea un perfil de asunción de roles, cada rol se asumirá en orden secuencial para resolver completamente las credenciales. Esta cadena se detiene cuando 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 sobre los 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
```

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

En el ejemplo anterior, el perfil `A` utiliza las credenciales en el perfil `B`. Al especificar que un comando del SDK o de la herramienta debe utilizar el perfil `A`, el SDK o la herramienta busca automáticamente las credenciales del perfil vinculado de `B` y las utiliza para solicitar credenciales temporales para el rol de IAM especificado. Para ello, el SDK o la herramienta utilizan la AssumeRole operación sts: en segundo plano. Luego, su código utiliza esas credenciales temporales para acceder a AWS los recursos. El rol especificado debe tener adjuntas políticas de permisos de IAM que permitan ejecutar el código solicitado, como el comando o el método de API. Servicio de AWS

`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 en Java/Kotlin

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

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 de `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 de OAuth 2.0 o un token de OpenID Connect que le proporcionó un proveedor de identidades. Las rutas relativas se consideran relativas al directorio de trabajo del proceso.

AWS Compatibilidad con los SDK

Los siguientes SDK admiten las características y los ajustes descritos en este tema. Se anotan todas las excepciones parciales. Todos los ajustes de propiedades del sistema JVM son compatibles con AWS SDK for Java y únicamente. AWS SDK para Kotlin

SDK	Ce	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 admitido. Úselo `AWS_ROLE_ARN` en lugar de. `AWS_IAM_ROLE_ARN` Úselo `AWS_ROLE_SESSION_NAME` en lugar de`AWS_IAM_ROLE_SESSION_NAME` .
SDK para Java 1.x	Parci	`mfa_serial` no admitido.
SDK para JavaScript 3.x	Sí	
SDK para 2.x JavaScript	Parci	`credential_source` no admitido.
SDK para Kotlin	Sí	Úselo `AWS_ROLE_ARN` en lugar de. `AWS_IAM_ROLE_ARN` Úselo `AWS_ROLE_SESSION_NAME` en lugar de`AWS_IAM_ROLE_SESSION_NAME` .
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	
Herramientas para PowerShell	Sí	

Proveedor de credenciales de contenedor

El proveedor de credenciales del contenedor obtiene las credenciales de la aplicación contenerizada del cliente. Este proveedor de credenciales es útil para los clientes de Amazon Elastic Container Service (Amazon ECS) y Amazon Elastic Kubernetes Service (Amazon EKS). Los SDK intentan cargar las credenciales desde el punto de conexión 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. Una vez configurado, Amazon ECS establece la variable `AWS_CONTAINER_CREDENTIALS_RELATIVE_URI` de entorno que los SDK y las herramientas utilizan para obtener 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.. Una vez configurado, Amazon EKS establece las variables `AWS_CONTAINER_CREDENTIALS_FULL_URI` y `AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE` de entorno que los SDK y las herramientas utilizan para obtener credenciales. 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
```

Compatibilidad con los SDK AWS

Los siguientes SDK admiten las características y los ajustes descritos en este tema. Se anotan todas las excepciones parciales. Todos los ajustes de propiedades del sistema JVM son compatibles con AWS SDK for Java y únicamente. AWS SDK para Kotlin

SDK	Ce	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	
SDK para Java 1.x	Parci	Amazon EKS Pod Identity y `AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE` no son compatibles.

SDK	Ce	Notas o más información
SDK para 3.x JavaScript	Sí	
SDK para 2.x JavaScript	Parci	Amazon EKS Pod Identity y `AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE` no son compatibles.
SDK para Kotlin	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	Amazon EKS Pod Identity y `AWS_CONTAINER_AUTHORIZATION_TOKEN_FILE` no son compatibles.
Herramientas para PowerShell	Sí	

Proveedor de credenciales del IAM Identity Center

Este mecanismo de autenticación se utiliza AWS IAM Identity Center para obtener acceso a tu código mediante el 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 al Centro de identidades de IAM. Cuando un usuario se autentica correctamente en el Centro de identidades de IAM, 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, consulteComprender la autenticación del Centro de identidades de IAM.

Hay dos formas de configurar el Centro de identidades de IAM a través del archivo `config`:

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

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

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

Las dos guías siguientes contienen información adicional sobre el Centro de identidades de IAM:

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

Requisitos previos

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

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

Configuración del proveedor de token de SSO

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

Al utilizar la configuración del proveedor de token de SSO, el AWS SDK o la herramienta actualizan automáticamente la sesión hasta el período de sesión extendido. Para obtener más información sobre la duración y la duración máxima de la sesión, consulte Configurar la duración de la sesión del 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 el formato de las secciones de un archivo `config`, consulte Formato del archivo de configuración.

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

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

En el siguiente ejemplo se configura el SDK para que solicite credenciales de Centro de identidades de IAM. También admite la actualización automática de los tokens.

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

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

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

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

```
sso_role_name = SampleRole

[profile prod]
sso_session = my-sso
sso_account_id = 111122223333
sso_role_name = SampleRole2

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

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 una autenticación de portador compatible, 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 guía individual Servicio de AWS para determinar si admite la autorización de un 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. Una solicitud puede pedir uno o varios ámbitos y el token de acceso emitido a la solicitud se limita a los ámbitos concedidos. Estos ámbitos definen los permisos cuya autorización se solicita para el cliente OIDC registrado y los tokens de acceso recuperados por el cliente. Para ver las opciones de ámbito de acceso compatibles, consulte los ámbitos de acceso en la Guía del usuario de AWS IAM Identity Center . El siguiente ejemplo establece `sso_registration_scopes` para proporcionar acceso para enumerar cuentas y roles.

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

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

Configuración heredada no actualizable

La actualización automática de tokens no se admite con la configuración no actualizable heredada. Se recomienda utilizar el 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 al portal de acceso al Centro de identidades de IAM de su organización. Para obtener más información sobre el portal de acceso al Centro de Identidad de IAM, consulte Uso del portal de AWS acceso en la Guía del AWS IAM Identity Center usuario.

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

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

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 Centro de identidades de IAM, 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 Centro de identidades de IAM 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 asignado.

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`. Los ámbitos autorizan el acceso a los puntos de conexión autorizados por el token

de portador del Centro de identidades de IAM. Para recuperar un token de actualización del servicio del Centro de identidades de IAM, se debe conceder un límite mínimo de `sso:account:access`. Para ver las opciones de ámbito de acceso compatibles, consulte los Ámbitos de acceso en la Guía del usuario de AWS IAM Identity Center . Esta configuración no aplica a la configuración heredada no actualizable. Los tokens emitidos con la configuración heredada tienen un alcance limitado de `sso:account:access` de forma implícita.

Compatibilidad con los AWS SDK

Los siguientes SDK admiten las características y los ajustes descritos en este tema. Se anotan todas las excepciones parciales. Todos los ajustes de propiedades del sistema JVM son compatibles con AWS SDK for Java y únicamente. AWS SDK para Kotlin

SDK	Ce	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 3.x	Sí	

SDK	Ce	Notas o más información
SDK para PHP 3.x	Sí	
SDK para Python (Boto3)	Sí	
SDK para Ruby 3.x	Sí	
SDK para Rust	Parci	Solo configuración heredada no actualizable.
Herramientas para PowerShell	Sí	

Proveedor de credenciales IMDS

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 sobre el tipo de instancias consulte Metadatos de instancias y datos de usuario en la Guía del usuario de Amazon EC2 para instancias de Linux o Metadatos de instancias y datos de usuario tipo de instancias en la Guía del usuario de Amazon EC2 para instancias de Windows. 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. Los SDK pueden usar ese punto de conexión para resolver las credenciales como parte de su cadena de proveedores predeterminados de credenciales. De forma predeterminada, se utiliza Instance Metadata Service Version 2 (IMDSv2), una versión más segura de IMDS que utiliza 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), se utiliza IMDSv1 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 AWS **`config`** archivos compartidos, **`AWS_EC2_METADATA_V1_DISABLED`** - variable de entorno, **`aws.disableEc2MetadataV1`**- Propiedad del sistema JVM: solo en Java/Kotlin

Si se debe utilizar o no el Servicio de metadatos de instancia versión 1 (IMDSv1) como alternativa en caso de que IMDSv2 falle.

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

Valor predeterminado: `false`.

Valores válidos:

- **`true`**: no utilice el IMDSv1 como alternativa.
- **`false`**: utilice el IMDSv1 como alternativa.

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

El tipo de punto de conexión.

Valor predeterminado: si `ec2_metadata_service_endpoint_mode` es igual a `IPv4`, el punto de conexión predeterminado es `http://169.254.169.254`. Valor predeterminado: si `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 en 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 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 de `AWS_EC2_METADATA_DISABLED` entorno 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.

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 shell, 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 shell y se reinicie el sistema, utilice el siguiente comando en su lugar:

```
$ setx AWS_EC2_METADATA_DISABLED=true
```

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

Compatibilidad con los SDK AWS

Los siguientes SDK admiten las características y los ajustes descritos en este tema. Se anotan todas las excepciones parciales. Todos los ajustes de propiedades del sistema JVM son compatibles con AWS SDK for Java y únicamente. AWS SDK para Kotlin

SDK	C 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	No utilice el IMDSv1 como alternativa.
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	No utilice el IMDSv1 como alternativa.
Herramientas para PowerShell	Sí	Puede deshabilitar el respaldo de IMDSv1 de forma explícita en el código mediante. `[Amazon.Util.EC2InstanceMetadata]::EC2MetadataV1Disabled = $true`

Proveedor de credenciales de proceso

Los SDK proporcionan una forma de ampliar la cadena de proveedores de credenciales para casos de uso personalizados.

IAM Roles Anywhere proporciona una forma de obtener credenciales temporales para una carga de trabajo o un proceso que se ejecuta fuera de. AWS Para configurar `credential_process` para este uso, consulte Funciones de IAM en cualquier lugar.

> **Warning**
>
> A continuación se describe un método para obtener credenciales de un proceso externo. Esto puede resultar peligroso, así que proceda con precaución. Si es posible, se debe dar preferencia a 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. Asegúrese de que la herramienta de credenciales personalizada no escriba ninguna información secreta en `StdErr`, ya que los SDK y la AWS CLI pueden capturar y registrar dicha información y podrían mostrarla 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 del programa o comando 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 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`.

  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"
  ```

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 `STDOUT` 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 clave `Expiration` es una marca temporal con formato RFC3339. Si la clave de `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.

Compatibilidad con los AWS SDK

Los siguientes SDK admiten las características y los ajustes descritos en este tema. Se anotan todas las excepciones parciales. Todos los ajustes de propiedades del sistema JVM son compatibles con AWS SDK for Java y únicamente. AWS SDK para Kotlin

SDK	Ce	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 3.x	Sí	
SDK para PHP 3.x	Sí	
SDK para Python (Boto3)	Sí	
SDK para Ruby 3.x	Sí	
SDK para Rust	Sí	
Herramientas para PowerShell	Sí	

Características estandarizadas

Muchas funciones se han estandarizado con valores predeterminados consistentes y para que funcionen de la misma manera en muchos SDK. Esta coherencia aumenta la productividad y la claridad a la hora de codificar en varios SDK. Todos los ajustes se pueden anular en el código. Consulta la API específica del SDK para obtener más información.

> **Important**
>
> No todos los SDK admiten todas las características, ni siquiera todos los aspectos de una característica.

Temas

- Metadatos de la instancia de Amazon EC2
- Puntos de acceso de Amazon S3
- Puntos de acceso multirregión de Amazon S3
- Región de AWS
- AWS STS Puntos finales regionalizados
- Puntos de conexión de doble pila y FIPS
- Detección de puntos de conexión
- Ajustes de configuración general
- Cliente IMDS
- Comportamiento de los reintentos
- Compresión de solicitudes
- Puntos de conexión específicos del servicio
- Valores predeterminados de configuración inteligente

Metadatos de la instancia de Amazon EC2

Amazon EC2 proporciona un servicio en instancias denominado Servicio de metadatos de instancias (IMDS). Para obtener más información sobre este servicio, consulte Metadatos de instancias y datos de usuario en la Guía del usuario de Amazon EC2 para instancias de Linux o Metadatos de

instancias y datos de usuario en la Guía del usuario de Amazon EC2 para instancias de Windows. 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 recuperan 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

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

Valor predeterminado: 1

Valores válidos: número mayor o igual a 1.

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

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

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

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

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

```
setx AWS_METADATA_SERVICE_NUM_ATTEMPTS 10
```

```
setx AWS_METADATA_SERVICE_TIMEOUT 10
```

Compatibilidad con los AWS SDK

Los siguientes SDK admiten las características y los ajustes descritos en este tema. Se anotan todas las excepciones parciales. Todos los ajustes de propiedades del sistema JVM son compatibles con AWS SDK for Java y únicamente. AWS SDK para Kotlin

SDK	C e	Notas o más información
AWS CLI v2	Sí	
SDK para C++	No	
SDK para Go V2 (1.x)	No	
SDK para Go 1.x (V1)	No	
SDK para Java 2.x	No	
SDK para Java 1.x	Parci	`metadata_service_num_attempts` no admitido.
SDK para 3.x JavaScript	No	
SDK para 2.x JavaScript	No	
SDK para Kotlin	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	
Herramientas para PowerShell	No	

Puntos de acceso de Amazon S3

El servicio Amazon S3 proporciona puntos de acceso como una forma alternativa de interactuar con los buckets de Amazon S3. Los puntos de acceso pueden tener políticas y configuraciones únicas que se pueden aplicar a ellos en lugar de directamente al bucket. Con AWS los SDK, puedes usar los nombres de recursos de Amazon (ARN) del punto de acceso 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 y los ARN, 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 archivos compartidos AWS **`config`**, **`AWS_S3_USE_ARN_REGION`** - variable de entorno, **`aws.s3UseArnRegion`**- Propiedad del sistema JVM: solo en 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.

Compatibilidad con los SDK AWS

Los siguientes SDK admiten las características y los ajustes descritos en este tema. Se anotan todas las excepciones parciales. Todos los ajustes de propiedades del sistema JVM son compatibles con AWS SDK for Java y únicamente. AWS SDK para Kotlin

SDK	C 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 3.x	Sí	No sigue la prioridad estándar; el valor del archivo compartido `config` tiene prioridad sobre la variable de entorno.
SDK para PHP 3.x	Sí	
SDK para Python (Boto3)	Sí	
SDK para Ruby 3.x	Sí	
SDK para Rust	No	

SDK	Ce	Notas o más información
Herramientas para PowerShell	Sí	No sigue la prioridad estándar; el valor del archivo compartido `config` tiene prioridad sobre la variable de entorno.

Puntos de acceso multirregión de Amazon S3

Los puntos de acceso 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 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 conocer más sobre cómo funcionan los Nombres de recurso de Amazon (ARN) multirregión, consulte Realizar solicitudes mediante un punto de acceso multirregión 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 Bibliotecas de Common Runtime (CRT) AWS.

Configure esta funcionalidad mediante lo siguiente:

`s3_disable_multiregion_access_points`- configuración de AWS **`config`** archivos compartidos, **`AWS_S3_DISABLE_MULTIREGION_ACCESS_POINTS`** - variable de entorno, **`aws.s3DisableMultiRegionAccessPoints`**- Propiedad del sistema JVM: solo en 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.

AWS Compatibilidad con los SDK

Los siguientes SDK admiten las características y los ajustes descritos en este tema. Se anotan todas las excepciones parciales. Todos los ajustes de propiedades del sistema JVM son compatibles con AWS SDK for Java y únicamente. AWS SDK para Kotlin

SDK	Ce	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 3.x	Sí	
SDK para PHP 3.x	Sí	
SDK para Python (Boto3)	Sí	
SDK para Ruby 3.x	Sí	

SDK	Ce	Notas o más información
SDK para Rust	Sí	
Herramientas para PowerShell	Sí	

Región de AWS

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 compatibles con cada una de ellas Servicio de AWS, consulte Cuotas y puntos de conexión del servicio.

Creación de clientes de servicio

Para acceder mediante programación Servicios de AWS, los SDK utilizan una clase u objeto de cliente para cada uno. 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, 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` AWS **`config`**- configuración de archivos compartidos, **`AWS_REGION`** - variable de entorno, **`aws.region`**- Propiedad del sistema JVM: solo en Java/Kotlin

Especifica el valor predeterminado que se debe usar en Región de AWS las solicitudes. AWS 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 final en el este de EE. Región de AWS UU. (Virginia del Norte).
- `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 de los SDK tienen un objeto de “configuración” que permite configurar la región predeterminada desde el código de la aplicación. Para obtener más información, consulte la guía específica AWS para desarrolladores del SDK.

Compatibilidad con los AWS SDK

Los siguientes SDK admiten las características y los ajustes descritos en este tema. Se anotan todas las excepciones parciales. Todos los ajustes de propiedades del sistema JVM son compatibles con AWS SDK for Java y únicamente. AWS SDK para Kotlin

SDK	Ce	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	Ce	Notas o más información
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	
Herramientas para PowerShell	Sí	

AWS STS Puntos finales regionalizados

De forma predeterminada, AWS Security Token Service (AWS STS) está disponible como un servicio global y todas las AWS STS solicitudes se envían a un único punto final en`https://sts.amazonaws.com`. Las solicitudes globales se asignan a la región EE. UU. Este (Virginia del Norte). AWS recomienda utilizar los AWS STS puntos finales regionales en lugar del punto final global. Para obtener más información sobre los AWS STS puntos finales, consulte los puntos finales en la referencia de la AWS Security Token Service API.

Configure esta funcionalidad mediante lo siguiente:

`sts_regional_endpoints`- configuración de archivos compartidos AWS **`config`**, **`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: `legacy`

> **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`,, para las siguientes AWS regiones: `ap-northeast-1``ap-south-1`,,`ap-southeast-1`,`ap-southeast-2`,`aws-global`,`ca-central-1`,`eu-central-1`,`eu-north-1`,`eu-west-1`,`eu-west-2`,`eu-west-3`,`sa-east-1`,`us-east-1`,`us-east-2`,`us-west-1`, y`us-west-2`. Todas las demás regiones utilizan automáticamente su punto de conexión regional respectivo.
- **`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
```

Compatibilidad con los AWS SDK

Los siguientes SDK admiten las características y los ajustes descritos en este tema. Se anotan todas las excepciones parciales. Todos los ajustes de propiedades del sistema JVM son compatibles con AWS SDK for Java y únicamente. AWS SDK para Kotlin

SDK	Ce	Notas o más información
AWS CLI v2	Parci	El valor predeterminado es `regional`.
SDK para C++	Parci	No se admite la configuración de archivos y variables de entorno de `config`. El SDK funciona con la configuración de `regional`.
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	No	
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	
Herramientas para PowerShell	Sí	

Puntos de conexión de doble pila y FIPS

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 en 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 conexión de doble pila, que admiten tráfico de IPv4 e IPv6, consulte Uso de puntos de conexión 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 en 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 que cumple con la norma FIPS 140-2. Si esta configuración está habilitada y no existe un punto final FIPS para su servicio, es posible que se produzca un error en la llamada. Región de AWS 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 conexión del servicio Amazon Elastic Compute Cloud, consulte los Puntos de conexión de doble pila (IPv4 e 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.

Compatibilidad con los SDK AWS

Los siguientes SDK admiten las características y los ajustes descritos en este tema. Se anotan todas las excepciones parciales. Todos los ajustes de propiedades del sistema JVM son compatibles con AWS SDK for Java y únicamente. AWS SDK para Kotlin

SDK	Compatible	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	Ce	Notas o más información
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	
Herramientas para PowerShell	Sí	

Detección de puntos de conexión

Los SDK utilizan la detección de puntos finales para acceder a los puntos finales del servicio (URL para acceder a varios recursos) y, al mismo tiempo, mantienen la flexibilidad para modificar las URL según AWS 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 de `DescribeEndpoints` para recuperar los puntos de conexión disponibles y, a continuación, los utiliza para completar operaciones específicas, como `CreateDatabase` o `CreateTable`.

La detección de puntos de conexión es obligatoria en algunos servicios y opcional en otros. El valor predeterminado es `true` o `false`, depende de si el servicio requiere la detección de puntos de conexión. Por ejemplo, Timestream tiene el valor predeterminado de `true` y Amazon DynamoDB el valor predeterminado de `false`. En el caso de los servicios en los que no se requiere la detección de puntos de conexión, la detección de puntos de conexión no está habilitada. En cambio, las opciones de configuración están disponibles a través de las variables de entorno, el AWS `config` archivo compartido o las estructuras del código del SDK (por ejemplo, las clases de configuración). En el caso de las operaciones en las que se requiere la detección de puntos de conexión, el SDK intenta detectarlos automáticamente.

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 en Java/Kotlin, Para configurar el valor directamente en el código, consulte directamente su SDK específico.

Habilita o deshabilita la detección de puntos de conexión para los servicios en los que la detección de puntos de conexión es opcional. La detección de puntos de conexión es obligatoria en algunos servicios.

Valor predeterminado: `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.

AWS Compatibilidad con los SDK

Los siguientes SDK admiten las características y los ajustes descritos en este tema. Se anotan todas las excepciones parciales. Todos los ajustes de propiedades del sistema JVM son compatibles con AWS SDK for Java y únicamente. AWS SDK para Kotlin

SDK	Ce	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	C e	Notas o más información
SDK para Java 2.x	Sí	El SDK for Java 2.x utiliza el nombre `AWS_ENDPO INT_DISCOVERY_ENABLED` de la variable de entorno.
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 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.
Herramientas para PowerShell	Sí	

Ajustes de configuración general

Los SDK 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 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 `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 extensión `.pem`) que debe utilizarse al establecer conexiones SSL/TLS.

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
```

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

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

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.

Compatibilidad con los SDK AWS

Los siguientes SDK admiten las características y los ajustes descritos en este tema. Se anotan todas las excepciones parciales. Todos los ajustes de propiedades del sistema JVM son compatibles con AWS SDK for Java y únicamente. AWS SDK para Kotlin

SDK	Ce	Notas o más información
AWS CLI v2	Parci	`api_versions` no admitido.
SDK para C++	Sí	
SDK para Go V2 (1.x)	Parci	`api_versions` y `parameter_validation` no son compatibles.
SDK para Go 1.x (V1)	Parci	`api_versions` y `parameter_validation` no son compatibles. Para usar la configuración de archivos compartid os`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 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	C e	Notas o más información
Herramientas para PowerShell	No	

Cliente IMDS

Los SDK implementan un cliente Instance Metadata Service versión 2 (IMDSv2) mediante solicitudes orientadas a la sesión. Para obtener más información sobre IMDSv2, consulte Uso de IMDSv2 en la Guía del usuario de Amazon EC2 para instancias de Linux o Uso de IMDSv2 en la Guía del usuario de Amazon EC2 para instancias de Windows. 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 cero.

`port` - miembro del objeto de configuración del cliente

El puerto del punto de conexión.

Valor predeterminado: 80

Valores válidos: 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: Número.

`endpoint` - miembro del objeto de configuración del cliente

El tipo de punto de conexión.

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

Valores válidos: URI válido.

Las siguientes opciones son compatibles con la mayoría de los SDK. 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 cero.

`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 cero.

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

Establece un flujo de salida para la depuración.

Valor predeterminado: ninguno.

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

Compatibilidad con los SDK AWS

Los siguientes SDK admiten las características y los ajustes descritos en este tema. Se anotan todas las excepciones parciales. Todos los ajustes de propiedades del sistema JVM son compatibles con AWS SDK for Java y únicamente. AWS SDK para Kotlin

SDK	Ce	Notas o más información
AWS CLI v2	Sí	
SDK para C++	No	IMDSv2 se usa solo internamente. Consulte Proveedor de credenciales IMDS.
SDK para Go V2 (1.x)	Sí	
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	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	C e	Notas o más información
SDK para Rust	Sí	
Herramientas para PowerShell	Sí	

Comportamiento de los reintentos

El comportamiento de reintento incluye la configuración relativa a la forma en que los SDK intentan recuperarse de los errores derivados de las solicitudes realizadas a Servicios de AWS.

Configure esta funcionalidad mediante lo siguiente:

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

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

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

- Si `retry_mode` 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 `retry_mode` es `standard`: realiza tres intentos.
- Si `retry_mode` es `adaptive`: realiza tres intentos.

Valores válidos: un número mayor que cero.

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

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

Valor predeterminado: `legacy` es la estrategia de reintentos predeterminada.

Valores válidos:

- `legacy` – Específico para su SDK (consulte su guía de SDK específica o la base de código de su SDK).

- `standard` – El conjunto estándar de reglas de reintento en todos los AWS SDK. Este modo incluye un conjunto estándar de errores que se reintentan y admite cuotas de reintentos. El número máximo predeterminado de intentos con este modo es tres, a menos que `max_attempts` se configure de forma explícita.
- `adaptive` – Un modo de reintento experimental que incluye la funcionalidad del modo estándar, pero incluye una limitación automática del lado del cliente. Como este modo es experimental, podría cambiar su comportamiento en el futuro.

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`:

Los token buckets solo se utilizan en el modo de reintento `adaptive`. Los token buckets imponen una tasa máxima de solicitudes al requerir que haya un token disponible para poder iniciar una solicitud. El cliente del SDK se puede configurar para rechazar rápidamente la solicitud o bloquearla hasta que haya un token disponible.

La limitación de velocidad por parte del cliente es un algoritmo que, en un principio, permite realizar solicitudes a cualquier velocidad, hasta el límite de token permitido. Sin embargo, cuando se detecta

una respuesta limitada, el cliente queda limitado en consecuencia. rate-of-request La asignación de fichas también se incrementa en consecuencia si se reciben respuestas satisfactorias.

Con la limitación de velocidad adaptativa, los SDK pueden reducir la velocidad a la que se envían las solicitudes para adaptarse mejor a la capacidad de. Servicios de AWS

`SendHTTPRequest`:

La mayoría de AWS los SDK utilizan una biblioteca HTTP que utiliza grupos de conexiones para que puedas reutilizar una conexión existente al realizar una solicitud HTTP. Por lo general, las conexiones se reutilizan al reintentar las solicitudes debido a errores de limitación. Las solicitudes no se reutilizan al volver a intentarlo debido a errores transitorios.

`RequestBookkeeping`:

La cuota de reintentos debe actualizarse si la solicitud se realiza correctamente. Solo en el modo de reintento `adaptive`, la variable de estado `maxsendrate` 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`:

Se especifica mediante la configuración del archivo `config` o de la variable de entorno.

`HasRetryQuota`

Este paso limita las solicitudes de reintentos de limitaciones al requerir que haya un token disponible en el bucket de cuotas de reintentos. Los buckets de cuotas de reintentos son un mecanismo para

evitar reintentos que no tienen probabilidades de éxito. Estas cuotas dependen del SDK, suelen depender del cliente y, a veces, incluso de los puntos de conexión del servicio. Los tokens de cuota de reintentos disponibles se eliminan cuando las solicitudes fallan por varios motivos y se reponen cuando se cumplen correctamente. Si no queda ningún token, se cierra el ciclo de reintentos.

`ExponentialBackoff`

En el caso de un error que se pueda volver a intentar, el retraso del reintento se calcula mediante un retroceso exponencial truncado. Los SDK utilizan un retroceso exponencial binario truncado con fluctuación de fase. El siguiente algoritmo muestra cómo se define la cantidad de tiempo de reposo, en segundos, para una respuesta a una solicitud: `i`

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

En el algoritmo anterior, se aplican los siguientes valores:

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

`r = 2`

`MAX_BACKOFF = 20 seconds` para la mayoría de los SDK. Consulte la guía o el código fuente específicos del SDK para confirmarlo.

Compatibilidad con los SDK AWS

Los siguientes SDK admiten las características y los ajustes descritos en este tema. Se anotan todas las excepciones parciales. Todos los ajustes de propiedades del sistema JVM son compatibles con AWS SDK for Java y únicamente. AWS SDK para Kotlin

SDK	C 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	Ce	Notas o más información
SDK para Java 1.x	Sí	
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 para Kotlin	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	
Herramientas para PowerShell	Sí	

Compresión de solicitudes

AWS Los SDK y las herramientas pueden comprimir automáticamente las cargas útiles al enviar solicitudes a las Servicios de AWS que admiten la recepción de cargas útiles comprimidas. Comprimir la carga útil en el cliente antes de enviarla a un servicio puede reducir el número total de solicitudes y el ancho de banda necesario para enviar datos al servicio, así como reducir las solicitudes que se realizan incorrectamente debido a las limitaciones del servicio en cuanto al tamaño de la carga útil. Para la compresión, 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 en 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 archivos compartidos AWS **`config`**, **`AWS_REQUEST_MIN_COMPRESSION_SIZE_BYTES`** - variable de entorno, **`aws.requestMinCompressionSizeBytes`**- Propiedad del sistema JVM: solo en 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.

AWS Compatibilidad con los SDK

Los siguientes SDK admiten las características y los ajustes descritos en este tema. Se anotan todas las excepciones parciales. Todos los ajustes de propiedades del sistema JVM son compatibles con AWS SDK for Java y únicamente. AWS SDK para Kotlin

SDK	Ce	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 3.x	Sí	
SDK para PHP 3.x	Sí	
SDK para Python (Boto3)	Sí	
SDK para Ruby 3.x	Sí	
SDK para Rust	No	
Herramientas para PowerShell	Sí	

Puntos de conexión específicos del servicio

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 en Java/Kotlin

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

También puedes usar esta configuración dentro de una `services` sección de un 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 de `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 en 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 en Java/Kotlin

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

Tenga en cuenta que cualquier punto de conexión explícito establecido en el código o en el propio cliente de servicio se utiliza independientemente de esta configuración. Por ejemplo, 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 dentro de un `profile`, ese punto de conexión se convierte en el punto de conexión global.
- `endpoint_url` anidado bajo una clave identificadora de servicio dentro de 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 las URL de punto final específicas del servicio para Amazon S3 y AWS Elastic Beanstalk en el mismo perfil. AWS Elastic Beanstalk tiene un número de. `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.

Compatibilidad con los AWS SDK

Los siguientes SDK admiten las características y los ajustes descritos en este tema. Se anotan todas las excepciones parciales. Todos los ajustes de propiedades del sistema JVM son compatibles con AWS SDK for Java y únicamente. AWS SDK para Kotlin

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

SDK	C e	Notas o más información
Herramientas para PowerShell	Sí	

Identificadores de punto de conexión específicos del servicio

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

serviceId	Cl id ac de se pa el ar cc o AV **cc**	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
AccessAnalyzer	ac ly	AWS_ENDPOINT_URL_ACCESSANALYZER
Account	ac	AWS_ENDPOINT_URL_ACCOUNT
ACM	ac	AWS_ENDPOINT_URL_ACM
ACM PCA	ac	AWS_ENDPOINT_URL_ACM_PCA
Alexa For Business	a: _I	AWS_ENDPOINT_URL_ALEXA_FOR_BUSINESS

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

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

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

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

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

serviceId	Cl id ac de se pa el ar co o A\ **co**	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
Chime SDK Messaging	ch _r g	AWS_ENDPOINT_URL_CHIME_SDK_MESSAGING
Chime SDK Voice	ch _\	AWS_ENDPOINT_URL_CHIME_SDK_VOICE
CleanRooms	cl s	AWS_ENDPOINT_URL_CLEANROOMS
CleanRoomsML	cl sr	AWS_ENDPOINT_URL_CLEANROOMSML
Cloud9	cl	AWS_ENDPOINT_URL_CLOUD9
CloudControl	cl rc	AWS_ENDPOINT_URL_CLOUDCONTROL
CloudDirectory	cl cf	AWS_ENDPOINT_URL_CLOUDDIRECTORY
CloudFormation	cl at	AWS_ENDPOINT_URL_CLOUDFORMATION

serviceId	Cl id ac de se pa el ar cc o A\ **cc**	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
CloudFront	c t	AWS_ENDPOINT_URL_CLOUDFRONT
CloudFront KeyValueStore	c t_ e:	AWS_ENDPOINT_URL_CLOUDFRONT_KEYVALUESTORE
CloudHSM	c	AWS_ENDPOINT_URL_CLOUDHSM
CloudHSM V2	c v2	AWS_ENDPOINT_URL_CLOUDHSM_V2
CloudSearch	c cl	AWS_ENDPOINT_URL_CLOUDSEARCH
CloudSearch Domain	c cl	AWS_ENDPOINT_URL_CLOUDSEARCH_DOMAIN
CloudTrail	c l	AWS_ENDPOINT_URL_CLOUDTRAIL
CloudTrail Data	c l_	AWS_ENDPOINT_URL_CLOUDTRAIL_DATA

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

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

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

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

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

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

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

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

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

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

serviceId	Cl id ac de se pa el ar cc o A\ **cc**	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
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	ir de	AWS_ENDPOINT_URL_IMAGEBUILDER
ImportExport	ir o:	AWS_ENDPOINT_URL_IMPORTEXPORT

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

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

serviceId	Cl id ac de se pa el ar cc o AV **co**	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
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
KafkaConnect	ka ec	AWS_ENDPOINT_URL_KAFKACONNECT
kendra	ke	AWS_ENDPOINT_URL_KENDRA

serviceId	Cl id ac de se pa el ar cc o AV **co**	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
Kendra Ranking	ke nl	AWS_ENDPOINT_URL_KENDRA_RANKING
Keyspaces	ke	AWS_ENDPOINT_URL_KEYSPACES
Kinesis	ki	AWS_ENDPOINT_URL_KINESIS
Kinesis Video Archived Media	ki id iv a	AWS_ENDPOINT_URL_KINESIS_VIDEO_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
Kinesis Video WebRTC Storage	ki id to e	AWS_ENDPOINT_URL_KINESIS_VIDEO_WEBRTC_STORAGE

serviceId	Cl id ac de se pa el ar cc o A\ **cc**	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
Kinesis Analytics	k: na	AWS_ENDPOINT_URL_KINESIS_ANALYTICS
Kinesis Analytics V2	k: na v2	AWS_ENDPOINT_URL_KINESIS_ANALYTICS_V2
Kinesis Video	k: 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
Lex Model Building Service	le _l _s	AWS_ENDPOINT_URL_LEX_MODEL_BUILDING_ SERVICE

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

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

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

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

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

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

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

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

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

serviceId	Cl id ac de se pa el ar cc o AV **co**	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
Pipes	p:	AWS_ENDPOINT_URL_PIPES
Polly	po	AWS_ENDPOINT_URL_POLLY
Pricing	p:	AWS_ENDPOINT_URL_PRICING
PrivateNetworks	p: tv	AWS_ENDPOINT_URL_PRIVATENETWORKS
Proton	p:	AWS_ENDPOINT_URL_PROTON
QBusiness	qb	AWS_ENDPOINT_URL_QBUSINESS
QConnect	qc	AWS_ENDPOINT_URL_QCONNECT
QLDB	q:	AWS_ENDPOINT_URL_QLDB
QLDB Session	q: ic	AWS_ENDPOINT_URL_QLDB_SESSION
QuickSight	qu t	AWS_ENDPOINT_URL_QUICKSIGHT
RAM	ra	AWS_ENDPOINT_URL_RAM

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

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

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

serviceId	Cl id ac de se pa el ar co o AV **co**	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
SageMaker	sa	AWS_ENDPOINT_URL_SAGEMAKER
SageMaker A2I Runtime	sa _a ir	AWS_ENDPOINT_URL_SAGEMAKER_A2I_RUNTIME
Sagemaker Edge	sa _e	AWS_ENDPOINT_URL_SAGEMAKER_EDGE
SageMaker FeatureSt ore Runtime	sa _f to ir	AWS_ENDPOINT_URL_SAGEMAKER_FEATUREST ORE_RUNTIME
SageMaker Geospatial	sa _g ai	AWS_ENDPOINT_URL_SAGEMAKER_GEOSPATIAL
SageMaker Metrics	sa _m	AWS_ENDPOINT_URL_SAGEMAKER_METRICS
SageMaker Runtime	sa _r	AWS_ENDPOINT_URL_SAGEMAKER_RUNTIME

serviceId	Cl id ac de se pa el ar cc o A\ **cc**	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
savingsplans	sa ar	AWS_ENDPOINT_URL_SAVINGSPLANS
Scheduler	sc	AWS_ENDPOINT_URL_SCHEDULER
schemas	sc	AWS_ENDPOINT_URL_SCHEMAS
SimpleDB	sd	AWS_ENDPOINT_URL_SIMPLEDB
Secrets Manager	se ar	AWS_ENDPOINT_URL_SECRETS_MANAGER
SecurityHub	se ub	AWS_ENDPOINT_URL_SECURITYHUB
SecurityLake	se al	AWS_ENDPOINT_URL_SECURITYLAKE
ServerlessApplicationRepository	se sa ic to	AWS_ENDPOINT_URL_SERVERLESSAPPLICATIONREPOSITORY

serviceId	Cl id ac de se pa el ar cc o A\ **cc**	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
Service Quotas	se uc	AWS_ENDPOINT_URL_SERVICE_QUOTAS
Service Catalog	se at	AWS_ENDPOINT_URL_SERVICE_CATALOG
Service Catalog AppRegistry	se at p:	AWS_ENDPOINT_URL_SERVICE_CATALOG_APP REGISTRY
ServiceDiscovery	se sc	AWS_ENDPOINT_URL_SERVICEDISCOVERY
SES	se	AWS_ENDPOINT_URL_SES
SESv2	se	AWS_ENDPOINT_URL_SESV2
Shield	sh	AWS_ENDPOINT_URL_SHIELD
signer	si	AWS_ENDPOINT_URL_SIGNER
SimSpaceWeaver	si ea	AWS_ENDPOINT_URL_SIMSPACEWEAVER
SMS	sn	AWS_ENDPOINT_URL_SMS

serviceId	Cl id ac de se pa el ar cc o AV **co**	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
Snow Device Managemen t	sr ce me	AWS_ENDPOINT_URL_SNOW_DEVICE_MANAGEMENT
Snowball	sr	AWS_ENDPOINT_URL_SNOWBALL
SNS	sr	AWS_ENDPOINT_URL_SNS
SQS	sc	AWS_ENDPOINT_URL_SQS
SSM	ss	AWS_ENDPOINT_URL_SSM
SSM Contacts	ss ct	AWS_ENDPOINT_URL_SSM_CONTACTS
SSM Incidents	ss er	AWS_ENDPOINT_URL_SSM_INCIDENTS
Ssm Sap	ss	AWS_ENDPOINT_URL_SSM_SAP
SSO	ss	AWS_ENDPOINT_URL_SSO
SSO Admin	ss	AWS_ENDPOINT_URL_SSO_ADMIN
SSO OIDC	ss	AWS_ENDPOINT_URL_SSO_OIDC

serviceId	Cl id ac de se pa el ar cc o A\ **cc**	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
SFN	st	AWS_ENDPOINT_URL_SFN
Storage Gateway	st at	AWS_ENDPOINT_URL_STORAGE_GATEWAY
STS	st	AWS_ENDPOINT_URL_STS
SupplyChain	su in	AWS_ENDPOINT_URL_SUPPLYCHAIN
Support	su	AWS_ENDPOINT_URL_SUPPORT
Support App	su pp	AWS_ENDPOINT_URL_SUPPORT_APP
SWF	sv	AWS_ENDPOINT_URL_SWF
synthetics	sy s	AWS_ENDPOINT_URL_SYNTHETICS
Textract	te	AWS_ENDPOINT_URL_TEXTRACT
Timestream InfluxDB	ti m_ b	AWS_ENDPOINT_URL_TIMESTREAM_INFLUXDB

serviceId	Cl id ac de se pa el ar cc o AV **co**	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
Timestream Query	t: m_	AWS_ENDPOINT_URL_TIMESTREAM_QUERY
Timestream Write	t: m_	AWS_ENDPOINT_URL_TIMESTREAM_WRITE
tnb	tr	AWS_ENDPOINT_URL_TNB
Transcribe	t: e	AWS_ENDPOINT_URL_TRANSCRIBE
Transfer	t:	AWS_ENDPOINT_URL_TRANSFER
Translate	t:	AWS_ENDPOINT_URL_TRANSLATE
TrustedAdvisor	t: v:	AWS_ENDPOINT_URL_TRUSTEDADVISOR
VerifiedPermissions	ve e: s	AWS_ENDPOINT_URL_VERIFIEDPERMISSIONS
Voice ID	vo	AWS_ENDPOINT_URL_VOICE_ID

serviceId	Cl id ac de se pa el ar co o AV **co**	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
VPC Lattice	vp ce	AWS_ENDPOINT_URL_VPC_LATTICE
WAF	wa	AWS_ENDPOINT_URL_WAF
WAF Regional	wa na	AWS_ENDPOINT_URL_WAF_REGIONAL
WAFV2	wa	AWS_ENDPOINT_URL_WAFV2
WellArchitected	we te	AWS_ENDPOINT_URL_WELLARCHITECTED
Wisdom	wi	AWS_ENDPOINT_URL_WISDOM
WorkDocs	wo	AWS_ENDPOINT_URL_WORKDOCS
WorkLink	wo	AWS_ENDPOINT_URL_WORKLINK
WorkMail	wo	AWS_ENDPOINT_URL_WORKMAIL
WorkMailMessageFlow	wo es w	AWS_ENDPOINT_URL_WORKMAILMESSAGEFLOW

serviceId	Cl id ac de se pa el ar cc o A\ **cc**	**AWS_ENDPOINT_URL_<SERVICE>** variable de entorno
WorkSpaces	wc s	AWS_ENDPOINT_URL_WORKSPACES
WorkSpaces Thin Client	wc s_ ie	AWS_ENDPOINT_URL_WORKSPACES_THIN_CLIENT
WorkSpaces Web	wc s_	AWS_ENDPOINT_URL_WORKSPACES_WEB
XRay	x:	AWS_ENDPOINT_URL_XRAY

Valores predeterminados de configuración inteligente

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

Configure esta funcionalidad mediante lo siguiente:

defaults_mode- configuración de archivos compartidos AWS **config**, **AWS_DEFAULTS_MODE** - variable de entorno, **aws.defaultsMode**- Propiedad del sistema JVM: solo en Java/Kotlin

> Con esta configuración, puede elegir un modo que se alinee con la arquitectura de la aplicación y, a continuación, proporcionar valores predeterminados optimizados para la aplicación. Si 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 los SDK 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 del usuario, lo que puede introducir latencia. Si la startup es fundamental para tu aplicación, te recomendamos que elijas una `defaults_mode` explícita 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 regionalizados.
- `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`

Parámetro	modo **standard**	modo **in-region**	modo **cross-region**	modo **mobile**
`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 de `standard` se asignará para `retry_mode` (de las opciones `retry_mode` válidas) y el valor de `regional` se asignará para `stsRegionalEndpoints` (de las opciones `stsRegionalEndpoints` válidas).

Compatibilidad con los SDK AWS

Los siguientes SDK admiten las características y los ajustes descritos en este tema. Se anotan todas las excepciones parciales. Todos los ajustes de propiedades del sistema JVM son compatibles con AWS SDK for Java y únicamente. AWS SDK para Kotlin

SDK	Compatible	Notas o más información
AWS CLI v2	No	
SDK para C++	Sí	Parámetros no optimizados: `stsRegionalEndpoints` , `s3UsEast1RegionalEndpoints` , `tlsNegoti`

SDK	Compatible	Notas o más información
		`ationTimeoutInMill is` .
SDK para Go V2 (1.x)	Sí	Parámetros no optimizados: `retryMode` , `stsRegion alEndpoints` , `s3UsEast1RegionalE ndpoints` .
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` , `s3UsEast1RegionalE ndpoints` , `tlsNegoti ationTimeoutInMill is` . `connectTi meoutInMillis` se llama `connectionTimeout` .
SDK para 2.x JavaScript	No	
SDK para Kotlin	No	
SDK para .NET 3.x	Sí	Parámetros no optimizados: `connectTimeoutInMi llis` , `tlsNegoti ationTimeoutInMill is` .

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

Bibliotecas de Common Runtime (CRT) AWS

Las bibliotecas Common Runtime (CRT) de AWS son una biblioteca base de los SDK. El CRT es una familia modular de paquetes independientes, escrita en C. Cada paquete ofrece un buen rendimiento y ocupa un espacio mínimo para las diferentes funcionalidades requeridas. Estas funcionalidades son comunes y se comparten en todos los SDK, lo que proporciona una mejor reutilización, optimización y precisión del código. Los paquetes son:

- `awslabs/aws-c-auth`: autenticación de AWS del lado del cliente (proveedores de credenciales estándar y firma (sigv4))
- `awslabs/aws-c-cal`: tipos primitivos criptográficos, hashes (MD5, SHA256, SHA256 HMAC), firmantes, AES
- `awslabs/aws-c-common`: estructuras de datos básicas, tipos primitivos de subproceso/sincronización, administración de búferes, funciones relacionadas con stdlib
- `awslabs/aws-c-compression`: algoritmos de compresión (codificación/decodificación de Huffman)
- `awslabs/aws-c-event-stream`: procesamiento de mensajes de flujo de eventos (encabezados, preludio, carga útil, crc/trailer), implementación de llamadas a procedimientos remotos (RPC) sobre transmisiones de eventos
- `awslabs/aws-c-http`: implementación de las especificaciones de HTTP/1.1 y de HTTP/2 en C99
- `awslabs/aws-c-io`: sockets (TCP, UDP), DNS, canalizaciones, bucles de eventos, canales, SSL/TLS
- `awslabs/aws-c-iot`: implementación C99 de la integración de servicios AWS de IoT en la nube con dispositivos
- `awslabs/aws-c-mqtt`: protocolo de mensajería ligero y estándar para Internet de las cosas (IoT)
- `awslabs/aws-c-s3`: implementación de la biblioteca C99 para comunicarse con el servicio Amazon S3, diseñada para maximizar el rendimiento en las instancias Amazon EC2 de gran ancho de banda
- `awslabs/aws-c-sdkutils`: una biblioteca de utilidades para analizar y administrar perfiles de AWS
- `awslabs/aws-checksums`: CRC32c y CRC32 multiplataforma acelerados por hardware, que recurren a implementaciones de software eficientes

- `awslabs/aws-lc`: biblioteca criptográfica de uso general mantenida por el equipo de criptografía de AWS para AWS y sus clientes, basada en el código del proyecto Google BoringSSL y el proyecto OpenSSL
- `awslabs/s2n`: implementación C99 de los protocolos TLS/SSL, diseñada para ser pequeña y rápida, con la seguridad como prioridad

El CRT está disponible en todos los SDK excepto en Go.

Dependencias de CRT

Las bibliotecas CRT forman una red compleja de relaciones y dependencias. Conocer estas relaciones es útil si necesita crear el CRT directamente desde la fuente. Sin embargo, la mayoría de los usuarios acceden a la funcionalidad CRT a través del SDK de su idioma (como el SDK de AWS para C++ o el SDK de AWS para Java) o el SDK para dispositivos IoT de su idioma (como el SDK de AWS IoT para C++ o el SDK de AWS IoT para Java). En el siguiente diagrama, el recuadro de enlaces CRT de idiomas hace referencia al paquete que contiene las bibliotecas CRT de un SDK de lenguaje específico. Se trata de una colección de paquetes con este formato `aws-crt-*`, donde "*" es un lenguaje del SDK (como `aws-crt-cpp` o `aws-crt-java`).

La siguiente es una ilustración de las dependencias jerárquicas de las bibliotecas CRT.

Mantenimiento y soporte

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

En los siguientes temas se tratan las políticas de mantenimiento y soporte de versiones de los SDK de AWS.

Temas

- Política de mantenimiento de SDK y herramientas AWS
- AWS Matriz de soporte de versiones de SDK y herramientas
- Kits de herramientas IDE

Política de mantenimiento de SDK y herramientas AWS

Información general

Este documento describe la política de mantenimiento de las herramientas y kits de desarrollo de software (SDK) de AWS, incluidos los SDK móviles y de IoT, y sus dependencias subyacentes. AWS proporciona periódicamente a los SDK y a las herramientas de AWS actualizaciones que pueden incluir compatibilidad con API nuevas o actualizadas, nuevas funciones, mejoras, correcciones de errores, parches de seguridad o actualizaciones de la documentación de AWS. 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 SDK de AWS tienen el formato X.Y.Z, donde X representa la versión principal. El aumento de la versión principal de un SDK indica que este ha tenido cambios considerables y sustanciales para admitir nuevos modismos y patrones en el idioma. Las versiones

principales se introducen cuando las interfaces públicas (como las clases, métodos, tipos, etc.), los comportamientos o la semántica cambian. Las aplicaciones deben actualizarse para que funcionen con la versión más reciente del SDK. Es importante actualizar las versiones principales con cuidado y de acuerdo con las pautas de actualización proporcionadas por AWS.

Ciclo de vida de las versiones principales del

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

- Vista previa para desarrolladores (fase 0): durante esta fase, los SDK no son compatibles, no deben usarse en entornos de producción y están pensados únicamente para facilitar el acceso anticipado y para enviar comentarios. Es posible que en futuras versiones se introduzcan cambios importantes. Una vez que AWS identifique una versión como un producto estable, puede marcarla como versión candidata a ser lanzada. Las versiones candidatas a ser lanzadas están listas para su publicación en GA, a menos que surjan errores importantes, y recibirán soporte técnico completo de AWS.
- Disponibilidad general (GA) (fase 1): durante esta fase, los SDK son totalmente compatibles. AWS proporcionará versiones periódicas del SDK que incluyen soporte para nuevos servicios, actualizaciones de API para los servicios existentes y correcciones de errores y de seguridad. En el caso de Herramientas, AWS publicará versiones periódicas que incluyen nuevas actualizaciones de características 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 publicará un anuncio público al menos 6 meses antes de que el SDK entre en modo de mantenimiento. Durante este período, el SDK seguirá siendo totalmente compatible. Por lo general, el modo de mantenimiento se anuncia al mismo tiempo que la siguiente versión principal pasa a GA.
- Mantenimiento (fase 3): durante el modo de mantenimiento, AWS limita las versiones del SDK para abordar únicamente las correcciones de errores críticos y los problemas de seguridad. Un SDK no recibirá actualizaciones de API para servicios nuevos o existentes, ni se actualizará para que sea compatible con nuevas regiones. El modo de mantenimiento tiene una duración predeterminada de 12 meses, a menos que se especifique lo contrario.
- Fin del soporte (fase 4): cuando un SDK llega al final del soporte, ya no recibirá actualizaciones ni versiones. Las versiones publicadas anteriormente seguirán estando disponibles a través de los gestores de paquetes públicos y el código permanecerá activo. GitHub El GitHub repositorio puede estar archivado. El uso de un SDK disponible end-of-support queda a discreción del usuario. Recomendamos a los usuarios que actualicen a la nueva versión principal.

La siguiente es una ilustración visual del ciclo de vida de la versión principal del SDK. Tenga en cuenta que los plazos que se muestran a continuación son ilustrativos y no vinculantes.

Ciclo de vida de

La mayoría de los SDK de AWS tienen dependencias subyacentes, como los tiempos de ejecución de los idiomas, los sistemas operativos o las bibliotecas y marcos de terceros. Estas dependencias suelen estar vinculadas a la comunidad lingüística o al proveedor propietario de ese componente en particular. Cada comunidad o proveedor publica su propio end-of-support cronograma para su producto.

Los siguientes términos se utilizan para clasificar las dependencias subyacentes de terceros:

- Sistema operativo (SO): algunos ejemplos incluyen Amazon Linux AMI, Amazon Linux 2, Windows 2008, Windows 2012, Windows 2016, etc.
- Lenguaje del tiempo de ejecución: algunos ejemplos son Java 7, Java 8, Java 11, .NET Core, .NET Standard, .NET PCL, etc.
- Biblioteca/Marco de trabajo de terceros: algunos ejemplos incluyen OpenSSL, .NET Framework 4.5, Java EE, etc.

Nuestra política consiste en seguir dando soporte a las dependencias del SDK durante al menos 6 meses después de que la comunidad o el proveedor hayan dejado de dar soporte a la dependencia. Sin embargo, esta política puede variar en función de la dependencia específica.

> **Note**
>
> AWS se reserva el derecho de dejar de dar soporte a 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 end-of-support, especificará los plazos de la campaña y proporcionará una guía de actualización.

- AWSLa documentación del SDK, como la documentación de referencia de la API, las guías de usuario, las páginas de marketing de los productos del SDK y los GitHub archivos readme (s), se actualiza para indicar el calendario de la campaña y proporcionar orientación sobre la actualización de las aplicaciones afectadas.
- Se publica una entrada de AWS blog en la que se describe el camino a seguir end-of-support y se reiteran los plazos de la campaña.
- Se añaden advertencias de obsolescencia a los SDK, en las que se describe la ruta de acceso a la documentación del SDK end-of-support y se enlaza con ella.

Para ver la lista de las principales versiones disponibles de los AWS SDK y las herramientas y en qué punto del ciclo de vida de mantenimiento se encuentran, consulte. the section called “Matriz de compatibilidad con versiones”

AWS Matriz de soporte de versiones de SDK y herramientas

En la siguiente matriz 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 principales versiones de AWS los SDK y las herramientas y sus dependencias subyacentes, consulte. the section called “Política de mantenimiento”

SDK	Versión principal	Fase actual	Fecha de disponibilidad general	Notas
AWS CLI	1.x	Disponibilidad general	2 de septiembre de 2013	
AWS CLI	2.x	Disponibilidad general	2/10/2020	
SDK para C++	1.x	Disponibilidad general	2/9/2015	
SDK para Go V2	V2 1.x	Disponibilidad general	19/1/2021	

SDK	Versión principal	Fase actual	Fecha de disponibilidad general	Notas
SDK para Go	1.x	Anuncio de mantenimiento	19/11/2015	Consulte el anuncio para conocer los detalles y las fechas
SDK para Java	1.x	Anuncio de mantenimiento	25 de marzo de 2010	Consulte el anuncio para conocer los detalles y las fechas
SDK para Java	2.x	Disponibilidad general	20/11/2018	
SDK para JavaScript	1.x	Fin-del-soporte	6/05/2013	
SDK para JavaScript	2.x	Anuncio de mantenimiento	19/06/2014	Consulte el anuncio para conocer los detalles y las fechas
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	Versión principal	Fase actual	Fecha de disponibilidad general	Notas
SDK para .NET	3.x	Disponibilidad general	28/7/2015	
SDK para PHP	2.x	Fin-del-soporte	2/11/2012	
SDK para PHP	3.x	Disponibilidad general	27/05/2015	
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 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	Vista previa para desarrolladores		
Herramientas para PowerShell	2.x	Fin-del-soporte	8/11/2013	
Herramientas para PowerShell	3.x	Fin-del-soporte	29/7/2015	

SDK	Versión principal	Fase actual	Fecha de disponibilidad general	Notas
Herramientas para PowerShell	4.x	Disponibilidad general	21/11/2019	

Kits de herramientas IDE

AWSLos kits de herramientas del entorno de desarrollo integrado (IDE) son complementos y extensiones que permiten el acceso a los servicios de AWS desde 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:

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

Las siguientes secciones contienen información de soporte, informes de mantenimiento y notificaciones sobre los kits de herramientas del IDE de AWS.

Notificación de telemetría

Los kits de herramientas IDE de AWS pueden recopilar y almacenar datos de telemetría del lado del cliente para informar las decisiones sobre futuras versiones del kit de herramientas de AWS. Los datos recopilados cuantifican el uso que usted hace del kit de herramientas de AWS.

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

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

- AWS Toolkit for Visual Studio

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

Algunos servicios de AWS a los que se puede acceder desde los kits de herramientas pueden recopilar datos de telemetría adicionales del lado del cliente, como Amazon CodeWhisperer. Para obtener información detallada sobre el tipo de datos que recopila CodeWhisperer o cómo optar por no utilizar la telemetría del lado del cliente para CodeWhisperer, consulte el tema Compartir sus datos con AWS en la Guía del usuario de Amazon CodeWhisperer.

Historial de documentos para la AWS Guía de referencia de SDK y herramientas

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

Cambio	Descripción	Fecha
Actualizaciones de configuraciones	Añada los ajustes de configuración del sistema JVM.	27 de marzo de 2024
Actualizaciones de la tabla de compatibilidad	Actualizaciones de la compatibilidad para el soporte del SDK y actualizaciones de los procedimientos del Centro de Identidad de IAM.	20 de febrero de 2024
Actualización de credenciales del contenedor. Actualización del IMDS.	Agregar soporte para Amazon EKS. Agregar configuración para desactivar el uso del IMDSv1 como 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 AWS Tools for PowerShell.	10 de diciembre de 2023
Actualizaciones de autenticación	Actualizaciones de los métodos de autenticación	1 de julio de 2023

	compatibles con SDK y herramientas.	
Actualizaciones de las prácticas recomendadas de IAM	Se ha actualizado la guía para implementar las prácticas recomendadas de IAM. Para obtener más información, consulte 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

Glosario de AWS

Para ver la terminología más reciente de AWS, consulte el Glosario de AWS en la Referencia de Glosario de AWS.

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la version original de inglés, prevalecerá la version en inglés.