



Guía del usuario de

AWS Kit de herramientas para VS Code



AWS Kit de herramientas para VS Code: Guía del usuario de

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Las marcas comerciales y la imagen comercial de Amazon no se pueden utilizar en relación con ningún producto o servicio que no sea de Amazon, de ninguna manera que pueda causar confusión entre los clientes y que menosprecie o desacredite a Amazon. Todas las demás marcas registradas que no son propiedad de Amazon son propiedad de sus respectivos propietarios, que pueden o no estar afiliados, conectados o patrocinados por Amazon.

Table of Contents

AWS Toolkit for Visual Studio Code	1
¿Qué es la AWS Toolkit for Visual Studio Code	1
Información relacionada	1
Amazon Q Developer y Amazon CodeWhisperer	2
Descargar el kit de herramientas	3
Descarga del kit de herramientas desde el Marketplace de VS Code	3
Kits de herramientas de IDE adicionales de AWS	3
Introducción	4
Instalación del kit de herramientas para VS Code	4
Requisitos previos	4
Descarga e instalación del AWS Toolkit for Visual Studio Code	4
Requisitos previos opcionales	5
Conectándose a AWS	6
Requisitos previos	6
Apertura del panel Inicio de sesión	7
Conectarse a AWS desde el kit de herramientas	7
Autenticación para Amazon CodeCatalyst	8
Cambio de las regiones de AWS	9
Adición de una región al Explorador de AWS	10
Ocultar una región en el Explorer de AWS	10
Configuración de la cadena de herramientas	10
Configurar una cadena de herramientas para .NET Core	10
Configurar una cadena de herramientas para Node.js	11
Configurar una cadena de herramientas para Python	11
Configurar una cadena de herramientas para Java	12
Configurar una cadena de herramientas para Go	12
Usar la cadena de herramientas	12
Autenticación y acceso	13
IAM Identity Center	13
Credenciales de IAM	13
Creación de un usuario de IAM	14
Crear un archivo de credenciales compartido a partir del AWS Toolkit for Visual Studio	
Code	15
Añadir perfiles de credenciales adicionales	16

ID de creador de AWS	17
Uso de un proceso de credenciales externo	17
Actualización de firewalls y puertas de enlace	17
AWS Toolkit for Visual Studio Code Puntos de enlace	17
Puntos de conexión del complemento de Amazon Q	18
Puntos de conexión de Amazon Q Developer	19
Puntos de conexión de la transformación de código de Amazon Q	19
Puntos de conexión de autenticación	19
Puntos de conexión de identidad	20
Telemetría	20
Referencias	21
Trabajando con AWS	22
Características experimentales	23
Explorador de AWS	23
Documentos de AWS	24
Introducción a AWS Documents	25
Visualización de documentación, autocompletado y validación en VS Code	25
Amazon CodeCatalyst	26
¿Qué es Amazon CodeCatalyst?	26
Introducción a CodeCatalyst	27
Usar recursos de CodeCatalyst	27
Uso de entornos de desarrollo	31
Resolución de problemas	34
Amazon API Gateway	36
AWS App Runner	36
Requisitos previos	37
Precios	40
Creación de servicios de App Runner	40
Administración de los servicios de App Runner	44
Desarrollador de aplicaciones de AWS	46
Trabajando con AWS Application Builder	47
AWS Infrastructure Composer	51
Uso de AWS Infrastructure Composer	51
AWS CDK	53
AWS CDK aplicaciones de	53
CloudFormation Pilas de	55

Eliminación de una pila de CloudFormation	56
Cree una CloudFormation plantilla	57
Amazon CloudWatch Logs	58
Visualización de grupos de registros y flujos de registro de CloudWatch	59
Uso de eventos de CloudWatch Logs	60
Buscar grupos de registro	62
Live Tail de Registros de CloudWatch	65
Amazon DocumentDB	67
Uso de Amazon DocumentDB	67
Amazon EC2	73
Trabajando con Amazon EC2	74
Solución de problemas de Amazon EC2	83
Amazon ECR	85
Uso de Amazon ECR	86
Creación de un servicio App Runner	96
Amazon ECS	98
Uso de IntelliSense para archivos de definición de tareas	98
Amazon ECS Exec	99
Amazon EventBridge	102
Trabajo con esquemas de Amazon EventBridge	102
AWS Analizador de acceso IAM	104
Trabajando con AWS IAM Access Analyzer	105
AWS IoT	109
AWS IoTRequisitos previos de	110
Objetos de AWS IoT	110
AWS IoTCertificados de	112
AWS IoTPolíticas de	115
Funciones de AWS Lambda	118
Trabajar con AWS Lambda funciones	119
AWS Lambda console a IDE	125
AWS Lambda LocalStack apoyo	127
Depuración remota de Lambda	132
Amazon Redshift	143
Uso de Amazon Redshift	143
Amazon S3	148
Trabajo con recursos de S3	149

Trabajo con objetos de S3	151
Estudio unificado de Amazon SageMaker	155
Aplicación sin servidor de AWS	155
Introducción	156
Uso de Serverless Land	164
Ejecución y depuración de funciones de Lambda directamente desde el código	166
Ejecución y depuración de recursos locales de Amazon API Gateway	171
Opciones de configuración para depurar aplicaciones sin servidor	175
Solución de problemas	184
AWS Systems Manager	186
Hipótesis y requisitos previos	186
Permisos de IAM para documentos de Systems Manager Automation	187
Creación de un nuevo documento de Automatización de Systems Manager	188
Apertura de un documento de automatización de Systems Manager existente	188
Edición de un documento de Automatización de Systems Manager	189
Publicación de un documento de Automatización de Systems Manager	190
Eliminación de un documento de Automatización de Systems Manager	190
Ejecución de un documento de Automatización de Systems Manager	191
Resolución de problemas	191
AWS Step Functions	192
Uso de Step Functions	192
Uso del Diseñador de flujo de trabajo	197
Threat Composer	201
Uso de Threat Composer	202
Recursos	203
Permisos de IAM para acceder a los recursos	204
Añadir recursos existentes e interactuar con ellos	204
Creación y edición de recursos	206
Resolución de problemas	209
Solución de problemas y prácticas recomendadas	209
El perfil ... no se ha podido encontrar en el archivo de configuración	210
Esquema JSON de SAM: no se puede cambiar el esquema en el archivo template.yaml	211
Seguridad	213
Protección de los datos	213
Historial de revisión	215

AWS Toolkit for Visual Studio Code

Esta es la guía del usuario del Kit de herramientas de AWS para VS Code. Si está buscando AWS Toolkit for Visual Studio, consulte la [Guía del usuario de AWS Toolkit for Visual Studio](#).

¿Qué es la AWS Toolkit for Visual Studio Code

El Kit de herramientas para VS Code es una extensión de código abierto para el editor de Visual Studio Code. Esta extensión facilita a los desarrolladores el desarrollo, la depuración local y la implementación de aplicaciones sin servidor que utilicen Amazon Web Services (AWS).

Temas

- [Introducción a la AWS Toolkit for Visual Studio Code](#)
- [Trabajar con AWS servicios y herramientas](#)

Información relacionada

Consulte los siguientes recursos para obtener acceso al código fuente de este conjunto de herramientas o para ver los problemas abiertos en ese momento.

- [Código fuente](#)
- [Identificador de problemas](#)

Para obtener más información sobre el editor de Visual Studio Code, visite <https://code.visualstudio.com/>.

Amazon Q Developer y Amazon CodeWhisperer

A partir del 30 de abril de 2024, Amazon CodeWhisperer pasa a formar parte de Amazon Q Developer, lo que incluye sugerencias de código en línea y escaneos de seguridad de Amazon Q Developer. Descargue la [Extensión de entorno de desarrollo integrado \(IDE\) Amazon Q Developer IDE desde el Marketplace de VS Code](#) para empezar a utilizarla.

Para obtener información detallada sobre el servicio Amazon Q Developer, consulte la Guía del usuario de [Amazon Q Developer](#). Para obtener información detallada sobre los planes y precios de Amazon Q, consulta la guía de [precios de Amazon Q](#).

Descarga del kit de herramientas para VS Code

Puede descargar, instalar y configurar el AWS Toolkit for Visual Studio Code desde el Marketplace de VS Code en su IDE. Si desea leer instrucciones detalladas, consulte la sección [Descarga e instalación](#) en el tema Introducción de esta Guía del usuario.

Descarga del kit de herramientas desde el Marketplace de VS Code

Si lo prefiere, puede descargar los archivos de instalación de AWS Toolkit for Visual Studio Code yendo al [Marketplace de VS Code](#) desde su navegador web.

Kits de herramientas de IDE adicionales de AWS

Además del AWS Toolkit for Visual Studio Code, AWS también ofrece kits de herramientas de IDE para JetBrains y Visual Studio.

Enlaces al AWS Toolkit for JetBrains

- Siga este enlace para [descargar el AWS Toolkit for JetBrains](#) desde el Marketplace de JetBrains.
- Para obtener más información sobre el AWS Toolkit for JetBrains, consulte la Guía del usuario de [AWS Toolkit for JetBrains](#).

Kit de herramientas para Visual Studio

- Siga este enlace para [descargar el kit de herramientas para Visual Studio](#) desde el Marketplace de Visual Studio.
- Para obtener más información sobre el kit de herramientas para Visual Studio, consulte la Guía del usuario del [Kit de herramientas para Visual Studio](#).

Introducción a la AWS Toolkit for Visual Studio Code

El AWS Toolkit for Visual Studio Code hace que sus servicios y recursos de AWS estén disponibles directamente desde su entorno de desarrollo integrado (IDE) de VS Code.

Para empezar, en los siguientes temas se explica cómo preparar, instalar y configurar el AWS Toolkit for Visual Studio Code.

Temas

- [Instalación del AWS Toolkit for Visual Studio Code](#)
- [Conectarse a AWS](#)
- [Cambio de las regiones de AWS](#)
- [Configuración de la cadena de herramientas](#)

Instalación del AWS Toolkit for Visual Studio Code

Requisitos previos

Para empezar a trabajar con el AWS Toolkit for Visual Studio Code desde VS Code, deben cumplirse los siguientes requisitos. Para obtener más información sobre cómo acceder a todos los servicios y recursos de AWS disponibles desde el AWS Toolkit for Visual Studio Code, consulte la sección [the section called “Requisitos previos opcionales”](#) de esta guía.

- VS Code requiere un sistema operativo Windows, macOS o Linux.
- El AWS Toolkit for Visual Studio Code requiere que trabaje con la versión 1.73.0 o una versión posterior de VS Code.

Para obtener información adicional sobre VS Code o para descargar la última versión de VS Code, visite el sitio web de [descargas de VS Code](#).

Descarga e instalación del AWS Toolkit for Visual Studio Code

Puede descargar, instalar y configurar el AWS Toolkit for Visual Studio Code desde el Marketplace de VS Code en su IDE. Si lo prefiere, puede descargar los archivos de instalación de AWS Toolkit for Visual Studio Code yendo al [Marketplace de VS Code](#) desde su navegador web.

Instalación del AWS Toolkit for Visual Studio Code desde el Marketplace del IDE de VS Code

1. Abra la extensión de AWS Toolkit for Visual Studio Code en su IDE de VS Code con el siguiente enlace: [Abrir el Marketplace de VS Code](#).

 Note

Si VS Code no se está ejecutando en su máquina aún, esta operación puede tardar unos minutos mientras VS Code se carga.

2. Desde la extensión de AWS Toolkit for Visual Studio Code en el Marketplace de VS Code, seleccione Instalar para iniciar el proceso de instalación.
3. Cuando se le solicite, elija reiniciar VS Code para completar el proceso de instalación.

Requisitos previos opcionales

Para poder utilizar algunas características del AWS Toolkit for Visual Studio Code, debe tener lo siguiente:

- Cuenta de Amazon Web Services (AWS): no es necesario tener una cuenta de AWS para utilizar el AWS Toolkit for Visual Studio Code, pero la funcionalidad se limita mucho sin ella. Para obtener una cuenta de AWS, vaya a la [página de inicio de AWS](#). Elija Crear una cuenta de AWS o bien Completar registro (si ya ha visitado el sitio con anterioridad).
- Desarrollo de código: el SDK que corresponda al lenguaje que desea usar. Puede realizar la descarga en los siguientes enlaces o usar su administrador de paquetes favorito:
 - SDK para .NET: <https://dotnet.microsoft.com/download>
 - SDK para Node.js: <https://nodejs.org/en/download>
 - SDK para Python: <https://www.python.org/downloads>
 - SDK de Java: <https://aws.amazon.com/corretto/>
 - Ir al SDK: <https://golang.org/doc/install>
- CLI de AWS SAM: es una herramienta de la CLI de AWS que facilita el desarrollo, las pruebas y el análisis locales de las aplicaciones sin servidor. Esto no es necesario para instalar el conjunto de herramientas. Sin embargo, le recomendamos que lo instale (así como Docker, que se describe a continuación) porque es necesario para cualquier funcionalidad de AWS Serverless Application Model (AWS SAM), como [Crear una nueva aplicación sin servidor \(local\)](#).

Para obtener más información, consulte el artículo sobre [instalación de la CLI de AWS SAM](#) en la [guía para desarrolladores de AWS Serverless Application Model](#).

- Docker: la CLI de AWS SAM requiere esta plataforma de contenedores de software de código abierto. Para obtener más información e instrucciones de descarga, consulte [Docker](#).
- Administrador de paquetes: un administrador de paquetes para que pueda descargar y compartir el código de la aplicación.
 - .NET: [NuGet](#)
 - Node.js: [npm](#)
 - Python: [pip](#)
 - Java: [Gradle](#) o [Maven](#)

Conectarse a AWS

La mayoría de los recursos de Amazon Web Services (AWS) se administran a través de una AWS cuenta. No es necesaria una AWS cuenta para utilizarlos AWS Toolkit for Visual Studio Code, pero las funciones del kit de herramientas están limitadas sin conexión.

Si anteriormente configuró una AWS cuenta y se autenticó a través de otro AWS servicio (como el AWS Command Line Interface), este AWS Toolkit for Visual Studio Code detectará automáticamente sus credenciales.

Requisitos previos

Si acabas de crear una cuenta AWS o no la has creado, debes seguir tres pasos principales para conectarla AWS Toolkit for Visual Studio Code a tu AWS cuenta:

1. Crear una AWS cuenta: puedes crear una AWS cuenta desde el [portal de AWS registro](#). Para obtener información detallada sobre cómo configurar una AWS cuenta nueva, consulta el tema de [descripción general](#) de la Guía del usuario de AWS configuración.
2. Configurar la autenticación: existen 3 métodos principales para autenticarse con su AWS cuenta desde AWS Toolkit for Visual Studio Code. Para obtener más información sobre cada uno de estos métodos, consulte el tema [Autenticación y acceso](#) de esta Guía del usuario.
3. Autenticación AWS desde el kit de herramientas: puede conectarse con su AWS cuenta desde el kit de herramientas siguiendo los procedimientos descritos en las siguientes secciones de esta Guía del usuario.

Apertura del panel Inicio de sesión

Siga uno de los procedimientos indicados a continuación para abrir el panel Inicio de sesión del kit de herramientas de AWS .

Para abrir el panel de inicio de sesión del AWS kit de herramientas desde el explorador: AWS

1. Desde el AWS Toolkit for Visual Studio Code, expande EXPLORER.
2. Seleccione el ícono ... para expandir el menú Más acciones....
3. En el menú Más acciones..., elija Conectarse a AWS para abrir el panel Inicio de sesión del kit de herramientas de AWS .

Para abrir el panel Inicio de sesión del kit de herramientas de AWS mediante la paleta de comandos de VS Code:

1. Pulse **Shift+Command+P (Ctrl+Shift+P Windows)** para abrir la paleta de comandos.
2. Escriba **AWS: Add a New Connection** en el campo de búsqueda.
3. Seleccione **AWS: Add a New Connection** para abrir el panel Inicio de sesión del kit de herramientas de AWS .

Conectarse a AWS desde el kit de herramientas

Autenticación y conexión con SSO

Para autenticarse y conectarse AWS con el usuario AWS IAM Identity Center, complete el siguiente procedimiento.

Note

La autenticación con AWS Builder ID o IAM Identity Center abre el portal de AWS autorización en el navegador web predeterminado. Cada vez que sus credenciales caduquen, debe repetir este proceso para renovar la conexión entre su AWS cuenta y el AWS Toolkit for Visual Studio Code.

Autentique y conéctese con el Centro de AWS Identidad de IAM

1. En el panel Inicio de sesión del kit de herramientas de AWS , seleccione la pestaña Personal y, a continuación, seleccione el botón Continuar para avanzar.
2. En el panel Iniciar sesión con IAM Identity Center, escriba la URL de inicio de su organización. Esta dirección URL se la proporciona un administrador o el servicio de asistencia de su empresa.
3. Seleccione su AWS región en el menú desplegable. Esta es la AWS región que aloja tu directorio de identidades.
4. Pulse el botón Continuar y confirme que desea abrir el sitio web de Solicitud de autorización de AWS en su navegador web predeterminado.
5. Siga cada una de las indicaciones en su navegador web predeterminado. Se le notificará cuando el proceso de autorización haya terminado, de modo que será seguro cerrar su navegador y regresar a VS Code.

Autentíquese y conéctese con las credenciales de IAM

Para autenticarse y conectarse con AWS las credenciales de IAM, complete el siguiente procedimiento.

Autentíquese y conéctese con las credenciales de IAM

1. En el panel Inicio de sesión del kit de herramientas de AWS , elija Credenciales de IAM y, a continuación, seleccione el botón Continuar para avanzar.
2. Introduzca las letras **Profile NameAccess Key**, y las **Secret Key** de su AWS cuenta en los campos correspondientes y, a continuación, pulse el botón Continuar para añadir el perfil al archivo de configuración y conectar el kit de herramientas con su cuenta. AWS
3. El Explorador de AWS del kit de herramientas se actualiza para mostrar sus servicios y recursos de AWS una vez completada la autenticación y establecida la conexión.

Autenticación para Amazon CodeCatalyst

Para empezar a trabajar con él CodeCatalyst desde el kit de herramientas, auténticate y conéctate con tu AWS Builder ID o las credenciales del IAM Identity Center.

Los siguientes procedimientos describen cómo autenticar y conectar el kit de herramientas con su cuenta de AWS .

Autenticate y conéctate con un Builder ID AWS

1. En el panel Inicio de sesión del kit de herramientas de AWS , seleccione la pestaña Personal y, a continuación, seleccione el botón Continuar para avanzar.
2. En la parte superior del panel Iniciar sesión con SSO, elija el enlace Ir al inicio de sesión.
3. Siga cada una de las indicaciones en su navegador web predeterminado. Se le notificará cuando el proceso de autorización haya terminado, de modo que será seguro cerrar su navegador y regresar a VS Code.

Autenticación y conexión con IAM Identity Center

1. En el panel Inicio de sesión del kit de herramientas de AWS , seleccione la pestaña Personal y, a continuación, seleccione el botón Continuar para avanzar.
2. En el panel Iniciar sesión con IAM Identity Center, escriba la URL de inicio de su organización. Esta dirección URL se la proporciona un administrador o el servicio de asistencia de su empresa.
3. Selecciona tu AWS región en el menú desplegable. Esta es la AWS región que aloja tu directorio de identidades.
4. Pulse el botón Continuar y confirme que desea abrir el sitio web de Solicitud de autorización de AWS en su navegador web predeterminado.
5. Siga cada una de las indicaciones en su navegador web predeterminado. Se le notificará cuando el proceso de autorización haya terminado, de modo que será seguro cerrar su navegador y regresar a VS Code.

Cambio de las regiones de AWS

Una región de AWS especifica dónde se administran sus recursos de AWS. Su región de AWS predeterminada se detecta cuando se conecta a su cuenta de AWS desde el AWS Toolkit for Visual Studio Code y se muestra automáticamente en el Explorador de AWS.

En las secciones siguientes se describe cómo añadir u ocultar una región desde el Explorador de AWS.

Adición de una región al Explorador de AWS

Realice el procedimiento que se indica a continuación para agregar una región al Explorador de AWS.

1. En VS Code, abra la paleta de comandos expandiendo Ver en el menú principal y seleccionando Paleta de comandos. O bien utilice las siguientes teclas de acceso directo:
 - Windows y Linux: pulse **Ctrl+Shift+P**.
 - macOS: pulse **Shift+Command+P**.
2. En la paleta de comandos, busque **AWS: Show or Hide Regions** y elija AWS: Mostrar u ocultar regiones para ver una lista de las regiones disponibles.
3. En la lista, seleccione la regiones de AWS que desea agregar al Explorador de AWS.
4. Elija el botón Aceptar para confirmar sus elecciones y actualizar el Explorador de AWS.

Ocultar una región en el Explorer de AWS

Siga el procedimiento que se indica a continuación para ocultar una región del Explorador de AWS.

1. Desde el Explorador de AWS, localice la región de AWS que deseé ocultar.
2. Abra el menú contextual (clic con el botón derecho) de la región que desea eliminar.
3. Elija Mostrar u ocultar regiones para abrir las opciones de AWS: Mostrar u ocultar regiones en VS Code.
4. Anule la selección de las regiones que quiere ocultar en la vista del Explorador de AWS.

Configuración de la cadena de herramientas

El AWS Toolkit for Visual Studio Code admite varios idiomas en todos los servicios de AWS. En las siguientes secciones, se describe cómo configurar la cadena de herramientas en diferentes idiomas.

Configurar una cadena de herramientas para .NET Core

1. Asegúrese de tener [instalado](#) el kit de herramientas de AWS para VS Code.
2. Instale la [extensión C#](#). Esta extensión permite a VS Code depurar aplicaciones de .NET Core.
3. Abra una aplicación de AWS Serverless Application Model (AWS SAM) o [cree una](#).

4. Abra la carpeta que contenga `template.yaml`.

Configurar una cadena de herramientas para Node.js

1. Asegúrese de tener [instalado](#) el kit de herramientas de AWS para VS Code.
2. Abra una aplicación de AWS SAM o [cree una](#).
3. Abra la carpeta que contenga `template.yaml`.

 Note

Al depurar una función de Lambda de TypeScript directamente desde el código fuente (la configuración de lanzamiento tiene "target": "code"), el compilador de TypeScript debe instalarse globalmente o en el `package.json` de su proyecto.

Configurar una cadena de herramientas para Python

1. Asegúrese de tener [instalado](#) el kit de herramientas de AWS para VS Code.
2. Instale la [extensión de Python para Visual Studio Code](#). Esta extensión permite a VS Code depurar aplicaciones de Python.
3. Abra una aplicación de AWS SAM o [cree una](#).
4. Abra la carpeta que contenga `template.yaml`.
5. Abra un terminal en la raíz de la aplicación y configure `virtualenv` ejecutando `python -m venv ./venv`.

 Note

Solo necesita configurar `virtualenv` una vez por sistema.

6. Para activar `virtualenv`, ejecute uno de estos comandos:
 - Shell Bash: `./venv/Scripts/activate`
 - PowerShell: `./venv/Scripts/Activate.ps1`

Configurar una cadena de herramientas para Java

1. Asegúrese de tener [instalado](#) el kit de herramientas de AWS para VS Code.
2. Instale la [extensión Java y Java 11](#). Esta extensión permite a VS Code reconocer las funciones de Java.
3. Instale la [extensión del depurador de Java](#). Esta extensión permite a VS Code depurar aplicaciones de Java.
4. Abra una aplicación de AWS SAM o [cree una](#).
5. Abra la carpeta que contenga `template.yaml`.

Configurar una cadena de herramientas para Go

1. Asegúrese de tener [instalado](#) el kit de herramientas de AWS para VS Code.
2. Se requiere Go 1.14 o superior para depurar las funciones de Lambda de Go.
3. Instale la [extensión de Go](#).

 Note

Se requiere la versión 0.25.0 o superior para depurar los tiempos de ejecución de Go1.15+.

4. Instale las herramientas de Go mediante la [paleta de comandos](#):
 - a. En la paleta de comandos, seleccione Go: Install/Update Tools.
 - b. En el conjunto de casillas de verificación, seleccione `dlv` y `gopls`.
5. Abra una aplicación de AWS SAM o [cree una](#).
6. Abra la carpeta que contenga `template.yaml`.

Usar la cadena de herramientas

Una vez que haya configurado su cadena de herramientas, puede usarla para [ejecutar o depurar](#) la aplicación de AWS SAM.

Autenticación y acceso del AWS Toolkit for Visual Studio Code

No es necesario autenticarse con AWS para empezar a usar el AWS Toolkit for Visual Studio Code. Sin embargo, la mayoría de los recursos de AWS se administran a través de una cuenta de AWS. Para acceder a todos los servicios y características del AWS Toolkit for Visual Studio Code, deberá autenticarse con AWS IAM Identity Center, el ID de creador de AWS o las credenciales de IAM.

Los temas siguientes contienen detalles adicionales sobre cada tipo de credencial.

Para obtener más información sobre cómo conectarse a AWS en el AWS Toolkit for Visual Studio Code con sus credenciales actuales, consulte el tema [Conexión a AWS](#) de esta Guía del usuario.

Temas

- [AWS Centro de identidad de IAM](#)
- [AWS Credenciales de IAM](#)
- [ID de creador de AWS para desarrolladores](#)
- [Uso de un proceso de credenciales externo](#)
- [Actualización de firewalls y puertas de enlace para permitir el acceso](#)

AWS Centro de identidad de IAM

AWS IAM Identity Center es la mejor práctica recomendada para gestionar la autenticación de su AWS cuenta.

Para obtener instrucciones detalladas sobre cómo configurar el Centro de Identidad de IAM para los kits de desarrollo de software (SDKs), consulte la sección de [autenticación del Centro de Identidad de IAM](#) de la Guía de referencia sobre herramientas AWS SDKs y herramientas.

Para obtener más información sobre cómo autenticar y conectar el AWS kit de herramientas con sus credenciales actuales del IAM Identity Center, consulte el AWS tema [Conectar a](#) de esta guía del usuario.

AWS Credenciales de IAM

AWS Autenticación de credenciales de IAM con su AWS cuenta mediante claves de acceso almacenadas localmente.

Para obtener más información sobre cómo autenticar y conectar el AWS kit de herramientas con sus credenciales de AWS IAM existentes, consulte el AWS tema [Conectar a](#) de esta guía del usuario.

En las siguientes secciones se describe cómo configurar las credenciales de IAM para autenticarse con su cuenta desde AWS AWS Toolkit for Visual Studio Code

Important

Antes de configurar las credenciales de IAM para autenticarse con su AWS cuenta, tenga en cuenta lo siguiente:

- Si ya configuraste las credenciales de IAM a través de otro AWS servicio (como el AWS CLI), las AWS Toolkit for Visual Studio Code detectará automáticamente y las pondrá a disposición en VS Code.
- AWS recomienda utilizar la autenticación del Centro de Identidad de IAM. Para obtener información adicional sobre las prácticas recomendadas de AWS IAM, consulte la sección [Prácticas recomendadas de seguridad en IAM](#) de la Guía del usuario de AWS Identity and Access Management.
- 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 su lugar, utilice la federación con un proveedor de identidad, como [¿Qué es IAM Identity Center?](#) en la Guía del usuario de AWS IAM Identity Center .

Creación de un usuario de IAM

Antes de configurar la AWS Toolkit for Visual Studio Code autenticación con su AWS cuenta, debe completar el paso 1: Crear un usuario de IAM y el paso 2: Obtener las claves de acceso en el tema [Autenticar con credenciales a largo plazo](#) de la guía de referencia y herramientas AWS SDKs

Note

Paso 3: Actualizar el archivo de credenciales compartidas en la AWS SDKs y la guía de referencia de herramientas es opcional.

Si completa el paso 3, AWS Toolkit for Visual Studio Code detectará automáticamente sus credenciales durante el procedimiento que [the section called “Crear un archivo de](#)

["credenciales compartido a partir del AWS Toolkit for Visual Studio Code"](#) se indica a continuación.

Si no ha completado el paso 3, le guiará AWS Toolkit for Visual Studio Code por el proceso de creación de una, tal `credentials file` como se describe en la [the section called "Crear un archivo de credenciales compartido a partir del AWS Toolkit for Visual Studio Code"](#) siguiente sección.

Crear un archivo de credenciales compartido a partir del AWS Toolkit for Visual Studio Code

El archivo de configuración compartido y el archivo de credenciales compartidos almacenan la información de configuración y credenciales de sus AWS cuentas. Para obtener más información sobre la configuración y las credenciales compartidas, consulte la sección [¿Dónde se almacenan los ajustes de configuración?](#) en la Guía del usuario de AWS Command Line Interface .

Crear un archivo de credenciales compartido a través del AWS Toolkit for Visual Studio Code

1. Pulse **Shift+Command+P** (**Ctrl+Shift+P** Windows) para abrir la paleta de comandos.
2. Escriba **AWS: Add a New Connection** en el campo de búsqueda.
3. Seleccione **AWS: Add a New Connection** para abrir el panel Inicio de sesión del kit de herramientas de AWS .
4. En el panel Inicio de sesión del kit de herramientas de AWS , elija Credenciales de IAM y, a continuación, seleccione el botón Continuar para avanzar.
5. Introduzca las **Profile Name** `Access Key`, y **Secret Key** las de su AWS cuenta en los campos correspondientes y, a continuación, pulse el botón Continuar para añadir el perfil al archivo de configuración y conectar el kit de herramientas con su AWS cuenta.
6. El Explorador de AWS del kit de herramientas se actualiza para mostrar sus servicios y recursos de AWS una vez completada la autenticación y establecida la conexión.

Note

En este ejemplo, supongamos que **[Profile Name]** contiene errores de sintaxis y provoca un error en la autenticación.

[Profile_Name]

```
xaws_access_key_id= AKIAI44QH8DHBEXAMPLE  
xaws_secret_access_key= wJalrXUtnFEMI/K7MDENG/bPxRficyEXAMPLEKEY
```

El siguiente es un ejemplo de un mensaje de registro que se genera como respuesta a un intento de autenticación fallido.

```
2022-11-02 22:01:54 [ERROR]: Profile [Profile_Name] is not a valid Credential  
Profile: not supported by the Toolkit  
2022-11-02 22:01:54 [WARN]: Shared Credentials Profile [Profile_Name] is not  
valid. It will not be used by the toolkit.
```

Añadir perfiles de credenciales adicionales

Puede añadir varias credenciales a sus archivos de configuración. Para ello, abra la paleta de comandos y seleccione Kit de herramientas de AWS : Crear perfil de credenciales. Esto abrirá el archivo de credenciales. En esta página, puede añadir un perfil nuevo debajo del primer perfil, como se muestra en el siguiente ejemplo:

```
# Amazon Web Services Credentials File used by AWS CLI, SDKs, and tools  
# This file was created by the AWS Toolkit for Visual Studio Code extension.  
#  
# Your AWS credentials are represented by access keys associated with IAM users.  
# For information about how to create and manage AWS access keys for a user, see:  
# https://docs.aws.amazon.com/IAM/latest/UserGuide/id\_credentials\_access-keys.html  
#  
# This credential file can store multiple access keys by placing each one in a  
# named "profile". For information about how to change the access keys in a  
# profile or to add a new profile with a different access key, see:  
# https://docs.aws.amazon.com/cli/latest/userguide/cli-config-files.html  
#  
[Profile1_Name]  
# The access key and secret key pair identify your account and grant access to AWS.  
aws_access_key_id = AKIAIOSFODNN7EXAMPLE  
# Treat your secret key like a password. Never share your secret key with anyone. Do  
# not post it in online forums, or store it in a source control system. If your secret  
# key is ever disclosed, immediately use IAM to delete the access key and secret key  
# and create a new key pair. Then, update this file with the replacement key details.  
aws_secret_access_key = wJalrXUtnFEMI/K7MDENG/bPxRficyEXAMPLEKEY
```

```
[Profile2_Name]
```

```
aws_access_key_id = AKIAI44QH8DHBEXAMPLE
aws_secret_access_key = je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY
```

ID de creador de AWS para desarrolladores

El ID de creador de AWS es una cuenta de AWS adicional que es opcional u obligatoria para determinados servicios de AWS. Para obtener información detallada sobre el método de autenticación del ID de creador de AWS, consulte el tema [Inicio de sesión con el ID de creador de AWS](#) de la Guía del usuario de Inicio de sesión en AWS.

Para obtener más información sobre cómo autenticar y conectar el kit de herramientas de AWS con su ID de creador de AWS actual, consulte el tema [Conectarse a AWS](#) de esta Guía del usuario.

Uso de un proceso de credenciales externo

Puede configurar los procesos AWS Toolkit for Visual Studio Code de credenciales que no son compatibles directamente con AWS, modificando su `shared config file`

La modificación de los procesos de obtención de credenciales es la misma para el AWS Toolkit for Visual Studio Code `.shared config file` AWS Command Line Interface Para obtener información detallada sobre cómo configurar las credenciales externas, consulte el tema [Obtención de credenciales con un proceso externo](#) en la Guía del AWS Command Line Interface usuario de .

Actualización de firewalls y puertas de enlace para permitir el acceso

Si filtra el acceso a AWS dominios o puntos de enlace de URL específicos mediante una solución de filtrado de contenido web, los siguientes puntos de enlace deben estar permitidos en la lista para poder acceder a todos los servicios y funciones disponibles a través de Amazon Q. AWS Toolkit for Visual Studio Code

AWS Toolkit for Visual Studio Code Puntos de enlace

Las siguientes son listas de puntos finales y referencias AWS Toolkit for Visual Studio Code específicos que deben incluirse en la lista.

Punto de conexión

<https://idetoolkits.amazonaws.com/endpoints.json>

Archivos alojados

<https://idetoolkits-hostedfiles.amazonaws.com/Notifications/VSCode/startup/1.x.json>
<https://idetoolkits-hostedfiles.amazonaws.com/Notifications/VSCode/emergency/1.x.json>

Compatibilidad con esquemas

<https://raw.githubusercontent.com/aws/serverless-application-model/main/samtranslator/schema/schema.json>
<https://api.github.com/repos/devfile/api/releases/latest>
[https://raw.githubusercontent.com/devfile/api/\\${devfileSchemaVersion}/schemas/latest/devfile.json](https://raw.githubusercontent.com/devfile/api/${devfileSchemaVersion}/schemas/latest/devfile.json)

cSharpSamDepurar el script de instalación

<https://aka.ms/getvsdbgps1>
<https://aka.ms/getvsdbgsh>

Puntos de conexión del complemento de Amazon Q

A continuación, se muestra una lista de referencias y puntos de conexión específicos del complemento de Amazon Q que deben figurar en la lista de permitidos.

https://idetoolkits-hostedfiles.amazonaws.com/* (Plugin for configs)
https://idetoolkits.amazonaws.com/* (Plugin for endpoints)
https://aws-toolkit-language-servers.amazonaws.com/* (Language Server Process)

<https://client-telemetry.us-east-1.amazonaws.com/> (Telemetry)
<https://cognito-identity.us-east-1.amazonaws.com> (Telemetry)
<https://aws-language-servers.us-east-1.amazonaws.com> (Language Server Process)

Puntos de conexión de Amazon Q Developer

A continuación, se muestra una lista de referencias y puntos de conexión específicos de Amazon Q Developer que deben figurar en la lista de permitidos.

<https://codewhisperer.us-east-1.amazonaws.com> (Inline, Chat, QSDA,...)
<https://q.us-east-1.amazonaws.com> (Inline, Chat, QSDA....)
<https://desktop-release.codewhisperer.us-east-1.amazonaws.com/> (Download url for CLI.)
<https://specs.q.us-east-1.amazonaws.com> (Url for autocomplete specs used by CLI)
* aws-language-servers.us-east-1.amazonaws.com (Local Workspace context)

Puntos de conexión de la transformación de código de Amazon Q

A continuación, se muestra una lista de referencias y puntos de conexión específicos de la transformación de código de Amazon Q que deben figurar en la lista de permitidos.

https://docs.aws.amazon.com/amazonq/latest/qdeveloper-ug/security_iam_manage-access-with-policies.html

Puntos de conexión de autenticación

A continuación, se muestra una lista de referencias y puntos de conexión de autenticación que deben figurar en la lista de permitidos.

[Directory ID or alias].awsapps.com
* oidc.[Region].amazonaws.com
*.sso.[Region].amazonaws.com
*.sso-portal.[Region].amazonaws.com
*.aws.dev
*.awsstatic.com

```
*.console.aws.a2z.com  
*.sso.amazonaws.com
```

Puntos de conexión de identidad

Las siguientes listas contienen puntos finales que son específicos de la identidad, como AWS IAM Identity Center el ID del AWS constructor.

AWS IAM Identity Center

Para obtener más información sobre los puntos de conexión necesarios para el IAM Identity Center, consulte el tema [Habilitar el IAM Identity Center](#) en la Guía del usuario.

IAM Identity Center para empresas

```
https://[Center director id].awsapps.com/start (should be permitted to initiate auth)  
https://us-east-1.signin.aws (for facilitating authentication, assuming IAM Identity  
Center is in IAD)  
https://oidc.(us-east-1).amazonaws.com  
https://log.sso-portal.eu-west-1.amazonaws.com.  
https://portal.sso.eu-west-1.amazonaws.com
```

AWS ID de constructor

```
https://view.awsapps.com/start (must be blocked to disable individual tier)  
https://codewhisperer.us-east-1.amazonaws.com and q.us-east-1.amazonaws.com (should be  
permitted)
```

Telemetría

A continuación, se muestran puntos de conexión específicos de telemetría que deben figurar en la lista de permitidos.

```
https://telemetry.aws-language-servers.us-east-1.amazonaws.com/  
https://client-telemetry.us-east-1.amazonaws.com
```

Referencias

A continuación, se muestra una lista de referencias de puntos de conexión.

```
idetoolkits-hostedfiles.amazonaws.com.  
cognito-identity.us-east-1.amazonaws.com.  
amazonwebservices.gallery.vsassets.io.  
eu-west-1.prod.pr.analytics.console.aws.azz.com.  
prod.pa.cdn.uis.awsstatic.com.  
portal.sso.eu-west-1.amazonaws.com.  
log.sso-portal.eu-west-1.amazonaws.com.  
prod.assets.shortbread.aws.dev.  
prod.tools.shortbread.aws.dev.  
prod.log.shortbread.aws.dev.  
a.b.cdn.console.awsstatic.com.  
assets.sso-portal.eu-west-1.amazonaws.com.  
oidc.eu-west-1.amazonaws.com.  
aws-toolkit-language-servers.amazonaws.com.  
aws-language-servers.us-east-1.amazonaws.com.  
idetoolkits.amazonwebservices.com.
```

Trabajar con AWS servicios y herramientas

AWS Toolkit for Visual Studio Code Hace que AWS los servicios, las herramientas y los recursos estén disponibles directamente en VS Code. A continuación, se muestra una lista de temas de la guías que abarcan cada uno de los servicios del kit de herramientas para VS Code y sus características. Elija un servicio o una herramienta para obtener más información sobre su utilidad, cómo configurarlo y cómo utilizar sus características básicas.

Temas

- [Uso de características experimentales](#)
- [Uso de los servicios de AWS en el Explorador de AWS](#)
- [Documentos de AWS](#)
- [Amazon CodeCatalyst para VS Code](#)
- [Uso de Amazon API Gateway](#)
- [Utilizándolo AWS App Runner con AWS Toolkit for Visual Studio Code](#)
- [Desarrollador de aplicaciones de AWS](#)
- [AWS Infrastructure Composer](#)
- [AWS CDK para VS Code](#)
- [Uso de pilas de AWS CloudFormation](#)
- [Uso de Registros de CloudWatch mediante el AWS Toolkit for Visual Studio Code](#)
- [Amazon DocumentDB](#)
- [Amazon Elastic Compute Cloud](#)
- [Uso del servicio Amazon Elastic Container Registry](#)
- [Trabajar con Amazon Elastic Container Service](#)
- [Trabajo con Amazon EventBridge](#)
- [AWS Analizador de acceso IAM](#)
- [Working with AWS IoT in AWS Toolkit for Visual Studio Code](#)
- [Funciones de AWS Lambda](#)
- [Amazon Redshift en el kit de herramientas para VS Code](#)
- [Uso de Amazon S3](#)
- [Estudio unificado de Amazon SageMaker para VS Code](#)
- [Uso de aplicaciones sin servidor](#)

- [Uso de documentos de Automatización de Systems Manager](#)
- [AWS Step Functions](#)
- [Uso de Threat Composer](#)
- [Uso de recursos](#)

Uso de características experimentales

Las características experimentales ofrecen acceso anticipado a las características en AWS Toolkit for Visual Studio Code antes de que se lancen oficialmente.

Warning

Dado que las características experimentales se siguen evaluando y actualizando, es posible que tengan problemas de usabilidad, y es posible que se eliminen del AWS Toolkit for Visual Studio Code sin previo aviso.

Puede activar características experimentales para determinados servicios de AWS en la sección Kit de herramientas de AWS del panel Ajustes en su IDE de VS Code.

1. Para editar la configuración de AWS en VS Code, seleccione Archivo, Preferencias, Configuración.
2. En el panel Configuración, expanda Extensiones y seleccione AWS Toolkit.
3. En AWS: Experimentos, seleccione las casillas de verificación de las características experimentales a las que desee acceder antes del lanzamiento. Si desea desactivar una característica experimental, desactive la casilla correspondiente.

Uso de los servicios de AWS en el Explorador de AWS

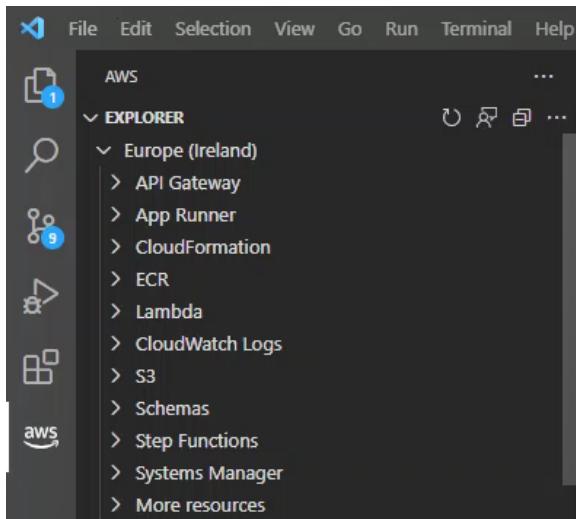
El Explorador de AWS ofrece una vista de algunos de los servicios de AWS con los que puede trabajar cuando utilice el AWS Toolkit for Visual Studio Code.

En esta sección se explica cómo obtener acceso y utilizar el Explorador de AWS en VS Code. Se da por hecho que ya ha [instalado y configurado](#) el kit de herramientas para VS Code en el sistema.

Algunos puntos importantes:

- Si el kit de herramientas se ha instalado y configurado correctamente, deberían aparecer elementos en el Explorador de AWS. Para ver el Explorador de AWS, elija el icono de AWS en la barra de actividades.

Por ejemplo:



- Para algunas características, se requieren permisos de AWS específicos. Por ejemplo, para ver las funciones de AWS Lambda en su cuenta de AWS, las credenciales que ha configurado en [Autenticación y acceso](#) deben incluir al menos permisos de Lambda de solo lectura. Consulte los temas siguientes para obtener más información acerca de los permisos que necesita cada característica.
- Si quiere interactuar con servicios de AWS que no estén visibles inmediatamente en el Explorador de AWS, puede ir a Más recursos y elegir entre cientos de recursos para añadir a la interfaz.

Por ejemplo, puede elegir AWS Toolkit:CodeArtifact::Repository entre los tipos de recurso disponibles. Después de añadir este tipo de recurso a Más recursos, puede expandir la entrada para ver una lista de recursos que crean diferentes repositorios de CodeArtifact con sus propias propiedades y atributos. Además, puede describir las propiedades y los atributos de los recursos en plantillas con formato JSON, que puede guardar para crear nuevos recursos en la nube de AWS.

Documentos de AWS

El AWS Toolkit for Visual Studio Code admite el JSON Schema de AWS Serverless Application Model para AWS SAM templates, lo que mejora la experiencia de creación de plantillas al permitir las definiciones, el autocompletado y la validación directamente en VS Code. AWS Documents

admite tanto recursos de AWS SAM como de CloudFormation. Para obtener más información, consulte los siguientes recursos:

- Para obtener información específica sobre el esquema JSON, consulte el sitio web JSON-Schema.org ([Esquema JSON](#)).
- Para obtener más información sobre las plantillas de AWS SAM, consulte el tema [Anatomía de la plantilla de AWS SAM](#) de la Guía para desarrolladores de AWS Serverless Application Model.
- Para obtener más información sobre los tipos de recursos y propiedades de AWS, consulte el tema [Referencia de tipos de recursos y propiedades de AWS](#) de la Guía del usuario de CloudFormation.
- Para obtener información detallada sobre el esquema de AWS SAM que utiliza el kit de herramientas de AWS, consulte el esquema de [AWS Serverless Application Model](#) en el repositorio de GitHub para AWS.

Introducción a AWS Documents

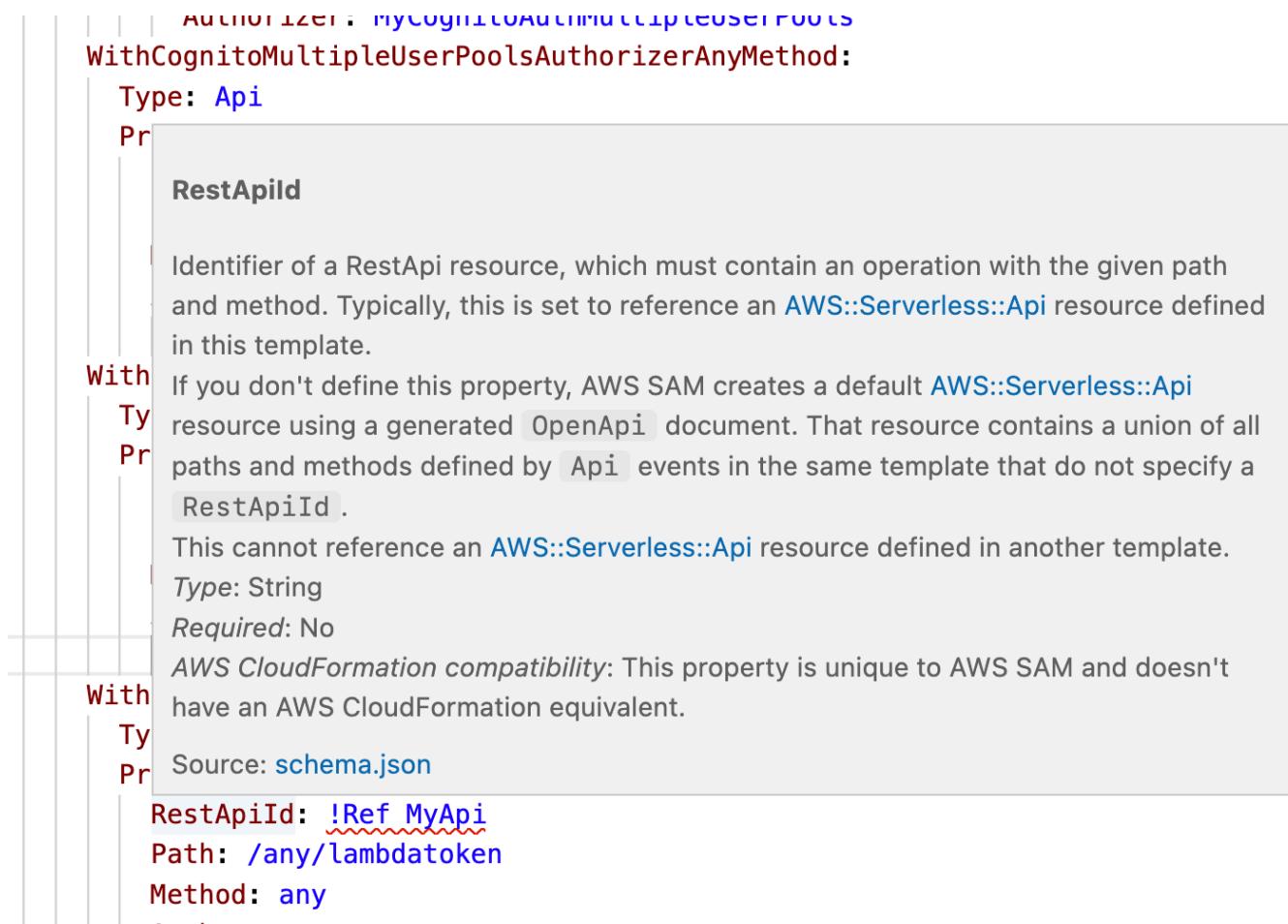
Para empezar a trabajar con AWS Documents en VS Code, instale la extensión del AWS Toolkit for Visual Studio Code desde el entorno de desarrollo integrado (IDE) o el [Marketplace de VS Code](#) y abra cualquier plantilla de AWS SAM.

Visualización de documentación, autocompletado y validación en VS Code

La visualización de documentación, el autocompletado y la validación son características incluidas con el kit de herramientas de AWS. Consulte la imagen siguiente para ver un ejemplo del aspecto de estas características en VS Code.

- Para ver documentación de la plantilla de AWS SAM abierta, coloque el puntero sobre una entrada de línea del documento.
- Para el autocompletado, empiece a escribir en la plantilla de AWS SAM a fin de activar una ventana emergente con sugerencias basadas en los datos ingresados.
- La plantilla de AWS SAM se analiza de forma automática para su validación y los errores se resaltan con el icono de una bombilla que puede seleccionar para obtener sugerencias adicionales.

Consulte la imagen siguiente para ver un ejemplo del aspecto de estas características en VS Code.



Amazon CodeCatalyst para VS Code

¿Qué es Amazon CodeCatalyst?

Amazon CodeCatalyst es un espacio de colaboración basado en la nube para equipos de desarrollo de software. A través de él AWS Toolkit for Visual Studio Code, puede ver y administrar sus CodeCatalyst recursos directamente desde VS Code. También puede trabajar directamente en la nube utilizando el AWS kit de herramientas para lanzar entornos de computación virtual Dev Environments que ejecuten VS Code. Para obtener más información sobre el CodeCatalyst servicio, consulta la Guía del CodeCatalyst usuario de [Amazon](#).

En los siguientes temas se describe cómo conectar VS Code y cómo trabajar con él CodeCatalyst desde el Toolkit for VS Code. CodeCatalyst

Temas

- [Cómo empezar con CodeCatalyst el Toolkit for VS Code](#)

- [Trabajar con CodeCatalyst los recursos de Amazon en VS Code](#)
- [Trabajo con el kit de herramientas en entornos de desarrollo](#)
- [Solución de problemas de Amazon CodeCatalyst y VS Code](#)

Cómo empezar con CodeCatalyst el Toolkit for VS Code

Para empezar a trabajar con CodeCatalyst VS Code, sigue estos procedimientos.

Temas

- [Crear una CodeCatalyst cuenta](#)
- [Conectar el AWS kit de herramientas con CodeCatalyst](#)

Crear una CodeCatalyst cuenta

Debe tener un ID de AWS constructor o AWS IAM Identity Center credenciales activos para conectarse CodeCatalyst desde el Toolkit for VS Code. Para obtener más información sobre AWS Builder ID, el centro de identidad de IAM y CodeCatalyst las credenciales, consulte la CodeCatalyst sección [Configuración con](#) de la Guía del CodeCatalyst usuario.

Conectar el AWS kit de herramientas con CodeCatalyst

Para conectar el AWS kit de herramientas con tu CodeCatalyst cuenta, consulta la CodeCatalyst sección [Autenticación para Amazon](#) en el AWS tema Conectarse a de esta Guía del usuario.

Trabajar con CodeCatalyst los recursos de Amazon en VS Code

En las siguientes secciones se proporciona una descripción general de las funciones de administración de CodeCatalyst recursos de Amazon que están disponibles en el Toolkit for VS Code.

Para obtener más información sobre los entornos de desarrollo y cómo puede acceder a ellos CodeCatalyst, consulte la sección [Entornos](#) de desarrollo de la Guía del CodeCatalyst usuario de Amazon.

En las secciones siguientes se describe cómo crear, abrir y usar entornos de desarrollo desde VS Code.

Temas

- [Clonación de un repositorio](#)
- [Apertura de un entorno de desarrollo](#)
- [Creación de un CodeCatalyst entorno de desarrollo](#)
- [Creación de un entorno de desarrollo a partir de un repositorio de terceros](#)
- [CodeCatalyst comandos en VS Code](#)

Clonación de un repositorio

CodeCatalyst es un servicio basado en la nube que requiere que esté conectado a la nube para trabajar en CodeCatalyst proyectos. Si prefieres trabajar en un proyecto de forma local, puedes clonar tus CodeCatalyst repositorios en tu máquina local y sincronizarlos con tu CodeCatalyst proyecto en línea la próxima vez que te conectes a la nube.

Para clonar un repositorio de tu CodeCatalyst cuenta a VS Code con el AWS kit de herramientas, sigue estos pasos:

 Note

Si va a clonar un repositorio desde un servicio de terceros, es posible que se le pida autenticarse con las credenciales de ese servicio.

Mientras se clona el repositorio, VS Code muestra el progreso en la ventana de estado

Clonando el repositorio. Una vez clonado el repositorio, se muestra el mensaje: ¿Desea abrir el repositorio clonado?.

1. En el kit de herramientas para VS Code, expanda el explorador de HERRAMIENTAS PARA DESARROLLADORES.
2. Amplíe CodeCatalyst y elija Clonar repositorio.
3. En el cuadro de diálogo Seleccione un CodeCatalyst repositorio, busque el repositorio que desee clonar y, a continuación, selecciónelo para abrir el cuadro de diálogo Elegir una carpeta para clonar.
4. Elija Seleccionar la ubicación del repositorio para cerrar el mensaje y empezar a clonar el repositorio.
5. En la ventana de diálogo, seleccione una de las siguientes opciones para completar el proceso de clonación:

- Para abrir el repositorio en la ventana actual de VS Code, seleccione Abrir.
- Para abrir el repositorio en una ventana de VS Code nueva, seleccione Abrir en una ventana nueva.
- Para completar el proceso de clonación sin abrir el repositorio, cierra la ventana de diálogo.

Apertura de un entorno de desarrollo

Para abrir un entorno de desarrollo existente en VS Code, complete los siguientes pasos.

Note

Al seleccionar el entorno de desarrollo, se inicia el proceso de conexión con VS Code CodeCatalyst al abrir su entorno de desarrollo. Durante este proceso, VS Code muestra las actualizaciones de progreso en una ventana de CodeCatalyst estado. Cuando el proceso se completa, la ventana de estado se actualiza.

- Si el entorno de desarrollo no se abre, el estado se actualiza con información sobre el motivo del error del proceso y un enlace para abrir los registros del proceso.
- Si el proceso se completa correctamente, su entorno de desarrollo se abre en una nueva ventana, desde VS Code.

1. En el kit de herramientas para VS Code, expanda el explorador de HERRAMIENTAS PARA DESARROLLADORES.
2. Expanda CodeCatalysty elija Abrir entorno de desarrollo para abrir el cuadro de diálogo Seleccionar CodeCatalyst un entorno de desarrollo en VS Code.
3. En el cuadro de diálogo CodeCatalyst Seleccione un entorno de desarrollo, elija el entorno de desarrollo que desee abrir.

Creación de un CodeCatalyst entorno de desarrollo

Para crea un entorno de desarrollo nuevo, complete los siguientes pasos:

Note

Al trabajar con la instrucción DETACH RL, observe lo siguiente:

- AWS recomienda especificar un alias porque simplifica la organización y mejora las capacidades de búsqueda en los entornos de desarrollo.
- Los entornos de desarrollo guardan su trabajo de manera permanente. Esto significa que puede pausar su entorno de desarrollo sin perder el trabajo. Detener su entorno de desarrollo reduce el costo necesario para mantenerlo en funcionamiento.
- El almacenamiento es la única configuración que no se puede cambiar después de haber creado su entorno de desarrollo.
- VS Code muestra el progreso de la creación de su entorno de desarrollo en una ventana de estado. Una vez creado el entorno de desarrollo, VS Code lo abre en una ventana nueva y aparece también el mensaje ¿Confía en los autores de los archivos en esta carpeta?. Acepte los términos y condiciones para seguir trabajando en su entorno de desarrollo.

1. En el kit de herramientas para VS Code, expanda el explorador de HERRAMIENTAS PARA DESARROLLADORES.
2. Amplíe CodeCatalyst elija la opción Crear un entorno de desarrollo para abrir el menú Crear un CodeCatalyst entorno de desarrollo en VS Code.
3. En la sección Código fuente, elija una de las siguientes opciones:
 - Usa un CodeCatalyst repositorio existente: crea un entorno de desarrollo a partir de un CodeCatalyst repositorio existente. Debe seleccionar el CodeCatalyst proyecto y la rama.
 - Crear un entorno de desarrollo vacío: crea un entorno de desarrollo vacío.
4. (Opcional) En la sección de alias, introduzca un nombre alternativo para su entorno de desarrollo.
5. (Opcional) En la sección de configuración de los entornos de desarrollo, cambie los siguientes ajustes para adaptarlos a sus necesidades específicas.
 - Computación: seleccione Editar computación para cambiar la cantidad de potencia de procesamiento y RAM asignadas a su sistema.
 - Tiempo de espera: seleccione Editar tiempo de espera para cambiar la cantidad de tiempo de inactividad del sistema que se permite antes de detener el entorno de desarrollo.
 - Almacenamiento: seleccione Editar tamaño de almacenamiento para cambiar la cantidad de espacio de almacenamiento asignado a su sistema.

6. Elija Crear entorno de desarrollo para crear su entorno de desarrollo.

Creación de un entorno de desarrollo a partir de un repositorio de terceros

Puede crear entornos de desarrollo a partir de un repositorio de terceros si lo vincula al repositorio como origen.

La vinculación a un repositorio de terceros como fuente se gestiona a nivel de proyecto en CodeCatalyst. Para obtener instrucciones y detalles adicionales sobre cómo conectar un repositorio de terceros a tu entorno de desarrollo, consulta el tema [Cómo vincular un repositorio de código fuente](#) en la Guía del CodeCatalyst usuario de Amazon.

CodeCatalyst comandos en VS Code

Hay comandos adicionales de VS Code que se asignan a funciones CodeCatalyst relacionadas que no se muestran directamente en el AWS kit de herramientas.

Para ver una lista de los comandos que están asignados CodeCatalyst desde la paleta de comandos, complete los siguientes pasos:

1. En el kit de herramientas para VS Code, expanda el explorador de HERRAMIENTAS PARA DESARROLLADORES.
2. Seleccione Mostrar CodeCatalyst comandos para abrir la paleta de comandos con una búsqueda previamente rellenada de CodeCatalyst.
3. Seleccione un CodeCatalyst comando de la lista para activarlo.

Trabajo con el kit de herramientas en entornos de desarrollo

Los entornos de desarrollo son entornos informáticos virtuales para Amazon CodeCatalyst. En las secciones siguientes se describe cómo crear, lanzar y trabajar desde entornos de desarrollo usando el AWS Toolkit for Visual Studio Code.

Para obtener información detallada sobre los entornos de desarrollo, consulte el tema [Entornos](#) de desarrollo en la Guía del CodeCatalyst usuario de Amazon.

Configuración del entorno de desarrollo con archivos de desarrollo

La especificación de `devfile` es un formato de estándar abierto para YAML que se puede usar para definir configuraciones para entornos de desarrollo. Cada entorno de desarrollo tiene un archivo de

desarrollo. Si crea un entorno de desarrollo sin un repositorio o desde un repositorio que no contiene un archivo de desarrollo, se aplicará automáticamente un valor predeterminado al código fuente. Los archivos de desarrollo se pueden actualizar desde CodeCatalyst o desde su IDE. Los procesos para actualizar un archivo de desarrollo en una instancia local o remota de VS Code son idénticos, pero si actualiza un archivo de desarrollo de forma local, debe enviar las actualizaciones a su repositorio de origen antes de que las actualizaciones surtan efecto.

Para obtener información detallada sobre la configuración de entornos de desarrollo con archivos de desarrollo, consulte el tema [Configuración de su entorno](#) de desarrollo en la Guía del CodeCatalyst usuario de Amazon.

El siguiente procedimiento describe cómo editar el archivo de desarrollo desde una instancia remota del kit de herramientas mientras se ejecuta en un entorno de desarrollo.

Important

Si edita el `Devfile` desde VS Code, tenga en cuenta lo siguiente:

- Al cambiar el nombre del archivo de desarrollo o del componente del archivo de desarrollo, se reemplaza el contenido de su directorio raíz. Todo el contenido anterior se pierde y no se puede recuperar.
- Si crea un entorno de desarrollo sin un archivo de desarrollo en la carpeta raíz o que no esté asociado a un repositorio de origen, al crearlo se generará un archivo de desarrollo con los ajustes de configuración predeterminados de su entorno de desarrollo.
- Para obtener instrucciones sobre cómo definir y configurar el `Devfile`, consulte el documento sobre cómo [añadir comandos](#) en el sitio web [devfile.io](#).

1. En el kit de herramientas para VS Code, expanda el explorador de HERRAMIENTAS PARA DESARROLLADORES.
2. Amplíe CodeCatalyst seleccione Abrir archivo de desarrollo para abrirlo `devfile.yaml` en una nueva ventana del editor, dentro de su entorno de desarrollo actual.
3. Actualice su archivo de desarrollo y guarde los cambios desde el editor de VS Code.
4. La próxima vez que inicie su entorno de desarrollo, la configuración se actualizará para que coincida con las especificaciones definidas en su `Devfile`.

Autenticarse y conectarse AWS desde su entorno de desarrollo

Para acceder a todos sus AWS recursos desde su entorno de desarrollo, debe autenticar y conectar su instancia remota del kit de herramientas a su cuenta. AWS La instancia remota del kit de herramientas se autentica automáticamente con las credenciales heredadas de la instancia local del kit de herramientas cuando se lanza el entorno de desarrollo.

Los procedimientos para actualizar las credenciales de una instancia remota del kit de herramientas son idénticos a los de la experiencia de autenticación en la instancia local del kit de herramientas. Para obtener instrucciones detalladas sobre cómo actualizar las credenciales, autenticarse y conectarse a AWS desde el kit de herramientas, consulte la sección AWS sección [Conexión a](#) en el tema Introducción de esta Guía de usuario.

Para obtener información adicional sobre cada uno de los métodos de AWS autenticación compatibles con el AWS Toolkit for Visual Studio Code, consulte el tema [Autenticación y acceso](#) de esta guía del usuario.

Uso del kit de herramientas para VS Code en entornos de desarrollo

Después de abrir o crear un entorno de desarrollo en VS Code, puede trabajar desde el kit de herramientas para VS Code de forma similar a como lo hace desde una instancia local de VS Code. Los entornos de desarrollo que ejecutan VS Code están configurados para instalar automáticamente el AWS kit de herramientas y conectarse con su ID de AWS constructor.

Detención de un entorno de desarrollo

Para detener su entorno de desarrollo actual:

1. En el kit de herramientas para VS Code, expanda el explorador de HERRAMIENTAS PARA DESARROLLADORES.
2. Amplíe CodeCatalyst elija Stop Dev Environment.
3. Cuando VS Code se lo solicite, confirme que desea detener su entorno de desarrollo.
4. Su entorno de desarrollo ha detenido correctamente cuando VS Code cierra la conexión remota y vuelve a una instancia de desarrollo local.

Apertura de la configuración del entorno de desarrollo

Para abrir la configuración de su entorno de desarrollo actual, siga los siguientes pasos:

Note

No puede cambiar la cantidad de espacio de almacenamiento que se asigna al entorno de desarrollo después de crearlo.

1. En el kit de herramientas para VS Code, expanda el explorador de HERRAMIENTAS PARA DESARROLLADORES.
2. Amplíe CodeCatalyst y elija Abrir configuración para abrir la vista de configuración del entorno de desarrollo correspondiente a su entorno de desarrollo actual.
3. En la vista Dev Environment Settings (Configuración del entorno de desarrollo), las siguientes secciones contienen opciones para su entorno de desarrollo:
 - Alias: consulte y cambie el alias asignado a su entorno de desarrollo.
 - Estado: consulte el estado actual de su entorno de desarrollo, el proyecto al que se ha asignado, y detenga su entorno de desarrollo.
 - Archivo de desarrollo: consulte el nombre y la ubicación del Devfile de su entorno de desarrollo. Pulse el botón Abrir en el editor para abrir el Devfile.
 - Compute Settings (Configuración de computación): cambie el tamaño y el valor predeterminado de Timeout Length (Duración del tiempo de espera) correspondientes a su entorno de desarrollo.

Solución de problemas de Amazon CodeCatalyst y VS Code

Los siguientes temas abordan posibles problemas técnicos al trabajar con Amazon CodeCatalyst y VS Code.

Temas

- [Versión de VS Code](#)
- [Permisos para Amazon CodeCatalyst](#)
- [Conexión a un entorno de desarrollo desde el kit de herramientas de VS Code](#)

Versión de VS Code

Se espera que su versión de VS Code configure un controlador `vscode://` URLs en su sistema. Sin este controlador, no puede acceder a todas las CodeCatalyst funciones del kit de AWS herramientas. Por ejemplo, se produce un error al iniciar un entorno de desarrollo desde VS Code Insiders. Esto se debe a que VS Code Insiders maneja `vscode-insiders://` URLs y no maneja `vscode://` URLs.

Permisos para Amazon CodeCatalyst

Los siguientes son los requisitos de permisos de archivos para trabajar con archivos CodeCatalyst desde AWS Toolkit for Visual Studio Code:

- Establezca sus propios permisos de acceso para el archivo `~/.ssh/config` en `read` y `write`. Restrinja los permisos de `write` para todos los demás usuarios.
- Establezca sus propios permisos de acceso para el `~/.ssh/id_dsa` y los archivos `~/.ssh/id_rsa` en `read` únicamente. Restrinja los permisos de `read`, `write` y `execute` para todos los demás usuarios.
- El archivo `globals.context.globalStorageUri.fsPath` debe estar en una ubicación en la que se pueda escribir.

Conexión a un entorno de desarrollo desde el kit de herramientas de VS Code

Si aparece el siguiente error al intentar conectarse a un entorno de desarrollo desde el AWS Toolkit for Visual Studio Code:

Su `~/.ssh/config` tiene una sección de `aws-devenv-*` que puede estar obsoleta.

- Pulse el botón Abrir configuración... para abrir su archivo `~/.ssh/config` en el editor de código VS.
- Desde el editor, seleccione y elimine el contenido de la sección `Host aws-devenv-*`.
- Guarde los cambios que haya realizado en el `Host aws-devenv-*` de `~/.ssh/config`. Guarde el archivo a continuación.
- Intente conectarse de nuevo a un entorno de desarrollo desde el kit de herramientas para VS Code.

Uso de Amazon API Gateway

Puede buscar y ejecutar los recursos remotos de API Gateway en su cuenta de AWS conectada mediante el AWS Toolkit for Visual Studio Code.

 Note

Esta característica no es compatible con la depuración.

Para buscar y ejecutar recursos remotos de API Gateway

1. Desde el Explorador de AWS, elija API Gateway para expandir el menú. Se muestra una lista con los recursos remotos de API Gateway.
2. Localice el recurso de API Gateway que desee invocar, abra su menú contextual (botón derecho del ratón) y seleccione Invocar en AWS.
3. En el formulario de parámetros, especifique los parámetros de invocación.
4. Para ejecutar el recurso remoto de API Gateway, elija Invocar. Los resultados se muestran en la vista Resultados de VS Code.

Utilizándolo AWS App Runner con AWS Toolkit for Visual Studio Code

[AWS App Runner](#) proporciona una forma rápida, sencilla y rentable de implementar directamente desde el código fuente o una imagen de contenedor hasta una aplicación web escalable y segura en la AWS nube. Al usarlo, no necesita aprender nuevas tecnologías, decidir qué servicio de cómputo usar ni saber cómo aprovisionar y configurar AWS los recursos.

Puede utilizarlos AWS App Runner para crear y administrar servicios basados en una imagen fuente o un código fuente. Si utiliza una imagen de origen, puede elegir una imagen de contenedor pública o privada que se almacena en un repositorio de imágenes. App Runner es compatible con los siguientes proveedores de repositorios de imágenes:

- Amazon Elastic Container Registry (Amazon ECR): almacena imágenes privadas en tu cuenta AWS
- Amazon Elastic Container Registry Public (Amazon ECR Public): almacena imágenes de lectura pública.

Si elige la opción de código fuente, puede implementarla desde un repositorio de código fuente mantenido por un proveedor de repositorios compatible. Actualmente, App Runner es compatible [GitHub](#) como proveedor de repositorios de código fuente.

Requisitos previos

Para interactuar con App Runner mediante el uso de lo, se AWS Toolkit for Visual Studio Code requiere lo siguiente:

- Una AWS cuenta
- Una versión de AWS Toolkit for Visual Studio Code esas características AWS App Runner

Además de estos requisitos básicos, asegúrese de que todos los usuarios pertinentes de IAM tengan permisos para interactuar con el servicio de App Runner. También necesita obtener información específica sobre su fuente de servicio, como el URI de la imagen del contenedor o la conexión al GitHub repositorio. Necesitará esta información al crear el servicio de App Runner.

Configuración de permisos de IAM para App Runner

La forma más sencilla de conceder los permisos necesarios para App Runner consiste en adjuntar una política AWS gestionada existente a la entidad AWS Identity and Access Management (IAM) correspondiente, concretamente a un usuario o grupo. App Runner proporciona dos políticas administradas que puede adjuntar a los usuarios de IAM:

- `AWSAppRunnerFullAccess`: permite a los usuarios realizar todas las acciones de App Runner.
- `AWSAppRunnerReadOnlyAccess`: permite a los usuarios publicar y ver detalles sobre los recursos de App Runner.

Además, si elige un repositorio privado de Amazon Elastic Container Registry (Amazon ECR) como origen de servicio, debe crear el siguiente rol de acceso para su servicio de App Runner:

- `AWSAppRunnerServicePolicyForECRAccess`: permite a App Runner acceder a imágenes de Amazon Elastic Container Registry (Amazon ECR) de su cuenta.

Puede crear este rol automáticamente al configurar la instancia de servicio con la paleta de comandos de VS Code.

Note

La función `AWSServiceRoleForAppRunner` vinculada al servicio permite AWS App Runner realizar las siguientes tareas:

- Envía los registros a los grupos de CloudWatch registros de Amazon Logs.
- Crea reglas de Amazon CloudWatch Events para suscribirte al envío de imágenes de Amazon Elastic Container Registry (Amazon ECR).

No necesita crear manualmente el rol vinculado a servicios. Al crear una AWS App Runner en la API Consola de administración de AWS o mediante operaciones de API a las que se invoca AWS Toolkit for Visual Studio Code, AWS App Runner crea automáticamente esta función vinculada al servicio.

Para obtener más información, consulte [Identity and access management for App Runner](#) (Administración de identidades y accesos para App Runner) en la Guía para desarrolladores de AWS App Runner .

Obtención de orígenes de servicio para App Runner

Puedes usar AWS App Runner para implementar servicios a partir de una imagen fuente o un código fuente.

Source image

Si vas a realizar la implementación desde una imagen de origen, puedes obtener un enlace al repositorio de esa imagen desde un registro de AWS imágenes público o privado.

- [Registro privado de Amazon ECR: copie el URI de un repositorio privado que utilice la consola de Amazon ECR en <https://console.aws.amazon.com/ecr/> los repositorios.](#)
- Registro público de Amazon ECR: copie el URI de un repositorio público que utiliza la Galería pública de Amazon ECR en <https://gallery.ecr.aws/>.

Note

También puede obtener el URI de un repositorio privado de Amazon ECR directamente desde el Explorador de AWS en el kit de herramientas para VS Code:

- Abra el AWS Explorador y expanda el nodo ECR para ver la lista de repositorios de esa región. AWS
- Haga clic con el botón derecho en un repositorio y elija Copy Repository URI (Copiar URI del repositorio) para copiar el enlace en el portapapeles.

Puede especificar el URI del repositorio de imágenes al configurar la instancia de servicio con la paleta de comandos de VS Code.

Para obtener más información, consulte [App Runner service based on a source image](#) (Servicio de App Runner basado en una imagen de origen) en la Guía para desarrolladores de AWS App Runner .

Source code

Para que tu código fuente se despliegue en un AWS App Runner servicio, dicho código debe almacenarse en un repositorio de Git mantenido por un proveedor de repositorios compatible. App Runner admite un proveedor de repositorios de código fuente: [GitHub](#).

Para obtener información sobre cómo configurar un GitHub repositorio, consulte la [documentación de introducción](#) en GitHub.

Para implementar el código fuente en un servicio de App Runner desde un GitHub repositorio, App Runner establece una conexión a GitHub. Si tu repositorio es privado (es decir, no es accesible públicamente en él GitHub), debes proporcionar a App Runner los detalles de conexión.

Important

Para crear GitHub conexiones, debes usar la consola de App Runner (<https://console.aws.amazon.com/apprunner>) para crear una conexión que enlace a GitHub. Puede seleccionar las conexiones que están disponibles en la página de GitHub conexiones al configurar su instancia de servicio con la paleta de comandos de VS Code.

Para obtener más información, consulte [Managing App Runner connections](#) (Administración de conexiones de App Runner) en la Guía para desarrolladores de AWS App Runner .

La instancia de servicio de App Runner proporciona un tiempo de ejecución gestionado que permite compilar y ejecutar el código. AWS App Runner actualmente admite los siguientes tiempos de ejecución:

- Tiempo de ejecución administrado de Python
- Tiempo de ejecución administrado de Node.js

Como parte de la configuración del servicio, proporciona información sobre cómo se crea e inicia el servicio de App Runner. Puede ingresar esta información mediante Command Palette (Paleta de comandos) o especificar un [archivo de configuración de App Runner](#) con formato YAML. Los valores de este archivo le indican a App Runner cómo crear e iniciar el servicio y proporcionan contexto de tiempo de ejecución. Esto incluye la configuración de red y las variables de entorno correspondientes. El archivo de configuración se llama `apprunner.yaml`. Se agrega de manera automática al directorio raíz del repositorio de la aplicación.

Precios

Se le cobrará por los recursos informáticos y de memoria que utiliza la aplicación. Además, si automatiza las implementaciones, también pagará una tarifa mensual fija por cada aplicación que cubre todas las implementaciones automatizadas de ese mes. Si opta por implementar desde el código fuente, también pagará una tarifa de compilación por el tiempo que demora App Runner en crear un contenedor a partir del código fuente.

Para más información, consulte [Precios de AWS App Runner](#).

Temas

- [Creación de servicios de App Runner](#)
- [Administración de los servicios de App Runner](#)

Creación de servicios de App Runner

Puede crear un servicio de App Runner en el kit de herramientas para VS Code mediante el Explorador de AWS y la paleta de comandos de VS Code. Una vez que decida crear un servicio en una AWS región específica, los pasos numerados que se indican en la paleta de comandos le guiarán durante el proceso de configuración de la instancia de servicio en la que se ejecuta la aplicación.

Antes de crear un servicio de App Runner, asegúrese de haber completado los [requisitos previos](#).

Esto incluye proporcionar los permisos de IAM pertinentes y confirmar el repositorio de origen específico que desea implementar.

Para crear un servicio de App Runner

1. Abra el AWS Explorador, si aún no está abierto.
2. Haga clic con el botón derecho en el nodo de App Runner y elija Create Service (Crear servicio).

Se muestra la paleta de comandos.

3. En Select a source code location type (Seleccionar un tipo de ubicación de código fuente), elija ECR o Repository (Repositorio).

Si elige ECR, especifique una imagen de contenedor de un repositorio mantenido por Amazon Elastic Container Registry. Si elige Repository (Repositorio), especifique un repositorio de código fuente mantenido por un proveedor de repositorios compatible. Actualmente, App Runner es compatible [GitHub](#) como proveedor de repositorios de código fuente.

Implementación desde ECR

1. En Select or enter an image repository (Seleccionar o ingresar un repositorio de imágenes), elija o ingrese la URL del repositorio de imágenes que mantiene su registro privado de Amazon ECR o la Galería pública de Amazon ECR.

 Note

Si especifica un repositorio de la Galería pública de Amazon ECR, asegúrese de que las implementaciones automáticas estén desactivadas, ya que App Runner no admite implementaciones automáticas para una imagen en un repositorio público de ECR.

Las implementaciones automáticas están desactivadas de forma predeterminada. Esto se indica cuando el ícono del encabezado de la paleta de comandos presenta una línea diagonal a través de él. Si eligió activar las implementaciones automáticas, un mensaje le informa que esta opción puede generar costos adicionales.

2. Si en el paso de la paleta de comandos se informa de que no se han encontrado etiquetas, debe retroceder un paso y seleccionar un repositorio que contenga una imagen de contenedor etiquetada.

3. Si utilizas un registro privado de Amazon ECR, necesitas el rol de acceso a ECR, Role, AppRunner ECRAccess que permite a App Runner acceder a las imágenes de Amazon Elastic Container Registry (Amazon ECR) de tu cuenta. Elija el icono "+" en el encabezado de la paleta de comandos para crear automáticamente este rol. (No se requiere un rol de acceso si la imagen se almacena en la Galería pública de Amazon ECR, en el que las imágenes están disponibles públicamente).
4. En Port (Puerto), ingrese el puerto IP utilizado por el servicio (por ejemplo, Puerto 8000).
5. En Configure environment variables (Configurar variables de entorno), puede especificar un archivo que contenga variables de entorno que se utilizan para personalizar el comportamiento de la instancia de servicio. También puede omitir este paso.
6. En Name your service (Asignar un nombre al servicio), ingrese un nombre único sin espacios y pulse Enter (Intro).
7. En Select instance configuration (Seleccionar la configuración de instancia), elija una combinación de unidades de CPU y memoria en GB para la instancia de servicio.

Cuando se crea el servicio, el estado cambia de Creating (Creando) a Running (En ejecución).

8. Una vez que el servicio comience a ejecutarse, haga clic derecho en él y elija Copy Service URL (Copiar URL del servicio).
9. Para acceder a la aplicación implementada, pegue la URL copiada en la barra de direcciones del navegador web.

Implementación desde un repositorio remoto

1. En Seleccione una conexión, elija una conexión a la que se enlace. GitHub AWS Las conexiones que están disponibles para su selección aparecen en la página de GitHub conexiones de la consola de App Runner.
2. En Seleccione un GitHub repositorio remoto, elija o introduzca una URL para el repositorio remoto.

Los repositorios remotos que ya están configurados con la administración del control de origen (SCM) de Visual Studio Code están disponibles para su selección. También puede pegar un enlace al repositorio si no aparece en la lista.

3. En Select a branch (Seleccionar una rama), elija qué rama Git del código fuente desea implementar.

4. En Choose configuration source (Elegir el origen de configuración), especifique cómo desea definir la configuración del tiempo de ejecución.

Si elige Use configuration file (Usar archivo de configuración), la instancia de servicio se configura mediante la configuración definida por el archivo de configuración de `apprunner.yaml`. Este archivo se encuentra en el directorio raíz del repositorio de la aplicación.

Si elige Configure todos los ajustes aquí, utilice la paleta de comandos para especificar lo siguiente:

- Runtime (Tiempo de ejecución): elija Python 3 o Nodejs 12.
 - Build command (Comando de creación): ingrese el comando para crear la aplicación en el entorno en tiempo de ejecución de la instancia de servicio.
 - Start command (Comando de inicio): ingrese el comando para iniciar la aplicación en el entorno en tiempo de ejecución de la instancia de servicio.
5. En Port (Puerto), ingrese el puerto IP utilizado por el servicio (por ejemplo, Puerto 8000).
 6. En Configure environment variables (Configurar variables de entorno), puede especificar un archivo que contenga variables de entorno que se utilizan para personalizar el comportamiento de la instancia de servicio. También puede omitir este paso.
 7. En Name your service (Asignar un nombre al servicio), ingrese un nombre único sin espacios y pulse Enter (Intro).
 8. En Select instance configuration (Seleccionar la configuración de instancia), elija una combinación de unidades de CPU y memoria en GB para la instancia de servicio.

Cuando se crea el servicio, el estado cambia de Creating (Creando) a Running (En ejecución).

9. Una vez que el servicio comience a ejecutarse, haga clic derecho en él y elija Copy Service URL (Copiar URL del servicio).
10. Para acceder a la aplicación implementada, pegue la URL copiada en la barra de direcciones del navegador web.

 Note

Si se produce un error en el intento de crear un servicio de App Runner, el servicio muestra un estado Create failed (Error al crear) en el Explorador de AWS . Para obtener sugerencias

acerca de la solución de problemas, consulte [When service creation fails](#) (Cuando falla la creación del servicio) en la Guía para desarrolladores de App Runner.

Administración de los servicios de App Runner

Tras crear un servicio de App Runner, puede administrarlo mediante el panel del AWS explorador para llevar a cabo las siguientes actividades:

- [Pausa y reanudación de los servicios de App Runner](#)
- [Implementación de los servicios de App Runner](#)
- [Visualización de secuencias de registros para App Runner](#)
- [Eliminación de servicios de App Runner](#)

Pausa y reanudación de los servicios de App Runner

Si necesitas deshabilitar tu aplicación web temporalmente y detener la ejecución del código, puedes pausar el servicio AWS App Runner. App Runner reduce la capacidad de computación del servicio a cero. Cuando esté listo para volver a ejecutar la aplicación, reanude el servicio de App Runner. App Runner aprovisiona nueva capacidad de computación, implementa la aplicación en ella y la ejecuta.

Important

Se le cobrará por App Runner solo cuando se esté ejecutando. Por lo tanto, puede pausar y reanudar la aplicación según sea necesario para administrar los costos. Esto resulta especialmente útil en escenarios de desarrollo y pruebas.

Para pausar el servicio de App Runner

1. Abre el AWS Explorador, si aún no está abierto.
2. Expanda App Runner para ver la lista de servicios.
3. Haga clic con el botón derecho en el servicio y elija Pause (Pausar).
4. En el cuadro de diálogo que se muestra, elija Confirm (Confirmar).

Mientras el servicio se pausa, el estado del servicio cambia de Running(En ejecución) a Pausing (Pausando) y luego a Paused (En pausa).

Para reanudar el servicio de App Runner

1. Abre el AWS Explorador, si aún no está abierto.
2. Expanda App Runner para ver la lista de servicios.
3. Haga clic con el botón derecho en el servicio y elija Resume (Reanudar).

Mientras se reanuda el servicio, el estado del servicio cambia de Resuming (Reanudando) a Running (En ejecución).

Implementación de los servicios de App Runner

Si elige la opción de implementación manual del servicio, debe iniciar explícitamente cada implementación en el servicio.

1. Abre el AWS Explorador, si aún no está abierto.
2. Expanda App Runner para ver la lista de servicios.
3. Haga clic con el botón derecho en el servicio y elija Start Deployment (Iniciar implementación).
4. Mientras se implementa la aplicación, el estado del servicio cambia de Deploying (Implementando) a Running (En ejecución).
5. Para confirmar que la aplicación se implementó correctamente, haga clic con el botón derecho en el mismo servicio y elija Copy Service URL (Copiar URL del servicio).
6. Para acceder a la aplicación web implementada, pegue la URL copiada en la barra de direcciones del navegador web.

Visualización de secuencias de registros para App Runner

Usa CloudWatch los registros para monitorear, almacenar y acceder a tus flujos de registros para servicios como App Runner. Un flujo de registro es una secuencia de eventos de registro que comparten la misma fuente.

1. Expanda App Runner para ver la lista de instancias de servicios.
2. Expanda una instancia de servicio específica para ver la lista de grupos de registros. (Un grupo de registros es un grupo de flujos de registro que comparten la misma configuración de retención, monitoreo y control de acceso).
3. Haga clic con el botón derecho en un grupo de registros y elija View Log Streams (Ver secuencias de registro).

4. En la paleta de comandos, elija un flujo de registro del grupo.

El editor de VS Code muestra la lista de eventos de registro que componen el flujo. Puede elegir cargar eventos anteriores o más recientes en el editor.

Eliminación de servicios de App Runner

Important

Si elimina el servicio de App Runner, se elimina de forma permanente y los datos almacenados se eliminan. Si necesita volver a crear el servicio, App Runner debe volver a recuperar el origen y crearlo si se trata de un repositorio de código. La aplicación web obtiene un nuevo dominio de App Runner.

1. Abre el AWS Explorador, si aún no está abierto.
2. Expanda App Runner para ver la lista de servicios.
3. Haga clic con el botón derecho en un servicio y elija Delete Service (Eliminar servicio).
4. En la paleta de comandos, escriba delete y después pulse la tecla Intro para confirmar.

El servicio eliminado muestra el estado Deleting (Eliminando) y, a continuación, el servicio desaparece de la lista.

Desarrollador de aplicaciones de AWS

Desarrollador de aplicaciones de AWS para el AWS Toolkit for Visual Studio Code es su guía para crear proyectos de forma visual, recorrerlos en iteración de manera local e implementar sus aplicaciones en AWS.

Los siguientes temas describen cómo trabajar con el Desarrollador de aplicaciones de AWS desde el AWS Toolkit for Visual Studio Code.

Temas

- [Trabajar con AWS Application Builder](#)

Trabajar con AWS Application Builder

En las siguientes secciones se describe cómo acceder a AWS Application Builder en AWS Toolkit for Visual Studio Code. Con Application Builder, puede crear proyectos de forma visual, iterarlos localmente e implementarlos AWS allí. Para obtener una descripción general de las funciones y los posibles casos de uso de Application Builder y su AWS Lambda experiencia local, consulte el YouTube vídeo para AWS desarrolladores [*New* AWS Lambda Local IDE Experience!](#).

Trabajando con el explorador de AWS Application Builder

Para acceder a Application Builder en el kit de AWS herramientas, abra el kit de AWS herramientas en VS Code y, a continuación, expanda el explorador de AWS Application Builder. El explorador de AWS Application Builder contiene un enlace para abrir el tutorial de Application Builder en una pestaña del editor de VS Code y muestra las carpetas de su espacio de trabajo actual de VS Code que contienen recursos relacionados con AWS Application Builder.

Desde el explorador de Application Builder del AWS kit de herramientas, hay cuatro project-folder-level acciones a las que se puede acceder desde los íconos de los botones situados junto a la carpeta del proyecto o abriendo el menú contextual de la carpeta del proyecto (haciendo clic con el botón derecho):

- Abrir archivo de plantilla: abre el archivo de plantilla en el explorador de VS Code.
- Abrir con Infrastructure Composer: abre el archivo de plantilla con AWS Infrastructure Composer en el editor de VS Code. Para obtener información detallada sobre cómo trabajar con AWS Infrastructure Composer, consulte el tema [Qué es AWS Infrastructure Composer](#) en la Guía para desarrolladores de AWS Infrastructure Composer.
- Crear una plantilla SAM: abre el cuadro de diálogo Especificar parámetros para la creación en el AWS kit de herramientas. Puede optar por Especificar marcas de compilación para la creación o Usar valores predeterminados de samconfig. Para obtener información detallada sobre AWS SAM las plantillas, consulte el tema sobre la [anatomía de las plantillas](#) en la Guía para AWS Serverless Application Modeldesarrolladores.
- Implementar aplicación de SAM: abre el cuadro de diálogo Select deployment command en VS Code, donde puede optar por implementar la aplicación o sincronizarla para actualizar una aplicación que ya ha implementado. Para obtener información detallada sobre la implementación de AWS SAM aplicaciones, consulte el tema [Implemente su aplicación y sus recursos](#) en la Guía para AWS Serverless Application Modeldesarrolladores.

Hay dos acciones a las que puede acceder desde los íconos de los botones situados junto a la AWS Lambda función en la carpeta del proyecto o haciendo clic con el botón derecho en la AWS Lambda función:

- Invocación y configuración de depuración locales: abre el formulario de invocación y configuración de depuración locales en su editor de VS Code. Con este formulario puede crear, editar y ejecutar configuraciones de lanzamiento de tipo:aws - sam. Para obtener más información sobre las configuraciones de depuración de SAM, consulte el tema [Opciones de configuración para la depuración de aplicaciones sin servidor](#) de esta Guía del usuario.

 Note

En la actualidad, VS Code no admite la depuración de una aplicación.NET Core en una ARM64 arquitectura. Si intenta depurar una aplicación .NET Core, aparece el siguiente error:

`The vsdbg debugger does not currently support the arm64 architecture. Function will run locally without debug.`

Para obtener más información sobre este problema, consulte este problema [VSCode-csharp](#) en el repositorio. DotNet GitHub

- Apertura del controlador de funciones: abre el archivo del proyecto que contiene el controlador de funciones.

Hay 2 acciones adicionales disponibles para las funciones desplegadas AWS Lambda .

- Invocación remota: abre el menú Remote invoke configuration en el editor de VS Code.
- Búsqueda de registros: abre el cuadro de diálogo Search logs en VS Code.

Tutorial del Desarrollador de aplicaciones

El tutorial de Application Builder es una guía step-by-step interactiva que explica el proceso de creación de una nueva aplicación con AWS Application Builder. Puede acceder al tutorial de Application Builder desde dos lugares: el explorador de Application Builder en la AWS Toolkit for Visual Studio Code pestaña de bienvenida de VS Code. Al seleccionar el tutorial de Application Builder en el explorador de Application Builder del AWS kit de herramientas, se abre el tutorial de Application Builder en la pestaña de bienvenida de VS Code de la ventana del editor de VS Code.

El Tutorial del Desarrollador de aplicaciones consta de cinco secciones principales:

1. Instalación

La sección de instalación comprueba si ha instalado las AWS CLI herramientas que requiere Application Builder y otras herramientas opcionales. Si no dispone de las herramientas necesarias o las que tiene están obsoletas, se le guiará por el proceso de instalación de las versiones correctas.

Para comprobar si tiene instaladas las herramientas correctas AWS CLI y opcionales, seleccione el botón correspondiente a la herramienta AWS CLI u otra que desee probar. Tras seleccionar un botón, los Registros del kit de herramientas de AWS se actualizarán y VS Code mostrará un mensaje de alerta con el estado de las herramientas. Si necesita instalar o actualizar las herramientas, el Tutorial del Desarrollador de aplicaciones se actualiza con las instrucciones y los recursos que necesita para continuar.

Para obtener información detallada sobre la instalación de AWS CLI, consulte la [sección Instalar o actualizar a la última versión del AWS CLI](#) tema en la Guía para AWS CLI desarrolladores. Para obtener información detallada sobre la instalación de la AWS SAM CLI, consulte el tema [Instalación de la AWS SAM CLI](#) en la Guía para desarrolladores de AWS SAM CLI.

2. Elegir la plantilla de aplicación

La sección Elegir la plantilla de aplicación le guía a través del proceso de creación de una aplicación a partir de una plantilla.

Para elegir una plantilla e inicializar la aplicación, complete los siguientes pasos:

1. En el Tutorial del Desarrollador de aplicaciones, seleccione la sección Elegir la plantilla de aplicación para que aparezca una lista de las opciones de plantilla en la pantalla.
2. Elija una plantilla de la lista y, a continuación, pulse el botón Initialize your project para abrir un cuadro de diálogo de VS Code.
3. Complete los pasos del cuadro de diálogo de VS Code para inicializar la nueva aplicación.
4. El AWS kit de herramientas registra la actualización con el estado de la aplicación durante el proceso de inicialización.
5. Para ver la aplicación en el explorador Desarrollador de aplicaciones, elija el ícono Actualizar explorador Desarrollador de aplicaciones para actualizar el explorador con los cambios.

3. Iterar localmente

La sección Iterar localmente contiene imágenes de ejemplo que muestran cómo se puede realizar iteraciones con las funciones de Application Builder disponibles en los exploradores de VS Code y Toolkit AWS.

Para obtener información adicional sobre todas las funciones de Application Builder disponibles en los exploradores de VS Code y AWS Toolkit, consulte la sección Cómo trabajar con el explorador de Application Builder, que se encuentra en este tema de la guía del usuario.

4. Implemente en AWS

La AWS sección Implementar en contiene información sobre cómo configurar las credenciales con las que conectarse con el AWS fin de implementar la aplicación y ejemplos de cómo implementar la aplicación con Application Builder.

Para conectarse AWS con sus credenciales existentes desde el tutorial de Application Builder, complete uno de los siguientes procedimientos.

Fuerza laboral: inicie sesión AWS con el inicio de sesión único.

1. En la AWS sección Implementar en del tutorial de Application Builder, pulse el botón Configurar credenciales para abrir el menú AWS: INICIO DE SESIÓN en el explorador del AWS kit de herramientas.
2. En el menú AWS: Iniciar sesión, elija Personal y, a continuación, elija el botón Continuar para continuar.
3. Escriba la URL de inicio en el campo proporcionado, elija la Región de AWS en el menú desplegable y, a continuación, elija el botón Continuar para avanzar.
4. En la ventana emergente de VS Code, confirme que desea abrir el sitio de AWS autenticación en su navegador predeterminado.
5. Complete los pasos de autenticación en el navegador predeterminado. Cuando la autenticación haya finalizado y sea seguro cerrar la ventana del navegador, se le notificará.

Credenciales de IAM: almacene las claves para usarlas con AWS CLI las herramientas.

1. En la AWS sección Implementar en del tutorial de Application Builder, pulse el botón Configurar credenciales para abrir el menú AWS: INICIO DE SESIÓN en el explorador del kit de AWS herramientas.

2. En el menú AWS: Iniciar sesión, elija Credenciales de IAM y, a continuación, elija el botón Continuar para avanzar.
3. Escriba un Nombre de perfil en el campo proporcionado, especifique sus valores de **Access Key** y **Secret Key** y, a continuación, elija el botón Continuar para avanzar.
4. VS Code muestra el estado de la autenticación y le notifica si se ha completado o si sus credenciales no son válidas.

Para obtener información detallada sobre cómo configurar sus credenciales para el despliegue con el AWS CLI, consulte el tema [Configurar el AWS CLI](#) tema en la Guía para AWS CLIdesarrolladores. Para obtener información adicional sobre cómo conectarse AWS desde el AWS kit de herramientas con sus credenciales actuales, consulte el AWS tema [Conectarse a](#) de esta guía del usuario.

AWS Infrastructure Composer

Puede usar el AWS Toolkit for Visual Studio Code para trabajar con AWS Infrastructure Composer. AWS Infrastructure Composer es un creador visual de aplicaciones de AWS que ayuda a diseñar la arquitectura de las aplicaciones y a visualizar su infraestructura de CloudFormation.

Para obtener información detallada sobre AWS Infrastructure Composer, consulte la Guía del usuario de [AWS Infrastructure Composer](#).

Los siguientes temas describen cómo trabajar con AWS Infrastructure Composer desde el AWS Toolkit for Visual Studio Code.

Temas

- [Uso de AWS Infrastructure Composer en el kit de herramientas](#)

Uso de AWS Infrastructure Composer en el kit de herramientas

AWS Infrastructure Composer para el AWS Toolkit for Visual Studio Code permite diseñar aplicaciones de forma visual a través de un lienzo interactivo. También puede usar Infrastructure Composer para visualizar y modificar las plantillas de CloudFormation y AWS Serverless Application Model (AWS SAM). Mientras trabaja con Infrastructure Composer, los cambios se almacenan de manera constante, lo que le permite cambiar sin problemas entre la edición de archivos directamente en el editor de VS Code o el uso del lienzo interactivo.

Para obtener información detallada sobre AWS Infrastructure Composer, información de introducción y tutoriales, consulte la Guía del usuario de [AWS Infrastructure Composer](#).

En las secciones siguientes se describe cómo acceder a AWS Infrastructure Composer desde el AWS Toolkit for Visual Studio Code.

Acceso a AWS Infrastructure Composer desde el kit de herramientas

Hay tres formas principales de acceder a AWS Infrastructure Composer desde el kit de herramientas.

Acceso a AWS Infrastructure Composer desde una plantilla existente

1. En VS Code, abra un archivo de plantilla existente en el editor de VS Code.
2. En la ventana del editor, haga clic en el botón AWS Infrastructure Composer ubicado en la esquina superior derecha de dicha ventana.
3. AWS Infrastructure Composer abre y visualiza el archivo de plantilla en la ventana del editor de VS Code.

Acceso a AWS Infrastructure Composer desde el menú contextual (con el botón derecho)

1. En VS Code, haga clic con el botón derecho en el archivo de plantilla que desea abrir con AWS Infrastructure Composer.
2. En el menú contextual, elija la opción Abrir con App Composer.
3. AWS Infrastructure Composer abre y visualiza el archivo de plantilla en una nueva ventana del editor de VS Code.

Acceso a AWS Infrastructure Composer desde la paleta de comandos

1. Desde VS Code, abra la paleta de comandos presionando **Cmd + Shift + P** o **Ctrl + Shift + P** (Windows)
2. En el campo de búsqueda, ingrese **AWS Infrastructure Composer** y elija AWS Infrastructure Composer cuando aparezca en los resultados.
3. Elija el archivo de plantilla que desea abrir. AWS Infrastructure Composer abrirá y visualizará el archivo de plantilla en una nueva ventana del editor de VS Code.

AWS CDK para VS Code

Esta es una documentación preliminar para una característica en versión de vista previa. Está sujeta a cambios.

El servicio AWS CDK le permite trabajar con aplicaciones de [AWS Cloud Development Kit \(AWS CDK\)](#). Puede encontrar información detallada sobre AWS CDK en la [guía para desarrolladores de AWS Cloud Development Kit \(AWS CDK\)](#).

Las aplicaciones de AWS CDK se componen de bloques constitutivos, conocidos como [componentes](#), que incluyen definiciones para las pilas de CloudFormation y los recursos de AWS que contienen. Utilice el Explorador de AWS CDK para visualizar las [pilas](#) y los [recursos](#) definidos en los componentes de AWS CDK. La visualización se proporciona en una vista de árbol en el panel Herramientas para desarrolladores del editor de Visual Studio Code (VS Code).

En esta sección se ofrece información sobre cómo acceder y utilizar AWS CDK en el editor de VS Code. Se da por hecho que ya ha [instalado y configurado](#) el kit de herramientas para VS Code de su IDE local.

Temas

- [Trabajar con aplicaciones de AWS CDK](#)

Trabajar con aplicaciones de AWS CDK

Esta es una documentación preliminar para una característica en versión de vista previa. Está sujeta a cambios.

Utilice el Explorador de AWS CDK del kit de herramientas de AWS para VS Code para visualizar las aplicaciones de AWS CDK y trabajar con ellas.

Requisitos previos

- Asegúrese de que el sistema cumple los requisitos previos especificados en [Instalación del kit de herramientas para VS Code](#).

- Instale la interfaz de la línea de comandos de AWS CDK, tal y como se describe en las primeras secciones de [Introducción a AWS CDK](#) en la Guía para desarrolladores de AWS Cloud Development Kit (AWS CDK).

 **Important**

La versión de AWS CDK debe ser 1.17.0 o posterior. Use `cdk --version` en la línea de comandos para ver qué versión se está ejecutando.

Visualización de una aplicación de AWS CDK

Utilice el Explorador de AWS CDK del kit de herramientas de AWS para VS Code para administrar las [pilas](#) y los [recursos](#) que están almacenados en los componentes de CDK de sus aplicaciones. El Explorador de AWS CDK muestra los recursos en una vista de árbol con la información definida en el archivo `tree.json`, que se crea al ejecutar el comando `cdk synth`. El archivo `tree.json` se encuentra en el directorio `cdk.out` de la aplicación de forma predeterminada.

Para empezar a utilizar el Explorador de AWS CDK del kit de herramientas, tendrá que crear una aplicación de CDK.

1. Complete los primeros pasos del [tutorial Hello World](#) ubicado en la [Guía para desarrolladores de AWS CDK](#).

 **Important**

Cuando llegue al paso de implementación de la pila del tutorial, deténgase y vuelva a esta guía.

 **Note**

Puede ejecutar los comandos proporcionados en el tutorial, como `mkdir` y `cdk init`, en la línea de comandos de un sistema operativo o en una ventana de Terminal dentro del editor de VS Code.

2. Una vez que haya completado los pasos necesarios del tutorial de CDK, abra el contenido de CDK que ha creado en el editor de VS Code.

3. En el panel de navegación de AWS, expanda el encabezado CDK (versión preliminar). Sus aplicaciones de CDK y sus recursos asociados se muestran ahora en la vista de árbol del Explorador de CDK.

Notas importantes

- Cuando cargue aplicaciones de CDK en el editor de VS Code, puede cargar varias carpetas a la vez. Cada carpeta puede contener varias aplicaciones de CDK, como se muestra en la imagen anterior. El Explorador de AWS CDK encuentra aplicaciones en el directorio raíz del proyecto y sus subdirectorios directos.
- Al realizar los primeros pasos del tutorial, puede observar que el último comando que ejecuta es **cdk synth**, que genera el archivo `tree.json`. Si cambia aspectos de una aplicación de CDK, por ejemplo, si agrega más recursos, debe ejecutar ese comando de nuevo para ver los cambios reflejados en la vista de árbol.

Realización de otras operaciones en una aplicación de AWS CDK

Puede utilizar el editor de VS Code para realizar otras operaciones en una aplicación de CDK tal y como lo haría utilizando la línea de comandos de su sistema operativo u otras herramientas. Por ejemplo, puede actualizar los archivos de código en el editor e implementar la aplicación mediante una ventana Terminal de VS Code.

Para probar este tipo de acciones, utilice el editor de VS Code para continuar el [tutorial Hello World](#) de la Guía para desarrolladores de AWS CDK. Asegúrese de realizar el último paso, Destruir los recursos de la aplicación, para no incurrir en costos inesperados en la cuenta de AWS.

Uso de pilas de AWS CloudFormation

El AWS Toolkit for Visual Studio Code ofrece soporte para pilas de [AWS CloudFormation](#). El kit de herramientas para VS Code le permite realizar algunas tareas con las pilas de AWS CloudFormation; por ejemplo, eliminarlas.

Temas

- [Eliminación de una pila de CloudFormation](#)
- [Cree una AWS CloudFormation plantilla utilizando el AWS Toolkit for Visual Studio Code](#)

Eliminación de una pila de CloudFormation

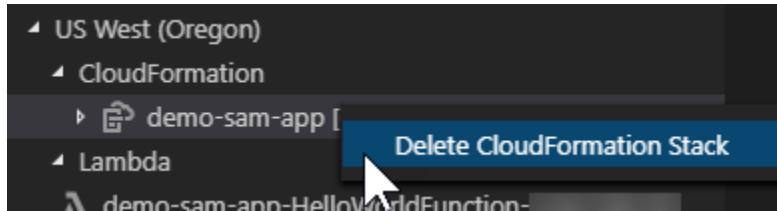
Puede usar el AWS Toolkit for Visual Studio Code para eliminar pilas de CloudFormation.

Requisitos previos

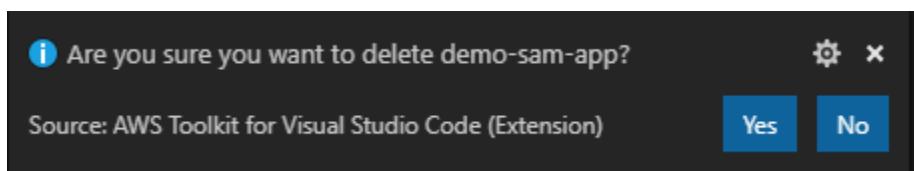
- Asegúrese de que el sistema cumple los requisitos previos especificados en [Instalación del kit de herramientas para VS Code](#).
- Asegúrese de que las credenciales configuradas en [Autenticación y acceso](#) incluyen acceso de lectura/escritura adecuado al servicio CloudFormation. Si en AWS Explorer, en CloudFormation, aparece el mensaje “Error loading CloudFormation resources” (Error al cargar recursos de CloudFormation) o un mensaje similar, verifique los permisos adjuntos a esas credenciales. Los cambios que lleve a cabo en los permisos tardarán unos minutos en afectar al Explorador de AWS en VS Code.

Eliminar una pila de CloudFormation

1. En AWS Explorer, abra el menú contextual de la pila de CloudFormation que quiere eliminar.



2. Elija Delete CloudFormation Stack (Eliminar pila de CloudFormation).
3. En el mensaje que aparece, elija Yes (Sí) para confirmar la eliminación.



Una vez que se elimine la pila, esta deja de estar en la lista de AWS Explorer.

Cree una AWS CloudFormation plantilla utilizando el AWS Toolkit for Visual Studio Code

AWS Toolkit for Visual Studio Code Pueden ayudarlo a escribir plantillas SAM AWS CloudFormation y a escribir.

Requisitos previos

Requisitos previos del kit de herramientas para VS Code y las credenciales

- Antes de poder acceder al CloudFormation servicio desde el Toolkit for VS Code, debe cumplir los requisitos descritos en la [guía del usuario Instalación del kit de herramientas](#) para VS Code.
- Las credenciales que creó [Autenticación y acceso](#) deben incluir el read/write acceso adecuado al servicio. AWS CloudFormation

 Note

Si el CloudFormation servicio muestra un mensaje de error al cargar CloudFormation los recursos, compruebe los permisos que ha adjuntado a esas credenciales. Tenga en cuenta también que los cambios realizados en los permisos pueden tardar unos minutos en actualizarse en el Explorador de AWS .

CloudFormation requisitos previos de la plantilla

- Instale y habilite la extensión [YAML VS Code de Red Hat Developer](#).
- Debe estar conectado a Internet cuando utilice la extensión YAML VS Code de Red Hat Developer, ya que sirve para descargar y utilizar esquemas JSON en su máquina.

Escribir una CloudFormation plantilla con YAML Schema Support

El kit de herramientas utiliza la compatibilidad con el lenguaje YAML y los esquemas JSON para agilizar el proceso de escritura CloudFormation y las plantillas SAM. Algunas características, como la validación de la sintaxis y la función autocompletar, no solo agilizan el proceso, sino que también ayudan a mejorar la calidad de la plantilla. Estas son las prácticas recomendadas a la hora de seleccionar un esquema para la plantilla.

CloudFormation plantilla

- El archivo tiene una extensión .yaml o .yml.
- El archivo tiene `AWSTemplateFormatVersion` en el nivel superior o un nodo `Resources`.

Plantilla de SAM

- Todos los criterios ya descritos para CloudFormation
- El archivo tiene un nodo `Transform` en el nivel superior que contiene un valor que comienza por `AWS::Serverless`.

El esquema se aplicará después de modificar el archivo. Por ejemplo, se aplicará un esquema de plantilla SAM después de añadir una transformación sin servidor a una CloudFormation plantilla y guardar el archivo.

Validación de la sintaxis

La extensión YAML aplicará automáticamente la validación de tipos a su plantilla. Resalta las entradas con tipos no válidos para una determinada propiedad. Si pasa el ratón por encima de una entrada resaltada, las extensiones muestran las acciones correctivas.

Función autocompletar

Cuando añada nuevos campos, valores enumerados u otros [tipos de recurso](#), puede iniciar la función autocompletar de la extensión YAML pulsando `Ctrl + espacio`.

Uso de Registros de CloudWatch mediante el AWS Toolkit for Visual Studio Code

Registros de Amazon CloudWatch le permite centralizar los registros de todos los sistemas, aplicaciones y servicios de AWS que utiliza, en un único servicio de gran escalabilidad. De este modo, los puede consultar fácilmente, buscar códigos de error o patrones específicos, filtrarlos en función de campos específicos o archivarlos de forma segura para análisis futuros. Para obtener más información, consulte [¿Qué es Amazon CloudWatch Logs?](#) en la Guía del usuario de Amazon CloudWatch.

En los temas siguientes se describe cómo utilizar AWS Toolkit for Visual Studio Code para trabajar con los Registros de CloudWatch en una cuenta de AWS.

Temas

- [Visualización de los grupos de registros y flujos de registro de CloudWatch mediante el AWS Toolkit for Visual Studio Code](#)
- [Uso de eventos de registros de CloudWatch en flujos de registros mediante el AWS Toolkit for Visual Studio Code](#)
- [Buscar grupos de registro de CloudWatch](#)
- [Live Tail de Registros de Amazon CloudWatch](#)

Visualización de los grupos de registros y flujos de registro de CloudWatch mediante el AWS Toolkit for Visual Studio Code

Un flujo de registro es una secuencia de eventos de registro que comparten la misma fuente. Cada fuente independiente de registros en CloudWatch Logs constituye un flujo de registro independiente.

Un grupo de registro es un grupo de flujos de registro que comparten la misma configuración de retención, monitorización y control de acceso. Puede definir grupos de registro y especificar los flujos que deben incluirse en cada uno. No hay límites en el número de flujos de registros que pueden pertenecer a un grupo de registros.

Para obtener más información, consulte [Trabajar con grupos de registros y flujos de registros](#) en la Guía del usuario de Amazon CloudWatch.

Temas

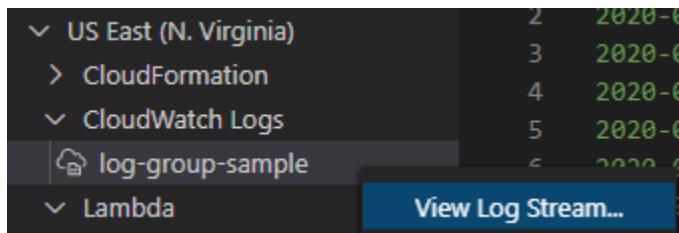
- [Visualización de grupos de registros y flujos de registro con el nodo de CloudWatch Logs](#)

Visualización de grupos de registros y flujos de registro con el nodo de CloudWatch Logs

1. En VS Code, elija Ver, Explorador para abrir el Explorador de AWS.
2. Haga clic en el botón CloudWatch Logs para expandir la lista de grupos de registros.

Los grupos de registros de la región de AWS actual se muestran en el nodo de CloudWatch Logs.

3. Para ver los flujos de registro de un grupo de registros, haga clic con el botón derecho en el nombre del grupo de registros y, a continuación, elija Ver flujo de registro.



4. En la paleta de comandos, elija un flujo de registro del grupo que desee ver.

 Note

La paleta de comandos muestra una marca de tiempo para el último evento de cada flujo.

Se inicia [el editor Flujo de registro](#) para mostrar los eventos de registro del flujo.

Uso de eventos de registros de CloudWatch en flujos de registros mediante el AWS Toolkit for Visual Studio Code

Después de abrir el editor Flujo de registro, puede acceder a los eventos de registro de cada flujo. Los eventos de registro son registros de actividades guardados por la aplicación o el recurso que se está monitoreando.

Temas

- [Visualización y copia de información de flujos de registro](#)
- [Guardar el contenido del editor de flujos de registro en un archivo local](#)

Visualización y copia de información de flujos de registro

Cuando abre un flujo de registro, el editor Flujo de registro muestra la secuencia de eventos de registro de ese flujo.

1. Para buscar un flujo de registro que desee ver, abra el editor Flujo de registro (consulte [Visualización de grupos de registros y flujos de registro de CloudWatch](#)).

Cada línea que muestra un evento tiene una marca de tiempo para mostrar cuándo se registró.

2. Puede ver y copiar información sobre los eventos del flujo mediante las siguientes opciones:

- Ver eventos por hora: para mostrar los eventos de registro más recientes y más antiguos, seleccione Load newer events (Cargar eventos más recientes) o Load older events (Cargar eventos más antiguos).

 Note

Inicialmente, el editor de flujos de registro carga un lote de las 10 000 líneas más recientes de eventos de registro o 1 MB de datos de registro (lo que sea menor). Si elige Load newer events (Cargar eventos más recientes), el editor muestra los eventos que se registraron después de cargar el último lote. Si elige Load older events (Cargar eventos más antiguos), el editor muestra un lote de los eventos que ocurrieron antes de los que se muestran actualmente.

- Copiar eventos de registro: seleccione los eventos que desea copiar, haga clic con el botón derecho y seleccione la opción Copy (Copiar) del menú.
- Copiar el nombre del flujo de registro: haga clic con el botón derecho en la pestaña del editor Flujo de registro y elija Copiar el nombre del flujo de registro.

 Note

También puede utilizar la paleta de comandos para ejecutar el comando Copiar el nombre del flujo de registro del kit de herramientas de AWS.

Guardar el contenido del editor de flujos de registro en un archivo local

Puede descargar el contenido del editor de flujos de registro de CloudWatch en un archivo log en el equipo local.

 Note

Con esta opción, guarda en un archivo solo los eventos de registro que se muestran actualmente en el editor de flujos de registro. Por ejemplo, si el tamaño total de un flujo de registro es de 5 MB y solo se cargan 2 MB en el editor, el archivo guardado también contendrá 2 MB de datos de registro solamente. Para mostrar más datos que se van a

guardar, elija Load newer events (Cargar eventos más recientes) o Load older events (Cargar eventos más antiguos) en el editor.

1. Para buscar un flujo de registro que desee copiar, abra el editor Flujo de registro (consulte [Visualización de grupos de registros y flujos de registro de CloudWatch](#)).
2. Seleccione el ícono Guardar situado junto a la pestaña que muestra el nombre del flujo de registro.

 Note

También puede utilizar la paleta de comandos para ejecutar el comando Guardar contenido de flujo de registro actual del kit de herramientas de AWS.

3. Utilice el cuadro de diálogo para seleccionar o crear una carpeta de descarga para el archivo de registros y haga clic en Save (Guardar).

Buscar grupos de registro de CloudWatch

Puede usar Buscar grupos de registro para buscar todos los flujos de registro de un grupo de registro.

Para obtener más información sobre el servicio Registros de Amazon CloudWatch, consulte el tema [Trabajar con grupos de registro y flujos de registro](#) en la guía del usuario de Amazon CloudWatch.

Buscar grupos de registro desde la paleta de comandos de VS Code

Para buscar grupos de registro desde la paleta de comandos de VS Code, complete los siguientes pasos.

Para obtener más información sobre los filtros y los patrones de los registros de Amazon CloudWatch, consulte la sección [Sintaxis de patrones y filtros](#) en la guía del usuario de Amazon CloudWatch.

1. Desde VS Code, abra la paleta de comandos presionando **cmd+shift+p** (Windows: **ctrl+shift+p**).

2. Desde la paleta de comandos, introduzca el comando **AWS: Search Log Group** y, a continuación, selecciónelo para abrir el cuadro de diálogo de búsqueda del grupo de registro en VS Code y siga las instrucciones para continuar.

 Note

En el primer cuadro de diálogo, tiene la opción de cambiar de región de AWS antes de continuar con los siguientes pasos.

3. En el cuadro de diálogo Seleccione un grupo de registro (1/3), elija el grupo de registro que desea buscar.
4. En el cuadro de diálogo Seleccione el filtro de tiempo (2/3), elija el filtro de tiempo que desee aplicar a la búsqueda.
5. En el cuadro de diálogo Buscar grupo de registro... (3/3), introduzca su patrón de búsqueda en el campo correspondiente y, a continuación, pulse la tecla **Enter** para continuar o la tecla **ESC** para cancelar la búsqueda.
6. Los resultados de la búsqueda se abren en el editor de VS Code cuando se completa la búsqueda.

Buscar grupos de registro desde el Explorador de AWS

Para buscar grupos de registro desde el Explorador de AWS Toolkit for Visual Studio Code, complete los siguientes pasos.

1. Desde el Explorador de AWS Toolkit for Visual Studio Code, expanda CloudWatch.
2. Abra el menú contextual (clic con el botón derecho) de Buscar grupo de registro donde desee buscar y, a continuación, elija Buscar grupo de registro para abrir el cuadro de diálogo de búsqueda.
3. Siga las instrucciones y seleccione un período de tiempo para continuar.
4. Cuando se le solicite, introduzca su patrón de búsqueda en el campo correspondiente y, a continuación, pulse la tecla **Enter** para continuar o la tecla **ESC** para cancelar la búsqueda.
5. Los resultados de la búsqueda se abren en el editor de VS Code cuando se completa la búsqueda.

Trabajar con los resultados de la búsqueda de registros

Después de completar correctamente una búsqueda de grupos de registros de CloudWatch, los resultados de la búsqueda se muestran en el editor de VS Code. Los siguientes procedimientos describen cómo trabajar con los resultados de la búsqueda de registros.

Note

Cuando consulte un único flujo de registro, las siguientes características se limitan a los resultados del flujo de registro actualmente activo.

Guardar los resultados de la búsqueda de grupos de registro

Para guardar los resultados de la búsqueda del grupo de registro de forma local, complete los siguientes pasos.

1. En los resultados de la búsqueda del grupo de registros, pulse el ícono Guardar registro en un archivo, situado en la esquina superior derecha del editor de VS Code.
2. En el cuadro de diálogo Guardar como, especifique el nombre y la ubicación en la que desea guardar el archivo.
3. Si selecciona Aceptar, el archivo se guarda en su equipo local.

Cambiar el intervalo de tiempo

Para cambiar el intervalo de tiempo que está activo en los resultados de la búsqueda del grupo de registro, complete los siguientes pasos.

1. En los resultados de la búsqueda del grupo de registro, pulse el ícono Buscar por fecha..., situado en la esquina superior derecha del editor de VS Code.
2. En el cuadro de diálogo Seleccione el filtro de tiempo, elija un nuevo intervalo de tiempo para los resultados de la búsqueda de registros.
3. Los resultados se actualizan cuando se cierra el cuadro de diálogo Seleccione el filtro de tiempo.

Cambiar el patrón de búsqueda

Para cambiar el patrón de búsqueda que está activo en los resultados de la búsqueda del grupo de registro, complete los siguientes pasos.

1. En los resultados de la búsqueda del grupo de registro, pulse el icono Buscar por patrón..., situado en la esquina superior derecha del editor de VS Code.
2. En el cuadro de diálogo Buscar grupo de registro, introduzca el nuevo patrón de búsqueda en el campo correspondiente.
3. Pulse la tecla **Enter** para cerrar el cuadro de diálogo y actualizar los resultados con el nuevo patrón de búsqueda.

Live Tail de Registros de Amazon CloudWatch

Transmita en directo los eventos de registro de CloudWatch a medida que se incorporan a un grupo de registro en particular con Live Tail de Registros de Amazon CloudWatch.

Para obtener más información sobre la característica Live Tail, consulte el tema [Solución de problemas con Live Tail de Registros de CloudWatch](#) en la Guía del usuario de Registros de Amazon CloudWatch.

Las sesiones de Live Tail generan costos según el tiempo de uso de la sesión por minuto. Para obtener información sobre los precios, consulte la pestaña Registros de la sección Nivel de pago de la guía [Precios de Amazon CloudWatch](#).

Inicio de una sesión de Live Tail desde la paleta de comandos de VS Code

Para iniciar una sesión de Live Tail desde la paleta de comandos de VS Code, complete los siguientes pasos.

Para obtener más información sobre los filtros y los patrones de los registros de Amazon CloudWatch, consulte la sección [Sintaxis de patrones y filtros](#) en la guía del usuario de Amazon CloudWatch.

Inicio de una sesión de seguimiento desde la paleta de comandos

1. En VS Code, pulse **cmd+shift+p** (Windows: **ctrl+shift+p**) para abrir la paleta de comandos.
2. En la paleta de comandos, escriba el comando AWS: Tail Log Group y, a continuación, selecciónelo para abrir el cuadro de diálogo Grupo de registro de Tail en VS Code y siga las instrucciones para continuar.

 Note

En la primera instrucción, tiene la opción de cambiar de región de AWS antes de continuar con los pasos siguientes.

3. En el mensaje Tail Log Group (1/3), elija el grupo de registro que desea seguir.
4. En el mensaje Include log events from... (2/3), elija el filtro de flujo de registro que desea aplicar a la sesión de seguimiento.
5. En el cuadro de diálogo Provide log event filter pattern... (3/3), especifique el patrón de filtro en el campo proporcionado y, a continuación, pulse la tecla **Enter** para continuar o la tecla **ESC** para cancelar la búsqueda.
6. Al finalizar, los resultados se transmiten al editor de VS Code.

 Note

Si una sesión de Live Tail que se ejecuta en la ventana de VS Code coincide con la configuración de un comando Grupo de registro de Tail recién enviado, no se iniciará una sesión nueva. En su lugar, la sesión existente se convierte en el editor de texto activo.

Inicio de una sesión de Live Tail desde el Explorador de AWS

Para iniciar una sesión de Live Tail desde el Explorador del kit de herramientas de AWS, complete los siguientes pasos.

Inicio de una sesión de seguimiento desde el Explorador de AWS

1. En el Explorador del kit de herramientas de AWS, expanda CloudWatch.
2. Abra el menú contextual (clic con el botón derecho) del grupo de registro que desea seguir y, a continuación, elija Grupo de registro de Tail para abrir el mensaje de seguimiento.
3. Siga las instrucciones para continuar.
4. Los resultados se transmitirán al editor de VS Code.

Detención de una sesión de Live Tail

Hay dos formas de detener una sesión de seguimiento en ejecución.

Detención de una sesión de seguimiento

1. Haga clic en el comando de CodeLens Stop tailing en la parte inferior del documento de texto de la sesión de seguimiento.
2. Cierre todos los editores que contengan el documento de texto de la sesión de seguimiento.

Amazon DocumentDB

Puede administrar los clústeres y las instancias de Amazon DocumentDB directamente en VS Code con el AWS Toolkit for Visual Studio Code. Amazon DocumentDB (compatible con MongoDB) es un servicio de base de datos rápido, fiable y completamente administrado que simplifica la configuración, el funcionamiento y el escalado de las bases de datos compatibles con MongoDB en la nube. Para obtener más información sobre el servicio Amazon DocumentDB, consulte la Guía para desarrolladores de [Amazon DocumentDB](#).

En los temas siguientes se describe cómo trabajar con Amazon DocumentDB con el AWS Toolkit for Visual Studio Code.

Temas

- [Uso de Amazon DocumentDB en el kit de herramientas](#)

Uso de Amazon DocumentDB en el kit de herramientas

Amazon DocumentDB (compatible con MongoDB) es un servicio de base de datos rápido, fiable y completamente administrado que simplifica la configuración, el funcionamiento y el escalado de las bases de datos compatibles con MongoDB en la nube.

Para obtener información detallada sobre Amazon DocumentDB, información de introducción y tutoriales, consulte la Guía para desarrolladores de [Amazon DocumentDB](#).

En las siguientes secciones se describe cómo trabajar con Amazon DocumentDB con el AWS Toolkit for Visual Studio Code.

Acceso a Amazon DocumentDB desde el kit de herramientas AWS

Para acceder a Amazon DocumentDB con el AWS kit de herramientas, complete el siguiente procedimiento.

Acceso a Amazon DocumentDB en el kit de herramientas AWS

1. Desde VS Code, abra AWS Toolkit for Visual Studio Code
2. Desde el AWS kit de herramientas, expanda el Explorador.
3. En el Explorador, expanda Amazon DocumentDB para mostrar los recursos de Amazon DocumentDB existentes.

Creación de un clúster basado en instancias.

Para empezar a trabajar con Amazon DocumentDB, complete el siguiente procedimiento a fin de crear un clúster.

Creación de un clúster basado en instancias

1. En AWS Toolkit for Visual Studio Code, abra el menú contextual de Amazon DocumentDB (haga clic con el botón derecho) y, a continuación, seleccione Crear clúster para abrir el cuadro de diálogo Crear clúster de Amazon DocumentDB en VS Code.
2. En la pantalla Tipo de clúster, elija Clúster basado en instancias.
3. En la pantalla Nombre del clúster, especifique un nombre para el nuevo clúster.
4. En la pantalla Seleccione la versión del motor, elija la versión del motor de Amazon DocumentDB que prefiera.
5. En las pantallas Nombre de usuario y contraseña del administrador, especifique un nombre de usuario y una contraseña del administrador para proteger el clúster.
6. En la pantalla Especificar cifrado de almacenamiento, elija si desea cifrar el clúster o no.
7. En la pantalla Número de instancias, configure el número de instancias que prefiera.
8. En la pantalla Seleccionar clase de instancia, elija la clase de instancia que prefiera y, a continuación, cree el nuevo clúster.

 Note

Es posible que el clúster tarde varios minutos en activarse.

Copia de un punto de conexión de un clúster

Para copiar el punto de conexión de su clúster de Amazon DocumentDB, realice el procedimiento siguiente:

Copia de un punto de conexión de un clúster

1. Desde AWS Toolkit for Visual Studio Code, expanda Amazon DocumentDB para mostrar sus clústeres de Amazon DocumentDB.
2. Haga clic con el botón derecho en el clúster del que desea copiar los detalles de conexión y, a continuación, elija Copiar punto de conexión para copiar la información del punto de conexión del clúster en el portapapeles.
3. Ahora puede pegar el punto de conexión del clúster en sus documentos.

Apertura en el navegador

Abra sus clústeres de Amazon DocumentDB en la AWS consola para obtener más funciones de administración de clústeres. Para abrir la AWS consola en su clúster de Amazon DocumentDB en su navegador web predeterminado, complete el siguiente procedimiento.

Abrir el clúster en la consola AWS

1. Desde AWS Toolkit for Visual Studio Code, expanda Amazon DocumentDB para mostrar sus clústeres de Amazon DocumentDB.
2. Haga clic con el botón derecho en el clúster que desee ver en la AWS consola y, a continuación, seleccione Abrir en el navegador.
3. La AWS consola se abre en el clúster de Amazon DocumentDB en su navegador web predeterminado.

Ampliación de un clúster existente

Para escalar los clústeres de Amazon DocumentDB mediante la adición de instancias, realice el procedimiento siguiente:

Adición de una instancia para expandir el clúster

1. Desde AWS Toolkit for Visual Studio Code, expanda Amazon DocumentDB para mostrar sus clústeres de Amazon DocumentDB.

2. Haga clic con el botón derecho en el clúster que desea ampliar y elija Agregar una instancia para abrir el cuadro de diálogo Agregar una instancia en VS Code.
3. Cuando se le solicite, ingrese un nombre para la nueva instancia en el campo de texto y, a continuación, pulse la tecla **Enter** para continuar.
4. Cuando se le solicite, seleccione una clase de instancia de la lista para continuar.
5. El Explorador de AWS muestra el estado de creación y se actualiza cuando la nueva instancia está lista.

Detención de un clúster

Para detener el clúster de Amazon DocumentDB, realice el procedimiento siguiente:

 Note

Mientras el clúster esté detenido, la mayoría de las características de administración de clústeres no estarán disponibles.

Detención del clúster de Amazon DocumentDB

1. Desde AWS Toolkit for Visual Studio Code, expanda Amazon DocumentDB para mostrar sus clústeres de Amazon DocumentDB.
2. Elija el botón Detener clúster situado junto al clúster que desea detener o haga clic con el botón derecho en el clúster y elija Detener clúster.
3. Cuando se le solicite, elija Sí para detener el clúster o Cancelar a fin de cancelar el proceso de detención y dejar el clúster en ejecución.
4. El Explorador de AWS muestra el estado del clúster y se actualiza cuando este se ha detenido.

Reinicio de una instancia

Reiniciar una instancia es útil para solucionar problemas y realizar cambios menores sin que afecte a todo el clúster. Para reiniciar una instancia de Amazon DocumentDB, realice el procedimiento siguiente:

Reinicio de una instancia del clúster

1. Desde AWS Toolkit for Visual Studio Code, expanda Amazon DocumentDB para mostrar sus clústeres de Amazon DocumentDB.
2. Haga clic con el botón derecho en la instancia del clúster que desea reiniciar y, a continuación, elija Reiniciar instancia.
3. Cuando se le solicite, elija Sí para reiniciar la instancia o Cancelar a fin de cancelar el proceso de reinicio y dejar la instancia detenida.
4. El Explorador de AWS muestra el estado del clúster y se actualiza cuando la instancia se ha reiniciado.

Eliminación de una instancia

Para eliminar una instancia del clúster de Amazon DocumentDB, realice el procedimiento siguiente.

 Note

Eliminar una instancia no afecta a los datos del clúster. Si elimina la instancia principal, una de las instancias de réplica toma el relevo como instancia de escritura.

Eliminación de una instancia del clúster

1. Desde AWS Toolkit for Visual Studio Code, expanda Amazon DocumentDB para mostrar sus clústeres de Amazon DocumentDB.
2. Haga clic con el botón derecho en la instancia de clúster que desee eliminar y, a continuación, seleccione Eliminar para abrir el cuadro de diálogo de delete-cluster-instance confirmación en VS Code.
3. Siga las instrucciones de confirmación y, a continuación, pulse la tecla **Enter** para eliminar su instancia del clúster.
4. El Explorador de AWS muestra el estado de la instancia del clúster y se actualiza cuando la instancia se ha eliminado.

Visualización, adición y eliminación de etiquetas

Las etiquetas se utilizan con el fin de organizar los recursos de su entorno y realizar un seguimiento de estos. Para ver o editar las etiquetas asociadas a su clúster de Amazon DocumentDB, realice uno de los procedimientos siguientes:

Visualización de etiquetas de clúster

1. Desde AWS Toolkit for Visual Studio Code, expanda Amazon DocumentDB para mostrar sus clústeres de Amazon DocumentDB.
2. Haga clic con el botón derecho en el clúster para el que desea ver las etiquetas y, a continuación, elija Etiquetas... a fin de abrir el cuadro de diálogo Etiquetas para **your cluster name**.
3. Las etiquetas aparecen en el cuadro de diálogo. Si no hay ninguna etiqueta asociada al clúster, se muestra el mensaje [No tags assigned].

Adición de etiquetas al clúster

1. Desde AWS Toolkit for Visual Studio Code, expanda Amazon DocumentDB para mostrar sus clústeres de Amazon DocumentDB.
2. Haga clic con el botón derecho en el clúster para el que desea agregar etiquetas y, a continuación, elija Etiquetas... a fin de abrir el cuadro de diálogo Etiquetas para **your cluster name**.
3. Elija el botón Agregar etiqueta... para abrir el cuadro de diálogo Add Tag en VS Code.
4. Ingrese una nueva etiqueta en el campo de texto y, a continuación, pulse la tecla Intro para continuar.
5. Introduzca un valor en el campo de texto y, a continuación, pulse Entrar para añadir el key/value par al clúster.

Eliminación de etiquetas del clúster

1. Desde AWS Toolkit for Visual Studio Code, expanda Amazon DocumentDB para mostrar sus clústeres de Amazon DocumentDB.
2. Haga clic con el botón derecho en el clúster del que desea eliminar etiquetas y, a continuación, elija Etiquetas... a fin de abrir el cuadro de diálogo Etiquetas para **your cluster name**.

3. Elija el botón Eliminar etiqueta... para abrir el cuadro de diálogo Remove a tag from **your cluster name** en VS Code.
4. Elija la etiqueta que desea eliminar en la lista proporcionada para eliminar la etiqueta del clúster.

Modificación de una clase de instancia

Para modificar la clase de instancia del clúster de Amazon DocumentDB, realice el procedimiento siguiente:

Modificación de una clase de instancia

1. Desde AWS Toolkit for Visual Studio Code, expanda Amazon DocumentDB para mostrar sus clústeres de Amazon DocumentDB.
2. Haga clic con el botón derecho en la instancia del clúster que desea modificar y, a continuación, elija Modificar clase... para abrir el cuadro de diálogo Select instance class en VS Code.
3. Elija una nueva clase de instancia en la lista para actualizar la clase.
4. El Explorador de AWS muestra el estado de la instancia del clúster y se actualiza cuando la clase de la instancia se ha actualizado.

Amazon Elastic Compute Cloud

Amazon Elastic Compute Cloud para el AWS Toolkit for Visual Studio Code permite lanzar sus instancias de Amazon EC2 y conectarse a ellas desde VS Code. Para obtener información detallada acerca de Amazon EC2, consulte el tema [¿Qué es Amazon EC2?](#) de la Guía del usuario de Amazon Elastic Compute Cloud.

Los siguientes temas describen cómo trabajar con el Desarrollador de aplicaciones de AWS desde el AWS Toolkit for Visual Studio Code.

Temas

- [Uso de Amazon Elastic Compute Cloud](#)
- [Solución de problemas de Amazon Elastic Compute Cloud](#)

Uso de Amazon Elastic Compute Cloud

En las secciones siguientes se describe cómo trabajar con Amazon Elastic Compute Cloud en el AWS Toolkit for Visual Studio Code.

Requisitos previos

Las funciones descritas en este tema de la guía del usuario se han probado en EC2 instancias de Amazon con los siguientes sistemas operativos:

- Windows 2016 y versiones posteriores

 Note

Este sistema operativo solo funciona cuando se conecta un terminal de VS Code. No funciona cuando la conexión se realiza mediante una instancia remota de VS Code completa. Para obtener más información sobre las instancias remotas y los terminales de VS Code, consulte los temas [Getting started with the terminal](#) y [VS Code Remote Development](#) de la documentación de VS Code.

- Amazon Linux 2023
- Ubuntu, 22.04

Se requiere un SSH instalado localmente para abrir una conexión remota a una EC2 instancia de Amazon, pero no es necesario para abrir un terminal a una EC2 instancia de Amazon.

El perfil de tu EC2 instancia de Amazon debe incluir los siguientes permisos AWS Identity and Access Management (IAM).

```
"ssmmessages:CreateControlChannel",
"ssmmessages:CreateDataChannel",
"ssmmessages:OpenControlChannel",
"ssmmessages:OpenDataChannel",
"ssm:DescribeAssociation",
"ssm>ListAssociations",
"ssm:UpdateInstanceInformation"
```

Note

Los permisos necesarios se incluyen en la siguiente política AWS gestionada.

- AmazonSSMManagedInstanceCore
- AmazonSSMManagedEC2InstanceDefaultPolicy

Visualización de EC2 instancias de Amazon existentes

Para ver tus EC2 instancias de Amazon existentes desde el AWS kit de herramientas, sigue estos pasos.

1. Desde el AWS kit de herramientas, expande el Explorador del kit de AWS herramientas.
2. Amplía la región que contiene las EC2 instancias de Amazon que quieras ver.
3. Amplía el EC2encabezado para mostrar tus EC2 instancias de Amazon existentes.

Lanzamiento de una nueva EC2 instancia de Amazon

Hay tres formas de crear una nueva EC2 instancia de Amazon con el AWS kit de herramientas.

Cada flujo de trabajo abre el asistente de inicialización de instancias en la consola de AWS . Para obtener información detallada sobre el lanzamiento de una nueva EC2 instancia de Amazon desde el asistente Launch an instance, consulte el tema [Lanzar una EC2 instancia mediante el asistente de lanzamiento de instancias en la consola](#) de la Guía del usuario de Amazon Elastic Compute Cloud. Para lanzar una nueva EC2 instancia de Amazon, complete uno de los siguientes procedimientos.

Lanzamiento de una nueva EC2 instancia de Amazon desde la paleta de comandos de VS Code

1. En VS Code, pulse **command + shift + P** (**Windows: ctrl + shift + P**) para abrir la paleta de comandos de VS Code.
2. En la paleta de comandos de VS Code, busca el **AWS: Launch EC2** comando y selecciónalo cuando aparezca en la lista para abrir la ventana de selección de región de la EC2 instancia de lanzamiento en VS Code.
3. En la ventana de selección de la EC2 instancia de lanzamiento, selecciona la región en la que quieras lanzar la nueva instancia y, a continuación, confirma que quieres abrir la AWS consola en tu navegador web predeterminado.

4. Desde la AWS consola del navegador web predeterminado, complete el proceso de autenticación para continuar con el asistente de lanzamiento de una instancia.
5. En el asistente Launch an instance, complete las secciones obligatorias y, a continuación, pulse el botón Launch instance para lanzar la nueva EC2 instancia de Amazon.
6. El AWS explorador se actualiza para mostrar tu nueva EC2 instancia de Amazon.

Lanzamiento de una nueva EC2 instancia de Amazon desde el AWS Explorador

1. Amplía el AWS Toolkit Explorer y, a continuación, expande la región en la que quieras crear la nueva EC2 instancia de Amazon.
2. Expande o coloca el cursor sobre el EC2encabezado y, a continuación, selecciona el icono + (Lanzar EC2 instancia).
3. Cuando se le solicite, confirme que desea abrir la AWS consola en su navegador web predeterminado.
4. Desde la AWS consola del navegador web, complete el proceso de autenticación para continuar con el asistente de lanzamiento de una instancia.
5. En el asistente Launch an instance, complete las secciones obligatorias y, a continuación, pulse el botón Launch instance para lanzar la nueva EC2 instancia de Amazon.
6. El AWS explorador se actualiza para mostrar tu nueva EC2 instancia de Amazon.

Lanzar una nueva EC2 instancia de Amazon desde el menú contextual (clic con el botón derecho)

1. Amplía el AWS Toolkit Explorer y, a continuación, expande la región en la que quieras crear la nueva EC2 instancia de Amazon.
2. Haga clic con el botón derecho EC2en el encabezado y, a continuación, seleccione Lanzar EC2 instancia.
3. Cuando se le solicite, confirme que desea abrir la AWS consola en su navegador web predeterminado.
4. Desde la AWS consola del navegador web, complete el proceso de autenticación para continuar con el asistente de lanzamiento de una instancia.
5. En el asistente Launch an instance, complete las secciones obligatorias y, a continuación, pulse el botón Launch instance para lanzar la nueva EC2 instancia de Amazon.
6. El AWS explorador se actualiza para mostrar tu nueva EC2 instancia de Amazon.

Conexión de VS Code a una EC2 instancia de Amazon

Hay 3 formas de conectarse a una EC2 instancia de Amazon desde VS Code. Para conectar VS Code a su EC2 instancia, complete uno de los siguientes procedimientos.

Conexión de VS Code a una EC2 instancia de Amazon desde la paleta de comandos

1. En VS Code, pulse **command + shift + P** (**Windows: ctrl + shift + P**) para abrir la paleta de comandos de VS Code.
2. En la paleta de comandos de VS Code, busque el **AWS: Connect VS Code to EC2 instance...** comando y selecciónelo cuando aparezca en la lista para abrir la ventana de selección de EC2 instancias en VS Code.
3. En la ventana de selección de EC2 instancias, elige la región que contiene la instancia a la que quieras conectarte y, a continuación, elige la instancia a la que quieras conectarte.
4. VS Code muestra el estado mientras se establece la conexión.
5. Se abrirá una nueva ventana para mostrar tu EC2 instancia de Amazon cuando se complete la conexión.

Conexión de VS Code a una EC2 instancia de Amazon desde el AWS Explorador.

1. Expanda el AWS Toolkit Explorer y, a continuación, expanda la región que contiene la EC2 instancia de Amazon a la que desea conectarse.
2. Coloca el cursor sobre la EC2 instancia de Amazon y, a continuación, selecciona el ícono (Connect VS Code to EC2 instance).

 Note

También puede elegir el ícono (Conectar código VS a la EC2 instancia) en el encabezado del EC2servicio en el AWS Explorador.

3. VS Code muestra el estado mientras se establece la conexión.
4. Se abrirá una nueva ventana para mostrar tu EC2 instancia de Amazon cuando se complete la conexión.

Conexión de VS Code a una EC2 instancia de Amazon desde el menú contextual

1. Expanda el AWS Toolkit Explorer y, a continuación, expanda la región que contiene la EC2 instancia de Amazon a la que desea conectarse.
2. Haz clic con el botón derecho en la EC2 instancia de Amazon a la que quieras conectarte y, a continuación, selecciona Connect VS Code to EC2 instance.

Note

También puedes hacer clic con el botón derecho en el encabezado del EC2 servicio en el AWS Explorador y elegir Connect VS Code to EC2 Instance.

3. VS Code muestra el estado mientras se establece la conexión.
4. Se abrirá una nueva ventana para mostrar tu EC2 instancia de Amazon cuando se complete la conexión.

Abrir un terminal a una EC2 instancia de Amazon.

Hay tres formas de conectarse a una EC2 instancia de Amazon desde el terminal VS Code.

Conexión de VS Code a una EC2 instancia de Amazon desde la paleta de comandos

1. En VS Code, pulse **command + shift + P** (**Windows: ctrl + shift + P**) para abrir la paleta de comandos de VS Code.
2. En la paleta de comandos de VS Code, busque el **AWS:Open terminal to EC2 instance...** comando y selecciónelo cuando aparezca en la lista para abrir la ventana de selección de EC2 instancias en VS Code.
3. En la solicitud Select EC2 Instance, elija la región que contiene la instancia que desea abrir en la terminal y, a continuación, elija la instancia.
4. VS Code muestra el estado mientras se establece la conexión.
5. Una vez completada la conexión, se abrirá el terminal de VS Code para mostrar su nueva sesión.

Abrir una EC2 instancia de Amazon en la terminal de VS Code desde el AWS Explorador.

1. Expanda el AWS Toolkit Explorer y, a continuación, expanda la región que contiene la EC2 instancia de Amazon a la que desea conectarse.

2. Coloca el cursor sobre la EC2 instancia de Amazon y, a continuación, selecciona (Abrir terminal a EC2 instancia...) ícono.

 Note

También puede elegir el (Abrir terminal a EC2 instancia...) ícono del encabezado del EC2servicio en el AWS Explorador.

3. VS Code muestra el estado mientras se establece la conexión.
4. Una vez completada la conexión, se abrirá el terminal de VS Code para mostrar su nueva sesión.

Abrir una EC2 instancia de Amazon en la terminal de VS Code desde el menú contextual

1. Expanda el AWS Toolkit Explorer y, a continuación, expanda la región que contiene la EC2 instancia de Amazon que quiere abrir en la terminal de VS Code.
2. Haz clic con el botón derecho en la EC2 instancia de Amazon que quieras abrir en la terminal y, a continuación, selecciona Abrir terminal a EC2 instancia... .

 Note

También puedes hacer clic con el botón derecho en el encabezado del EC2servicio en el AWS Explorador y elegir Abrir terminal para crear una EC2 instancia... .

3. VS Code muestra el estado mientras se establece la conexión.
4. Una vez completada la conexión, se abrirá el terminal de VS Code para mostrar su nueva sesión.

Iniciar o reiniciar una instancia de Amazon EC2

Hay tres formas de iniciar o reiniciar una EC2 instancia de Amazon.

Reiniciar una EC2 instancia de Amazon desde la paleta de comandos

1. En VS Code, pulse **command + shift + P** (**Windows: ctrl + shift + P**) para abrir la paleta de comandos de VS Code.

2. En la paleta de comandos de VS Code, busque el **AWS: Reboot EC2 instance** comando y selecciónelo cuando aparezca en la lista para abrir la ventana de selección de EC2 instancias en VS Code.

 Note

Para iniciar una instancia que no se está ejecutando, debe elegir el comando **AWS: Start EC2 instance**. El comando **AWS: Reboot EC2 instance** solo reinicia las instancias que están en ejecución.

3. En la línea de comandos Select EC2 Instance, selecciona la región que contiene la instancia que deseas iniciar o reiniciar.
4. VS Code muestra el estado mientras se reinicia la instancia.
5. El AWS explorador se actualiza para mostrar que la instancia se está ejecutando cuando ha terminado de reiniciarse.

Iniciar o reiniciar una EC2 instancia de Amazon desde el Explorador AWS

1. Expanda el AWS Toolkit Explorer y, a continuación, expanda la región que contiene la EC2 instancia de Amazon que desea iniciar o reiniciar.
2. Coloca el cursor sobre la EC2 instancia de Amazon y, a continuación, selecciona el icono (Reiniciar EC2 instancia).

 Note

Si la instancia está detenida, la única opción es el icono (Iniciar EC2 instancia)

3. VS Code muestra el estado mientras se reinicia la instancia.
4. El AWS explorador se actualiza para mostrar que la instancia se está ejecutando cuando ha terminado de reiniciarse.

Iniciar o reiniciar una EC2 instancia de Amazon desde el menú contextual

1. Expanda el AWS Toolkit Explorer y, a continuación, expanda la región que contiene la EC2 instancia de Amazon que desea iniciar o reiniciar.

2. Haz clic con el botón derecho en la EC2 instancia de Amazon a la que quieras conectarte y, a continuación, selecciona Reiniciar EC2 instancia.

 Note

Si la instancia está detenida, la única opción es iniciar la EC2 instancia.

3. VS Code muestra el estado mientras se reinicia la instancia.
4. El AWS explorador se actualiza para mostrar que la instancia se está ejecutando cuando ha terminado de reiniciarse.

Detener una EC2 instancia de Amazon

Hay tres formas de detener una EC2 instancia de Amazon.

Detener una EC2 instancia de Amazon desde la paleta de comandos

1. En VS Code, pulse **command + shift + P** (**Windows: ctrl + shift + P**) para abrir la paleta de comandos de VS Code.
2. En la paleta de comandos de VS Code, busque el **AWS: Stop EC2 instance** comando y selecciónelo cuando aparezca en la lista para abrir la ventana de selección de EC2 instancias en VS Code.
3. En la solicitud de selección de EC2 instancias, elija la región que contiene la instancia que desea detener.
4. VS Code muestra el estado mientras se detiene la instancia.
5. El AWS explorador se actualiza para mostrar que la instancia está detenida.

Detener una EC2 instancia de Amazon desde el AWS explorador

1. Expanda el AWS Toolkit Explorer y, a continuación, expanda la región que contiene la EC2 instancia de Amazon que quiere detener.
2. Coloca el cursor sobre la EC2 instancia de Amazon y, a continuación, selecciona el ícono (Detener EC2 instancia).
3. VS Code muestra el estado mientras se detiene la instancia.
4. El AWS explorador se actualiza para mostrar que la instancia se ha detenido.

Detener una EC2 instancia de Amazon desde el menú contextual

1. Expanda el AWS Toolkit Explorer y, a continuación, expanda la región que contiene la EC2 instancia de Amazon que quiere detener.
2. Haz clic con el botón derecho en la EC2 instancia de Amazon a la que quieras conectarte y, a continuación, selecciona Reiniciar EC2 instancia.
3. VS Code muestra el estado mientras se detiene la instancia.
4. El AWS explorador se actualiza para mostrar que la instancia se ha detenido.

Copia de un ID de instancia

Para copiar un ID de instancia, complete los siguientes pasos:

1. Haga clic con el botón derecho en la instancia de la que desea copiar el ID.
2. Elija Copiar ID de instancia.
3. El ID de la instancia se copia en el portapapeles local.

Copia de un nombre

Para copiar un nombre de instancia, complete los siguientes pasos:

1. Haga clic con el botón derecho en la instancia de la que desea copiar el nombre.
2. Elija Copiar nombre de instancia.
3. El nombre de la instancia se copia en el portapapeles local.

Copia de un ARN

Para copiar un nombre de recurso de Amazon (ARN) de una instancia, complete los siguientes pasos:

1. Haga clic con el botón derecho en la instancia de la que desea copiar el ARN.
2. Elija Copiar el ARN de la instancia.
3. El ARN de la instancia se copia en el portapapeles local.

Solución de problemas de Amazon Elastic Compute Cloud

En las secciones siguientes se describe cómo solucionar problemas conocidos que se pueden producir al trabajar con Amazon Elastic Compute Cloud en el AWS Toolkit for Visual Studio Code. Para obtener información detallada sobre la solución de problemas específicos del EC2 servicio de Amazon, consulta el tema [Solución de problemas con las EC2 instancias de Amazon](#) en la Guía del usuario de Amazon Elastic Compute Cloud.

Depuración general

Si tienes un problema de conexión remota por cualquier motivo, empieza por comprobar si se puede establecer una AWS Systems Manager conexión desde la AWS consola.

Para conectarse a una EC2 instancia de Amazon a través de Systems Manager desde la AWS consola, complete los siguientes pasos.

1. En su navegador web, vaya a la [consola de AWS](#).
2. Complete la autenticación para acceder a la AWS EC2 consola.
3. En el panel de EC2 navegación de Amazon, selecciona Instances.
4. Seleccione el cuadro situado junto a la instancia a la que desea conectarse.
5. Elija el botón Conectar para abrir la pantalla Conectarse a la instancia en una nueva pestaña del navegador.

 Note

Solo puede conectarse a una instancia si está en ejecución. Si no puede seleccionar el botón Conectar, asegúrese de que la instancia se esté ejecutando.

6. En la pantalla Conectarse a la instancia, elija la pestaña Administrador de sesiones y, a continuación, elija el botón Conectar para abrir la conexión de Systems Manager en la pestaña del navegador actual.

 Note

Si ha iniciado la instancia recientemente y no está disponible la opción para establecer una conexión con Systems Manager, puede que tenga que esperar unos minutos más antes de que la opción esté disponible.

La instancia de destino no se está ejecutando

Para conectarse a una EC2 instancia de Amazon desde el terminal o una conexión remota, la instancia debe estar ejecutándose. Antes de intentar conectarte a la instancia desde el AWS kit de herramientas, iníciala desde el AWS Explorador Consola de administración de AWS, o AWS Command Line Interface.

La instancia de destino no tiene un rol de IAM o tiene un rol de IAM con permisos inadecuados.

Para conectarse a tu EC2 instancia de Amazon, debe tener un rol de IAM con los permisos correctos adjuntos. Si intenta conectarse a una instancia que no tiene un rol de IAM asociado, VS Code se lo notificará.

Si intenta conectarse a una instancia que tiene un rol de IAM pero carece de los permisos necesarios, se le pedirá que agregue las acciones mínimas requeridas como política insertada al rol de IAM existente. Tras actualizar la política insertada, se conecta a su instancia. Para obtener información detallada sobre las funciones de IAM, los permisos y la asociación de una función a una instancia, consulte el EC2 tema Funciones de [IAM para Amazon](#) en la Guía del usuario de Amazon Elastic Compute Cloud y el tema [Paso 2: Verificar o añadir permisos de instancia para Session Manager](#) en la Guía del usuario de AWS Systems Manager.

El ejemplo siguiente contiene las acciones mínimas necesarias.

```
"ssmmessages:CreateControlChannel",
"ssmmessages:CreateDataChannel",
"ssmmessages:OpenControlChannel",
"ssmmessages:OpenDataChannel",
"ssm:DescribeAssociation",
"ssm>ListAssociations",
"ssm:UpdateInstanceInformation
```

Note

Los permisos necesarios se incluyen en la siguiente AWS política administrada.

- [AmazonSSMManagedEC2InstanceDefaultPolicy](#)

- `AmazonSSMManagedInstanceCore`

La instancia de destino no tiene un agente de Systems Manager en ejecución.

Este problema puede producirse por varios motivos distintos. Para solucionar el problema, comience por reiniciar la instancia y realizar otro intento de conexión. Como alternativa, establezca una conexión inicial de forma manual mediante un método de conexión distinto de SSM. Para obtener información más detallada sobre Systems Manager, consulte el tema [Uso de SSM Agent](#) de AWS Systems Manager.

Al arrancar, el EC2 estado de Amazon indica que se está ejecutando, pero las conexiones no se están realizando

Si ha iniciado o creado recientemente un nuevo rol de IAM para una instancia y no puede establecer una conexión, espere unos minutos más antes de volver a intentar establecerla.

Uso del servicio Amazon Elastic Container Registry

Amazon Elastic Container Registry (Amazon ECR) es un servicio de registro de contenedores administrado por AWS que es seguro y escalable. Varias funciones del servicio de Amazon ECR son accesibles desde el Explorador del kit de herramientas para VS Code.

- Creación de un repositorio.
- Creación de un servicio de AWS App Runner para su repositorio o imagen etiquetada.
- Acceso a los URI o ARN de etiqueta y repositorio de imágenes.
- Eliminación de etiquetas y repositorios de imágenes.

También puede acceder a la gama completa de funciones de Amazon ECR a través de la consola de VS Code si integra la CLI de AWS y otras plataformas con VS Code.

Para obtener más información acerca de Amazon ECR, consulte [¿Qué es Amazon ECR?](#) en la Guía del usuario de Amazon Elastic Container Registry.

Temas

- [Uso del servicio Amazon Elastic Container Registry](#)

- [Creación de un servicio App Runner a través de Amazon ECR](#)

Uso del servicio Amazon Elastic Container Registry

Puede acceder al servicio Amazon Elastic Container Registry (Amazon ECR) directamente desde AWS el explorador en VS Code y utilizarlo para insertar una imagen de programa en un repositorio de Amazon ECR. Para comenzar, debe seguir estos pasos:

1. Cree un Dockerfile que contenga la información necesaria para crear una imagen.
2. Cree una imagen a partir de ese Dockerfile y etiquétala para procesarla.
3. Cree un repositorio que esté dentro de su instancia de Amazon ECR.
4. Inserte la imagen etiquetada en el repositorio.

Requisitos previos

Debe completar estos pasos para tener acceso al servicio de Amazon ECR desde el Explorador de VS Code.

Creación de un usuario de IAM

Antes de poder acceder a un AWS servicio, como Amazon ECR, debe proporcionar las credenciales. Esto permite al servicio determinar si tiene permisos para obtener acceso a sus recursos. No le recomendamos que acceda AWS directamente a través de las credenciales de su AWS cuenta raíz. En su lugar, utilice AWS Identity and Access Management (IAM) para crear un usuario de IAM y, a continuación, agréguelo a un grupo de IAM con permisos administrativos. A continuación, puede acceder AWS mediante una URL especial y las credenciales del usuario de IAM.

Si te has registrado AWS pero no has creado un usuario de IAM para ti, puedes crear uno mediante la consola de IAM.

Para crear un usuario administrador, elija una de las siguientes opciones.

Elegir una forma de administrar el administrador	Para	Haga esto	También puede
En IAM Identity Center (recomendado)	Usar credenciales a corto plazo para acceder a AWS. Esto se ajusta a las prácticas recomendadas de seguridad. Para obtener información sobre las prácticas recomendadas, consulta Prácticas recomendadas de seguridad en IAM en la Guía del usuario de IAM.	Siga las instrucciones en Introducción en la Guía del usuario de AWS IAM Identity Center .	Configure el acceso mediante programación configurando el AWS CLI que se utilizará AWS IAM Identity Center en la Guía del AWS Command Line Interface usuario.
En IAM (no recomendado)	Usar credenciales a largo plazo para acceder a AWS.	Siguiendo las instrucciones de Crear un usuario de IAM para acceso de emergencia de la Guía del usuario de IAM.	Configure el acceso programático mediante Administrar las claves de acceso de los usuarios de IAM en la Guía del usuario de IAM.

Para iniciar sesión como este nuevo usuario de IAM, cierre sesión en la AWS consola y, a continuación, utilice la siguiente URL. En la siguiente URL, donde `your_aws_account_id` es su número de AWS cuenta sin guiones (por ejemplo, si su número de cuenta es, su AWS ID de cuenta es): `1234-5678-9012 AWS 123456789012`

[https://*your_aws_account_id*.signin.aws.amazon.com/console/](https://your_aws_account_id.signin.aws.amazon.com/console/)

Escriba el nombre y la contraseña del usuario de IAM que acaba de crear. Cuando haya iniciado sesión, en la barra de navegación se mostrará “su_nombre_de_usuario @ su_id_de_cuenta_de_aws”.

Si no quieras que la URL de tu página de inicio de sesión contenga tu AWS ID de cuenta, puedes crear un alias de cuenta. En el panel de IAM, elija Personalizar e ingrese un Alias de cuenta. Puede ser el nombre de su empresa. Para obtener más información, consulta [tu ID de AWS cuenta y su alias](#) en la Guía del usuario de IAM.

Para iniciar sesión después de crear un alias de cuenta, use la siguiente dirección URL:

[https://*your_account_alias*.signin.aws.amazon.com/console/](https://your_account_alias.signin.aws.amazon.com/console/)

Para verificar el enlace de inicio de sesión de los usuarios de IAM de su cuenta, abra la consola de IAM y compruebe el valor de IAM users sign-in link (Enlace de inicio de sesión de los usuarios de IAM) en el panel.

Para obtener más información acerca de IAM, consulte la [Guía del usuario de AWS Identity and Access Management](#).

Instalar y configurar Docker

Para instalar y configurar Docker, seleccione su sistema operativo preferido en la guía del usuario [Instalar Docker Engine](#) y siga las instrucciones.

Instalación y configuración de la AWS CLI versión 2

Instale y configure la versión 2 de AWS CLI seleccionando su sistema operativo preferido en la guía del usuario [Instalación, actualización y desinstalación de la AWS CLI versión 2](#).

1. Creación de un Dockerfile

Docker usa un archivo llamado Dockerfile para definir una imagen que se puede insertar y almacenar en un repositorio remoto. Para poder subir una imagen a un repositorio de ECR, debe crear un Dockerfile y, a continuación, crear una imagen a partir de ese Dockerfile.

Creación de un Dockerfile

1. Use el explorador del kit de herramientas para VS Code para navegar hasta el directorio en el que desee almacenar su Dockerfile.
2. Cree un nuevo archivo que se llame Dockerfile.

 Note

VS Code puede solicitarle que seleccione un tipo o una extensión de archivo. Si se le solicita, seleccione plaintext. VS Code tiene una extensión "dockerfile". Sin embargo, no recomendamos usarla. Esto se debe a que la extensión puede provocar conflictos con determinadas versiones de Docker u otras aplicaciones asociadas.

Edición del Dockerfile con VS Code

Si el Dockerfile tiene una extensión de archivo, abra el menú contextual (haga clic con el botón derecho) del archivo y elimine la extensión de archivo.

Después de eliminar la extensión de archivo del Dockerfile:

1. Abra el Dockerfile vacío directamente en VS Code.
2. Copie el contenido del siguiente ejemplo en su Dockerfile:

Example Plantilla de imagen de Dockerfile

```
FROM ubuntu:18.04

# Install dependencies
RUN apt-get update && \
    apt-get -y install apache2

# Install apache and write hello world message
RUN echo 'Hello World!' > /var/www/html/index.html

# Configure apache
RUN echo '. /etc/apache2/envvars' > /root/run_apache.sh && \
    echo 'mkdir -p /var/run/apache2' >> /root/run_apache.sh && \
    echo 'mkdir -p /var/lock/apache2' >> /root/run_apache.sh && \
    echo '/usr/sbin/apache2 -D FOREGROUND' >> /root/run_apache.sh && \
```

```
chmod 755 /root/run_apache.sh  
  
EXPOSE 80  
  
CMD /root/run_apache.sh
```

Se trata de un archivo Dockerfile que utiliza una imagen Ubuntu 18.04. Las instrucciones RUN actualizan las memorias caché de paquete. Instale algunos paquetes de software para el servidor web y, a continuación, escriba el contenido "Hello World!" en la raíz de documentos del servidor web. La instrucción EXPOSE expone el puerto 80 en el contenedor y la instrucción CMD inicia el servidor web.

3. Guarde el archivo Dockerfile.

Important

Asegúrese de que el Dockerfile no tenga una extensión asociada al nombre. Un Dockerfile con extensiones podría provocar conflictos con determinadas versiones de Docker u otras aplicaciones asociadas.

2. Creación de la imagen a partir del Dockerfile

El Dockerfile que ha creado contiene la información necesaria para crear una imagen para un programa. Para poder insertar esa imagen en su instancia de Amazon ECR, primero debe crear la imagen.

Creación de una imagen a partir del Dockerfile

1. Utilice la CLI de Docker o una CLI que esté integrada con su instancia de Docker para navegar al directorio que contenga su Dockerfile.
2. Ejecute el comando Docker build para crear la imagen que está definida en su Dockerfile.

```
docker build -t hello-world .
```

3. Ejecute el comando Docker images para comprobar que la imagen se ha creado correctamente.

```
docker images --filter reference=hello-world
```

Example Ejemplo de resultados:

REPOSITORY	TAG	IMAGE ID	CREATED
hello-world	latest	e9ffedc8c286	4 minutes ago
241MB			

4.

 Note

Este paso no es necesario para crear o insertar su imagen, pero puede ver cómo funciona la imagen del programa cuando se ejecuta.

Para ejecutar la imagen recién creada, utilice el comando Docker run.

```
docker run -t -i -p 80:80 hello-world
```

La opción `-p` que se ha especificado en el ejemplo anterior asigna el puerto 80 expuesto en el contenedor al puerto 80 del sistema host. Si ejecuta Docker de forma local, utilice su navegador web para ir a <http://localhost:80>. Si el programa se ha ejecutado correctamente, se muestra la declaración “Hello World!”.

Para obtener más información sobre el comando Docker run, consulte [Docker run reference](#) en el sitio web de Docker.

3. Creación de un repositorio nuevo

Para subir la imagen a su instancia de Amazon ECR, cree un nuevo repositorio en el que pueda almacenarse.

Creación de un repositorio de Amazon ECR

1. En la barra de actividades de VS Code, elija el icono del kit de herramientas de AWS .
2. Amplíe el menú del AWS explorador.

3. Busca la AWS región predeterminada asociada a tu AWS cuenta. A continuación, selecciónala para ver una lista de los servicios que se encuentran a través del kit de herramientas para VS Code.
4. Elija la opción ECR + para iniciar el proceso de creación de un nuevo repositorio.
5. Siga las indicaciones para completar el proceso.
6. Una vez completada, podrás acceder a tu nuevo repositorio desde la sección ECR del menú del AWS explorador.

4. Inserción, extracción y eliminación de imágenes

Después de crear una imagen a partir de su Dockerfile y crear un repositorio, puede insertarla en su repositorio de Amazon ECR. Además, si utiliza el AWS explorador con Docker y la AWS CLI, puede hacer lo siguiente:

- Extraer una imagen del repositorio.
- Eliminar una imagen que esté almacenada en su repositorio.
- Eliminar su repositorio.

Autenticación de Docker con su registro predeterminado

Se requiere autenticación para intercambiar datos entre las instancias de Amazon ECR y Docker.

Para autenticar Docker con su registro:

1. Abra un sistema operativo de línea de comandos que esté conectado a su instancia de AWS CLI.
2. Usa el `get-login-password` método para autenticarte en tu registro ECR privado.

```
aws ecr get-login-password --region region | docker login --username AWS --password-stdin AWS_account_id.dkr.ecr.region.amazonaws.com
```

Important

En el comando anterior, debe actualizar la `region` y el `AWS_account_id` con la información específica de su cuenta de AWS .

Etiquetado e inserción de una imagen en el repositorio

Después de autenticar Docker con tu instancia de AWS, envía una imagen a tu repositorio.

1. Utilice el comando Docker images para ver las imágenes que ha almacenado localmente e identifique la que desea etiquetar.

```
docker images
```

Example Ejemplo de resultados:

REPOSITORY	TAG	IMAGE ID	CREATED
hello-world	latest	e9ffedc8c286	4 minutes ago
241MB			

2. Etiquete su imagen Docker con el comando Docker tag.

```
docker tag hello-world:latest AWS_account_id.dkr.ecr.region.amazonaws.com/hello-world:latest
```

3. Inserte la imagen etiquetada en su repositorio con el comando Docker tag.

```
docker push AWS_account_id.dkr.ecr.region.amazonaws.com/hello-world:latest
```

Example Ejemplo de resultados:

```
The push refers to a repository [AWS_account_id.dkr.ecr.region.amazonaws.com/hello-world] (len: 1)
e9ae3c220b23: Pushed
a6785352b25c: Pushed
0998bf8fb9e9: Pushed
0a85502c06c9: Pushed
latest: digest:
sha256:215d7e4121b30157d8839e81c4e0912606fca105775bb0636b95aed25f52c89b size: 6774
```

Una vez que la imagen etiquetada se haya subido correctamente a tu repositorio, estará visible en el menú del AWS Explorador.

Extracción de una imagen de Amazon ECR

- Puede extraer una imagen en su instancia local del comando Docker tag.

```
docker pull AWS_account_id.dkr.ecr.region.amazonaws.com/hello-world:latest
```

Example Ejemplo de resultados:

```
The push refers to a repository [AWS_account_id.dkr.ecr.region.amazonaws.com/hello-world] (len: 1)
e9ae3c220b23: Pushed
a6785352b25c: Pushed
0998bf8fb9e9: Pushed
0a85502c06c9: Pushed
latest: digest:
sha256:215d7e4121b30157d8839e81c4e0912606fca105775bb0636b95aed25f52c89b size: 6774
```

Eliminación de una imagen de su repositorio de Amazon ECR

Hay dos métodos para eliminar una imagen desde VS Code . El primer método consiste en utilizar el AWS Explorador.

1. Desde el AWS Explorador, expanda el menú ECR
2. Expanda el repositorio del que desea eliminar una imagen.
3. Abra el menú contextual (haga clic con el botón derecho) de la etiqueta de imagen asociada a la imagen que desea eliminar.
4. Seleccione la opción Eliminar etiqueta... para eliminar todas las imágenes almacenadas que estén asociadas con esa etiqueta.

Eliminar una imagen mediante la AWS CLI

- También puede eliminar una imagen del repositorio con el batch-delete-image comando AWS ecr.

```
AWS ecr batch-delete-image \
  --repository-name hello-world \
  --image-ids imageTag=latest
```

Example Ejemplo de resultados:

```
{
  "failures": [],
  "imageIds": [
    {
      "imageTag": "latest",
      "imageDigest":
"sha256:215d7e4121b30157d8839e81c4e0912606fca105775bb0636b95aed25f52c89b"
    }
  ]
}
```

Eliminación de un repositorio desde la instancia de Amazon ECR

Hay dos métodos para eliminar un repositorio desde VS Code. El primer método consiste en utilizar el AWS Explorador.

1. Desde el AWS Explorador, expanda el menú ECR
2. Abra el menú contextual (haga clic con el botón derecho) del repositorio que desea eliminar.
3. Elija la opción Eliminar repositorio... del repositorio seleccionado.

Eliminar un repositorio de Amazon ECR de la AWS CLI

- Puede eliminar un repositorio con el comando AWS ecr delete-repository.

Note

De forma predeterminada, no puede eliminar un repositorio que contenga imágenes. Sin embargo, el indicador --force le permite hacerlo.

```
AWS ecr delete-repository \
  --repository-name hello-world \
  --force
```

Example Ejemplo de resultados:

```
{
  "failures": [],
  "imageIds": [
    {
      "imageTag": "latest",
      "imageDigest":
"sha256:215d7e4121b30157d8839e81c4e0912606fca105775bb0636b95aed25f52c89b"
    }
  ]
}
```

Creación de un servicio App Runner a través de Amazon ECR

En el siguiente tema se describe cómo crear y lanzar un AWS App Runner servicio desde el nodo Amazon Elastic Container Registry (Amazon ECR), en el. AWS Toolkit for Visual Studio Code Para obtener información detallada sobre los servicios AWS App Runner y los servicios de Amazon ECR, consulte las guías del [AWS App Runner](#)usuario de [Amazon ECR](#).

Requisitos previos

Antes de poder crear y lanzar una AWS App Runner desde Amazon ECR en el AWS kit de herramientas, debe completar lo siguiente. Para obtener una guía detallada sobre cómo realizar estos procedimientos, consulte el tema [Uso del servicio Amazon Elastic Container Registry](#) de esta Guía del usuario.

1. Creación de una dockerfile.
2. Cree una imagen a partir de su dockerfile.
3. Cree un nuevo repositorio.

4. Etiquete e inserte una imagen en el repositorio.

Creación de un AWS App Runner servicio a partir de un repositorio de Amazon ECR existente

El siguiente procedimiento describe cómo crear un AWS App Runner servicio a partir de un repositorio de Amazon ECR existente, en el AWS kit de herramientas.

1. Desde el AWS Explorador, amplíe la región que contiene el repositorio de Amazon ECR desde el que desea crear un AWS App Runner servicio.
2. Expanda el nodo del servicio Amazon ECR para ver los repositorios de Amazon ECR.
3. Abra el menú contextual (haga clic con el botón derecho) del repositorio o imagen del repositorio de Amazon ECR desde el que desee crear un AWS App Runner servicio.
4. En el menú contextual, seleccione Crear servicio de App Runner para abrir el asistente de AWS App Runner creación en VS Code
5. En Enter a port for the new service (1/5), escriba el número de puerto que desea utilizar y, a continuación, pulse **Enter** para continuar.
6. En Configure environment variables (2/5), elija Use file... para examinar y seleccionar sus archivos locales o elija Skip para omitir este paso.
7. En Select a role to pull from ECR (3/5), elija un rol de IAM existente de la lista.

 Note

El AppRunnerECRAccessrol de acceso al rol es necesario para crear un AWS App Runner servicio desde un registro privado de Amazon ECR. Si un rol válido no está disponible en la lista, elija + (Crear rol...) icono para crear y asignar automáticamente AppRunnerECRAccessel rol a su registro.

8. En Name your service (4/5), escriba un nombre para el nuevo servicio y, a continuación, pulse **Enter** para continuar.
9. En Select instance configuration (5/5), elija la configuración de **vCPU** y **Memory** de la lista para crear el servicio.
10. Desde el AWS Explorador, expanda el nodo de servicio de App Runner para ver sus AWS App Runner recursos. Cuando el nuevo servicio se haya creado correctamente, el estado se actualizará de forma automática a En ejecución.

Trabajar con Amazon Elastic Container Service

El AWS Toolkit for Visual Studio Code proporciona cierta compatibilidad con [Amazon Elastic Container Service \(Amazon ECS\)](#). El kit de herramientas para VS Code le ayuda con algunas tareas relacionadas con Amazon ECS, como la creación de definiciones de tareas.

Temas

- [Uso de IntelliSense para archivos de definición de tareas de Amazon ECS](#)
- [Ejecutivo de Amazon Elastic Container Service en AWS Toolkit for Visual Studio Code](#)

Uso de IntelliSense para archivos de definición de tareas de Amazon ECS

Una de las cosas que puede hacer al trabajar con Amazon Elastic Container Service (Amazon ECS) es crear definiciones de tareas, tal y como se describe en [Cómo crear una definición de tarea](#) en la Guía para desarrolladores de Amazon Elastic Container Service. La instalación del AWS Toolkit for Visual Studio Code incluye la funcionalidad de IntelliSense para los archivos de definición de tareas de Amazon ECS.

Requisitos previos

- Asegúrese de que el sistema cumple los requisitos previos especificados en [Instalación del kit de herramientas para VS Code](#).

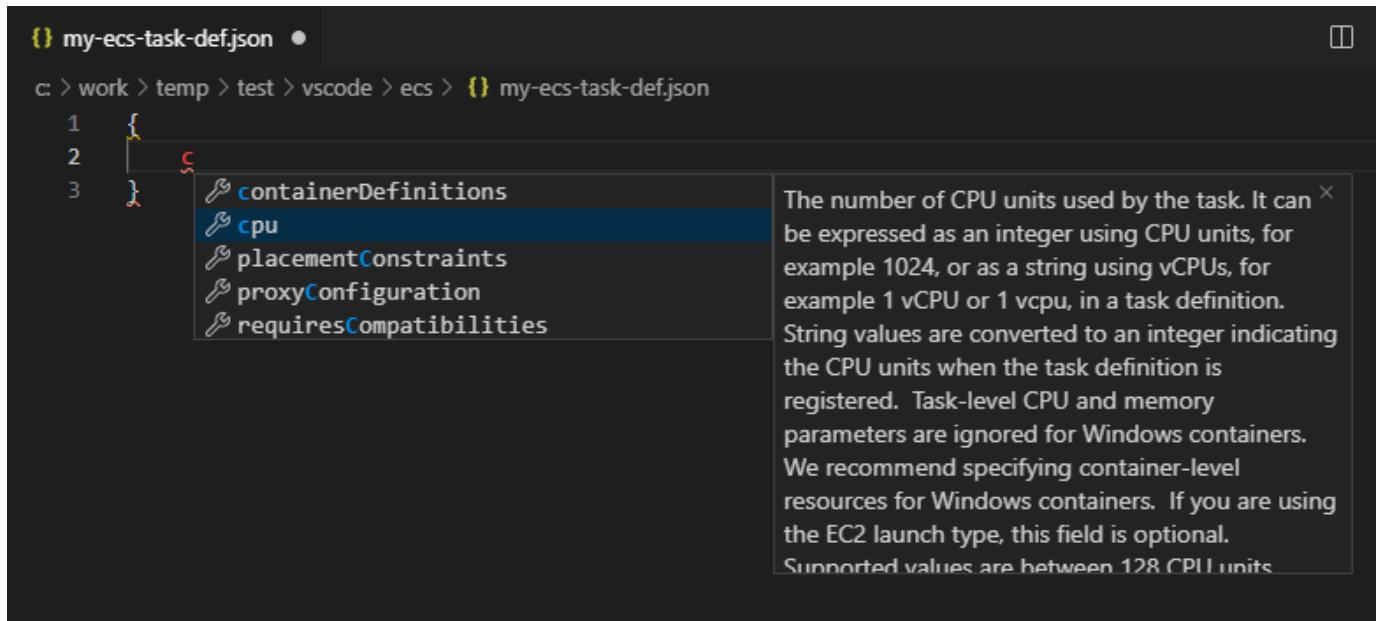
Uso de IntelliSense en archivos de definición de tareas de Amazon ECS

El siguiente ejemplo muestra cómo puede sacar partido a IntelliSense en archivos de definición de tareas de Amazon ECS.

1. Cree un archivo JSON para la definición de la tarea de Amazon ECS. El nombre del archivo debe contener `ecs-task-def.json` al final, pero puede tener caracteres adicionales al principio.

Para este ejemplo, cree un archivo denominado `.my-ecs-task-def.json`

2. Abra el archivo en un editor de VS Code y escriba las llaves iniciales.
3. Escriba la letra «`c`» como si quisiera agregar `cpu` a la definición. Observe el cuadro de diálogo IntelliSense que se abre, que es similar al siguiente.



Ejecutivo de Amazon Elastic Container Service en AWS Toolkit for Visual Studio Code

Puede emitir comandos individuales en un contenedor de Amazon Elastic Container Service (Amazon ECS) con AWS Toolkit for Visual Studio Code la función Amazon ECS Exec.

⚠ Importante

La activación y desactivación de Amazon ECS Exec cambia el estado de los recursos de su AWS cuenta. Esto incluye la detención y el reinicio del servicio. La modificación del estado de los recursos mientras Amazon ECS Exec está activado puede generar resultados impredecibles. Para obtener más información sobre Amazon ECS, consulte [Utilización de Amazon ECS Exec para la depuración](#) en la Guía para desarrolladores.

Requisitos previos de Amazon ECS Exec

Para poder utilizar la característica Amazon ECS Exec, deben cumplirse algunas condiciones previas.

Requisitos de Amazon ECS

En función de si sus tareas están alojadas en Amazon EC2 o AWS Fargate Amazon ECS Exec tiene diferentes requisitos de versión.

- Si utilizas Amazon EC2, debes usar una AMI optimizada para Amazon ECS que se haya publicado después del 20 de enero de 2021, con una versión de agente de 1.50.2 o superior. Encontrará información adicional en la guía para desarrolladores [Amazon ECS optimizada AMIs](#).
- Si la utiliza AWS Fargate, debe utilizar la versión 1.4.0 o superior de la plataforma. Encontrará información adicional sobre los requisitos de Fargate en [Versiones de la plataforma de AWS Fargate](#) en la Guía para desarrolladores.

AWS configuración de la cuenta y permisos de IAM

Para utilizar la función Amazon ECS Exec, debe tener un clúster de Amazon ECS existente asociado a su AWS cuenta. Amazon ECS Exec usa Systems Manager para establecer una conexión con los contenedores del clúster y requiere permisos de rol de IAM de tareas específicos para comunicarse con el servicio de SSM.

Puede encontrar información sobre el rol y la política de IAM, específica de Amazon ECS Exec, en la guía para desarrolladores [Permisos de IAM requeridos para ECS Exec](#).

Trabajar con Amazon ECS Exec

Puede activar o desactivar Amazon ECS Exec directamente desde el AWS explorador en el Toolkit for VS Code. Cuando está habilitado Amazon ECS Exec, puede elegir contenedores en el menú de Amazon ECS y ejecutar comandos en ellos.

Activación de Amazon ECS Exec

1. Desde el AWS explorador, localice y amplíe el menú de Amazon ECS.
2. Amplíe el clúster con el servicio que desee modificar.
3. Abra el menú contextual (haga clic con el botón derecho) del servicio y elija Enable Command Execution (Activar ejecución de comandos).

⚠ Important

Esto iniciará una nueva implementación de su servicio y puede tardar unos minutos. Para obtener más información, consulte la nota al principio de esta sección.

Desactivación de Amazon ECS Exec

1. Desde el AWS explorador, localice y amplíe el menú de Amazon ECS.
2. Expanda el clúster que almacene el servicio que desea.
3. Abra el menú contextual (haga clic con el botón derecho) del servicio y elija Disable Command Execution (Desactivar ejecución de comandos).

⚠ Important

Esto iniciará una nueva implementación de su servicio y puede tardar unos minutos. Para obtener más información, consulte la nota al principio de esta sección.

Ejecución de comandos en un contenedor

Para ejecutar comandos en un contenedor mediante el AWS Explorador, Amazon ECS Exec debe estar habilitado. Si no está habilitado, consulte el procedimiento para habilitar ECS Exec en esta sección.

1. Desde el AWS explorador, localice y amplíe el menú de Amazon ECS.
2. Expanda el clúster que almacene el servicio que desea.
3. Expanda el servicio para enumerar los contenedores asociados.
4. Abra el menú contextual (haga clic con el botón derecho) del contenedor y seleccione Run Command in Container (Ejecutar comando en el contenedor).
5. Se abrirá un mensaje con una lista de tareas en ejecución. Elija el ARN de la tarea que deseé.

 ⓘ Note

Si solo hay una tarea en ejecución para ese servicio, se seleccionará automáticamente y se omitirá este paso.

6. Cuando se le solicite, escriba el comando que desee ejecutar y pulse Intro para continuar.

Trabajo con Amazon EventBridge

El AWS Toolkit for Visual Studio Code (VS Code) proporciona compatibilidad con [Amazon EventBridge](#). Con el kit de herramientas para VS Code, puede trabajar con ciertos aspectos de EventBridge, como los esquemas.

Temas

- [Trabajo con esquemas de Amazon EventBridge](#)

Trabajo con esquemas de Amazon EventBridge

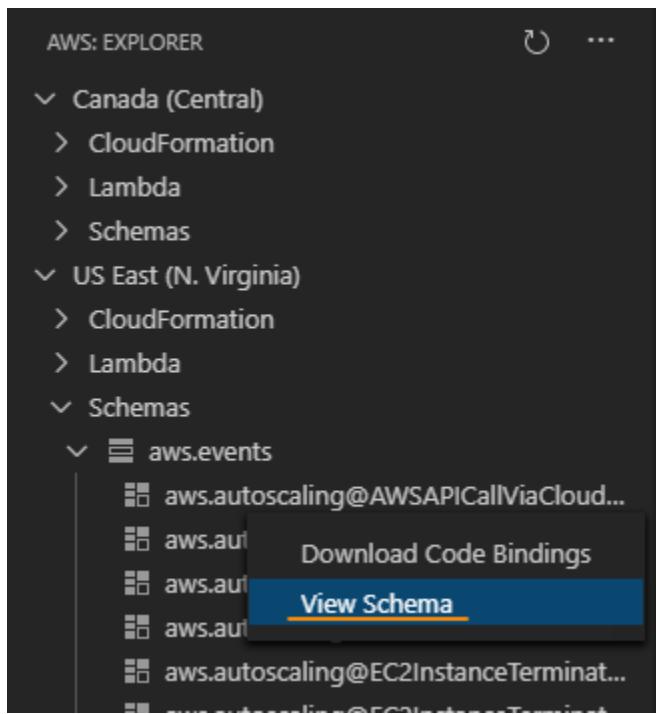
Puede utilizar el AWS Toolkit for Visual Studio Code (VS Code) para realizar diversas operaciones en los [esquemas de Amazon EventBridge](#).

Requisitos previos

- Asegúrese de que el sistema cumple los requisitos previos especificados en [Instalación del kit de herramientas para VS Code](#).
- El esquema de EventBridge con el que desea trabajar debe estar disponible en la cuenta de AWS. Si no lo está, créelo o cárguelo. Para obtener más información, consulte [Esquemas de Amazon EventBridge Schemas](#) en la [Guía del usuario de Amazon EventBridge](#).

Visualización de un esquema disponible

1. Desde el Explorador de AWS, expanda Schemas (Esquemas).
2. Expanda el nombre del registro que contiene el esquema que desea ver. Por ejemplo, muchos de los esquemas que AWS proporciona están en el registro aws.events.
3. Para ver un esquema en el editor, abra el menú contextual del esquema y, a continuación, elija View Schema (Ver esquema).

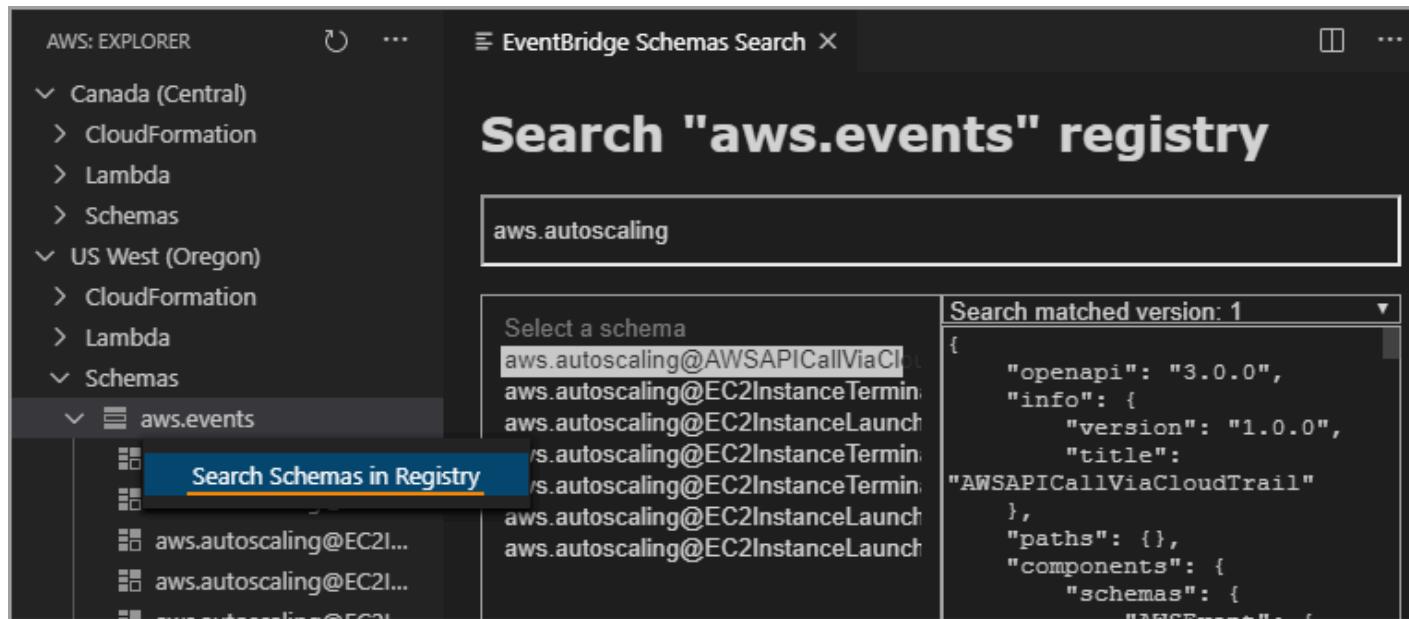


Búsqueda de un esquema disponible

En el Explorador de AWS, realice una o varias de las siguientes acciones:

- Escriba el título del esquema que desea buscar. El Explorador de AWS resaltará los títulos de esquema que contengan una coincidencia. (debe expandir un registro para ver los títulos resaltados).
- Abra el menú contextual de Schemas (Esquemas) y, a continuación, elija Search Schemas (Buscar esquemas). De forma alternativa, expanda Schemas (Esquemas), abra el menú contextual del registro que contiene el esquema que desea buscar y, a continuación, elija Search Schemas in Registry (Buscar esquemas en el registro). En el cuadro de diálogo Búsqueda de esquemas de EventBridge, comience a escribir el título del esquema que deseé buscar. El cuadro de diálogo muestra los títulos de esquema que contienen una coincidencia.

Para mostrar el esquema en el cuadro de diálogo, seleccione el título del esquema.



Generación de código para un esquema disponible

1. En el Explorador de AWS, expanda Schemas (Esquemas).
2. Expanda el nombre del registro que contiene el esquema para el que desea generar el código.
3. Haga clic con el botón derecho en el título del esquema y, a continuación, elija Download code bindings (Descargar vínculos de código).
4. En las páginas del asistente que aparecerán, elija lo siguiente:
 - La versión del esquema
 - El lenguaje del vínculo de código
 - La carpeta del espacio de trabajo donde desea almacenar el código generado en la máquina de desarrollo local

AWS Analizador de acceso IAM

Puede realizar comprobaciones de las políticas de [AWS Identity and Access Analyzer \(IAM\) Access Analyzer](#) en sus políticas de IAM creadas en CloudFormation plantillas, planes de Terraform y documentos de políticas de JSON, utilizando el analizador de acceso de IAM del AWS Toolkit for Visual Studio Code

Las comprobaciones de políticas del Analizador de acceso de AWS IAM incluyen la validación de las políticas y comprobaciones de políticas personalizadas. La validación de políticas ayuda a validar sus políticas de IAM de acuerdo con los estándares detallados en la [gramática del lenguaje de políticas JSON de IAM y las mejores prácticas de AWS seguridad en los temas de IAM](#), que se encuentran en la Guía del usuario. AWS Identity and Access Management Los resultados de la validación de sus políticas incluyen advertencias de seguridad, errores, advertencias generales y sugerencias relativas a la política.

También puede ejecutar comprobaciones de políticas personalizadas de nuevo acceso en función de sus estándares de seguridad. Se asocia un cargo a cada comprobación de política personalizada de nuevo acceso. Para obtener información detallada acerca de los precios, consulte el sitio [Precios del Analizador de acceso de AWS IAM](#). Para obtener más información sobre las comprobaciones de políticas del Analizador de acceso de AWS IAM, consulte el tema [Comprobaciones para validar políticas](#) de la Guía del usuario de AWS Identity and Access Management.

En los temas siguientes se describe cómo trabajar con las comprobaciones de políticas del Analizador de acceso de AWS IAM en el AWS Toolkit for Visual Studio Code.

Temas

- [Uso de AWS IAM Access Analyzer](#)

Uso de AWS IAM Access Analyzer

En las secciones siguientes se describe cómo realizar la validación de las políticas de IAM, así como comprobaciones de políticas personalizadas, en el AWS Toolkit for Visual Studio Code. Para obtener más información, consulte los siguientes temas de la guía del AWS Identity and Access Management usuario: [validación de políticas de IAM Access Analyzer y comprobaciones de políticas personalizadas](#) de [IAM Access Analyzer](#).

Requisitos previos

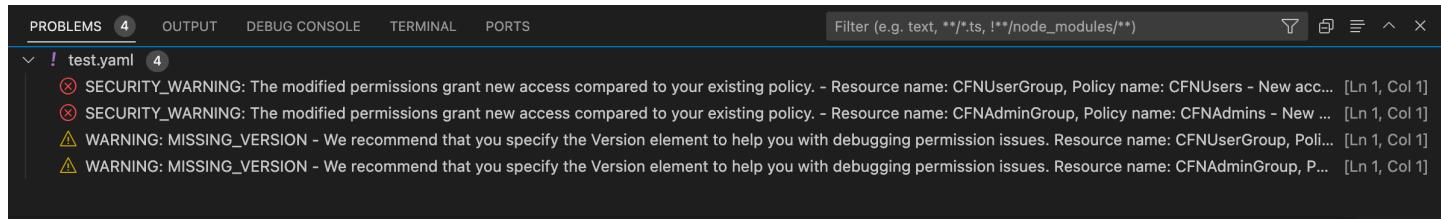
Se deben cumplir los siguientes requisitos previos para poder trabajar con las comprobaciones de políticas del Analizador de acceso de IAM en el kit de herramientas.

- Instale la versión 3.6 de Python o una versión posterior.
- Instale el [Validador de políticas de IAM para CloudFormation](#) o el [Validador de políticas de IAM para Terraform](#) que requieren las herramientas de la CLI de Python y que se especifica en la ventana de comprobaciones de políticas de IAM.

- Configure las credenciales de su rol. AWS

Verificaciones de políticas del Analizador de acceso de IAM

Puede realizar comprobaciones de políticas para CloudFormation plantillas, planes de Terraform y documentos de políticas de JSON utilizando el AWS Toolkit for Visual Studio Code. Los resultados de sus comprobaciones se pueden ver en el panel Problemas de VS Code. En la imagen siguiente se muestra el panel Problemas de VS Code.



El Analizador de acceso de IAM ofrece cuatro tipos de comprobaciones:

- Validar la política
- CheckAccessNotGranted
- CheckNoNewAccess
- CheckNoPublicAccess

En las secciones siguientes se describe cómo ejecutar cada tipo de comprobación.

Note

Configure las credenciales de su AWS rol antes de realizar cualquier tipo de verificación. Los archivos compatibles incluyen los siguientes tipos de documentos: CloudFormation plantillas, planes de Terraform y documentos de políticas de JSON.

El administrador o el equipo de seguridad suelen proporcionar las referencias a las rutas de los archivos, que pueden ser una ruta de archivo del sistema o un URI de bucket de Amazon S3. Para usar el URI de un bucket de Amazon S3, su rol actual debe tener acceso a dicho bucket.

Se asocia un cargo a cada comprobación de política personalizada. Para obtener más información sobre los precios de la comprobación de políticas personalizadas, consulte la guía [Precios del Analizador de acceso de AWS IAM](#).

Ejecución de Validar la política

La comprobación de la política de validación, también conocida como validación de políticas, valida la política según la gramática y AWS las mejores prácticas de las políticas de IAM. Para obtener información adicional, consulte los temas sobre la [gramática del lenguaje de políticas JSON de IAM](#) y [las prácticas recomendadas de AWS seguridad en IAM](#), que se encuentran en la Guía del usuario. AWS Identity and Access Management

1. En VS Code, abra un archivo compatible que contenga políticas de AWS IAM en el editor de VS Code.
2. Para abrir las comprobaciones de políticas del Analizador de acceso de IAM, abra la paleta de comandos de VS Code; para ello, pulse **CTRL+Shift+P**, busque **IAM Policy Checks** y, a continuación, haga clic para abrir el panel IAM Policy Checks en el editor de VS Code.
3. En el panel IAM Policy Checks, seleccione su tipo de documento en el menú desplegable.
4. En la sección Validate Policies, elija el botón Run Policy Validation para ejecutar la comprobación de validación de las políticas.
5. En el panel Problemas de VS Code, revise los resultados de sus comprobaciones de políticas.
6. Actualice la política y repita este procedimiento; vuelva a ejecutar la comprobación de validación de las políticas hasta que dejen de aparecer errores o advertencias de seguridad en los resultados de dichas comprobaciones.

En ejecución CheckAccessNotGranted

CheckAccessNotGranted es una comprobación de políticas personalizada para comprobar que su política no permite acciones específicas de IAM.

Note

El administrador o el equipo de seguridad suelen proporcionar las referencias a las rutas de los archivos, que pueden ser una ruta de archivo del sistema o un URI de bucket de Amazon S3. Para usar el URI de un bucket de Amazon S3, su rol actual debe tener acceso a dicho bucket. Debe especificarse al menos una acción o recurso y el archivo debe estar estructurado conforme al siguiente ejemplo:

```
{"actions": ["action1", "action2", "action3"], "resources": ["resource1", "resource2", "resource3"]}
```

1. En VS Code, abra un archivo compatible que contenga políticas de AWS IAM en el editor de VS Code.
2. Para abrir las comprobaciones de políticas del Analizador de acceso de IAM, abra la paleta de comandos de VS Code; para ello, pulse **CTRL+Shift+P**, busque **IAM Policy Checks** y, a continuación, haga clic para abrir el panel IAM Policy Checks en el editor de VS Code.
3. En el panel IAM Policy Checks, seleccione su tipo de documento en el menú desplegable.
4. En la sección Comprobaciones de políticas personalizadas, seleccione **CheckAccessNotGranted**.
5. En el campo de entrada de texto, puede introducir una lista separada por comas que contenga acciones y recursos. ARNs Debe proporcionarse al menos una acción o recurso.
6. Elija el botón **Run Custom Policy Check**.
7. En el panel Problemas de VS Code, revise los resultados de sus comprobaciones de políticas. Las comprobaciones de políticas personalizadas devuelven el resultado **PASS** o **FAIL**.
8. Actualice su política y repita este procedimiento y vuelva a ejecutar la comprobación hasta que se devuelva. **CheckAccessNotGranted PASS**

En ejecución CheckNoNewAccess

CheckNoNewAccess es una verificación de política personalizada para verificar si su política otorga nuevos accesos en comparación con una política de referencia.

1. En VS Code, abra un archivo compatible que contenga políticas de AWS IAM en el editor de VS Code.
2. Para abrir las comprobaciones de políticas del Analizador de acceso de IAM, abra la paleta de comandos de VS Code; para ello, pulse **CTRL+Shift+P**, busque **IAM Policy Checks** y, a continuación, haga clic para abrir el panel IAM Policy Checks en el editor de VS Code.
3. En el panel IAM Policy Checks, seleccione su tipo de documento en el menú desplegable.
4. En la sección Comprobaciones de políticas personalizadas, seleccione **CheckNoNewAccess**.
5. Ingrese un documento de política JSON de referencia. También puede proporcionar una ruta de archivo que haga referencia a un documento de política JSON.

6. En Select a Reference Policy Type, seleccione el tipo de política de referencia que coincide con el de su documento de referencia.
7. Elija el botón Run Custom Policy Check.
8. En el panel Problemas de VS Code, revise los resultados de sus comprobaciones de políticas. Las comprobaciones de políticas personalizadas devuelven el resultado PASS o FAIL.
9. Actualice su política y repita este procedimiento y vuelva a ejecutar la CheckNoNewAccess comprobación hasta que se devuelvaPASS.

En ejecución CheckNoPublicAccess

CheckNoPublicAccess es una comprobación de políticas personalizada para comprobar si tu política concede acceso público a los tipos de recursos compatibles en tu plantilla.

Para obtener información específica sobre los tipos de recursos compatibles, consulta los [terraform-iam-policy-validator](#) GitHub repositorios [cloudformation-iam-policy-validator](#).

1. En VS Code, abra un archivo compatible que contenga políticas de AWS IAM en el editor de VS Code.
2. Para abrir las comprobaciones de políticas del Analizador de acceso de IAM, abra la paleta de comandos de VS Code; para ello, pulse **CTRL+Shift+P**, busque **IAM Policy Checks** y, a continuación, haga clic para abrir el panel IAM Policy Checks en el editor de VS Code.
3. En el panel IAM Policy Checks, seleccione su tipo de documento en el menú desplegable.
4. En la sección Comprobaciones de políticas personalizadas, seleccione CheckNoPublicAccess.
5. Elija el botón Run Custom Policy Check.
6. En el panel Problemas de VS Code, revise los resultados de sus comprobaciones de políticas. Las comprobaciones de políticas personalizadas devuelven el resultado PASS o FAIL.
7. Actualice su política y repita este procedimiento y vuelva a ejecutar la CheckNoNewAccess comprobación hasta que se devuelvaPASS.

Working with AWS IoT in AWS Toolkit for Visual Studio Code

AWS IoT en AWS Toolkit for Visual Studio Code permite interactuar con el servicio AWS IoT y, al mismo tiempo, minimizar las interrupciones de su flujo de trabajo en VS Code. En esta guía se explica cómo empezar a utilizar las funciones del servicio AWS IoT que están disponibles en el AWS

Toolkit for Visual Studio Code. Para obtener más información sobre el servicio AWS IoT, consulte [¿Qué es AWS IoT?](#) en la Guía para desarrolladores.

AWS IoTRequisitos previos de

Para empezar a utilizar AWS IoT desde el kit de herramientas para VS Code, asegúrese de que su cuenta de AWS y VS Code cumplen los requisitos de estas guías:

- Para obtener información sobre los requisitos de la cuenta de AWS y los permisos de usuario de AWS específicos del servicio AWS IoT , consulte [Introducción a AWS IoT Core](#) en la Guía para desarrolladores.
- Para conocer los requisitos específicos del kit de herramientas para VS Code, consulte [Configuración del kit de herramientas para VS Code](#) en la guía del usuario.

Objetos de AWS IoT

AWS IoT conecta dispositivos a servicios y recursos en la nube de AWS. Puede conectar sus dispositivos a AWS IoT mediante objetos. Un objeto es una representación de una entidad lógica o dispositivo concreto. Puede ser un dispositivo físico o un sensor (por ejemplo, una bombilla o un interruptor en la pared). Para obtener más información sobre los objetos de AWS IoT, consulte [Administración de dispositivos con AWS IoT](#) en la guía para desarrolladores.

Administración de objetos de AWS IoT

El kit de herramientas de VS Code tiene varias características que aumentan la eficiencia de la administración de sus objetos de AWS IoT. Hay varias maneras en las que puede usar el kit de herramientas de VS Code para administrar objetos de AWS IoT:

- [Create a thing](#)
- [Attach a certificate to a thing](#)
- [Detach a certificate from a thing](#)
- [Delete a thing](#)

Crear un objeto

1. Desde el Explorador de AWS, expanda el encabezado del servicio IoT y seleccione con el botón secundario (clic con el botón derecho) Objetos.

2. Seleccione Crear objeto en el menú contextual para abrir un cuadro de diálogo.
3. Siga las instrucciones e introduzca un nombre para su objeto de IoT en el campo Nombre del objeto.
4. Cuando termine, aparecerá un ícono de objeto seguido del nombre que ha especificado que sea visible en la sección Objeto.

Para asociar un certificado a un objeto

1. Desde el Explorador de AWS, expanda la sección del servicio IoT.
2. En la subsección Objetos, busque el objeto al que va a adjuntar el certificado.
3. Abra el menú contextual (clic con el botón derecho) del objeto y seleccione Adjuntar certificado para abrir un selector de entrada con una lista de sus certificados.
4. En la lista, elija el ID de certificado que corresponda al que desee adjuntar a su objeto.
5. Una vez completado este paso, su certificado estará accesible en el Explorador de AWS como un elemento del objeto al que lo haya adjuntado.

Para desasociar un certificado de un objeto

1. Desde el Explorador de AWS, expanda la sección del servicio IoT.
2. En la subsección Things (Objetos), localice el objeto del que desea desasociar un certificado.
3. Abra el menú contextual (clic con el botón derecho) del objeto y seleccione Desasociar certificado.
4. Cuando finalice este paso, el certificado desasociado dejará de mostrarse debajo de ese objeto en el Explorador de AWS, pero seguirá siendo accesible desde la subsección Certificados.

Para eliminar un objeto

1. Desde el Explorador de AWS, expanda la sección del servicio IoT.
2. En la subsección Objetos, localice el objeto que desea eliminar.
3. Abra el menú contextual (clic con el botón derecho) del objeto y seleccione Eliminar objeto para eliminarlo.

4. Una vez completado este paso, el objeto eliminado ya no estará disponible en la subsección Objetos.

 Note

Nota: Solo puede eliminar un objeto que no tenga un certificado asociado.

AWS IoT Certificados de

Los certificados son una forma habitual de crear una conexión segura entre sus servicios de AWS IoT y sus dispositivos. Los certificados X.509 son certificados digitales que utilizan el estándar de infraestructura de clave pública X.509 para asociar una clave pública a una identidad contenida en un certificado. Para obtener más información sobre los certificados de AWS IoT, consulte [Autenticación \(IoT\)](#) en la guía para desarrolladores.

Administración de certificados

El kit de herramientas para VS Code le ofrece diversas formas de administrar sus certificados de AWS IoT directamente desde el Explorador de AWS.

- [Create a certificate](#)
- [Change a certificate status](#)
- [Attach a policy to a certificate](#)
- [Delete a certificate](#)

Para crear un certificado de AWS IoT

Puede utilizar un certificado X.509 para conectarse con su instancia de AWS IoT.

1. Desde el Explorador de AWS, expanda la sección del servicio IoT y seleccione con el botón secundario (clic con el botón derecho) Certificados.
2. Seleccione Crear certificado en el menú contextual para abrir un cuadro de diálogo.
3. Seleccione un directorio en su sistema de archivos local para guardar su par de claves RSA y certificado X.509.

Note

- Los nombres de archivo predeterminados contienen el ID de certificado como prefijo.
- Solo se almacena el certificado X.509 con su cuenta de AWS mediante el servicio AWS IoT.
- Su par de claves RSA solo puede emitirse una vez. Guárdelo en una ubicación segura de su sistema de archivos cuando se le solicite.
- Si el certificado o el par de claves no pueden guardarse en su sistema de archivos en este momento, el kit de herramientas de AWS elimina el certificado de su cuenta de AWS.

Para modificar el estado de un certificado

El estado de un certificado individual se muestra junto a su ID en el Explorador de AWS y puede definirse como activo, inactivo o revocado.

Note

- Un certificado debe tener el estado **active** (activo) para poder usarlo para conectar su dispositivo al servicio AWS IoT.
- Un certificado inactivo puede activarse, tanto si se ha desactivado previamente como si está inactivo de forma predeterminada.
- Un certificado que se ha revocado no se puede reactivar.

1. Desde el Explorador de AWS, expanda la sección del servicio IoT.
 2. En la subsección Certificados, busque el certificado que desee modificar.
 3. Seleccione con el botón secundario (clic con el botón derecho) el certificado para abrir un menú contextual que muestra las opciones de cambio de estado disponibles para ese certificado.
-
- Si un certificado tiene el estado **inactive** (inactivo), elija **activate** (activar) para cambiar el estado a **active** (activo).

- Si un certificado tiene el estado active (activo), elija deactivate (desactivar) para cambiar el estado a inactive (inactivo).
- Si un certificado tiene el estado active (activo) o inactive (inactivo), elija revoke (revocar) para cambiar el estado a revoked (revocado).

 Note

Puede acceder también a todas estas acciones de cambio de estado seleccionando un certificado asociado a un objeto mientras se muestra en la subsección **Objetos**.

Para asociar una política de IoT a un certificado

1. Desde el Explorador de AWS, expanda la sección del servicio IoT.
2. En la subsección Certificados, busque el certificado que desee modificar.
3. Seleccione con el botón secundario (clic con el botón derecho) el certificado y elija Asociar política en el menú contextual para abrir un selector de entrada con una lista de sus políticas disponibles.
4. Elija la política que desee asociar al certificado.
5. Una vez completado este paso, la política que haya seleccionado se agregará al certificado como un elemento de submenú.

Para desasociar una política de IoT de un certificado

1. Desde el Explorador de AWS, expanda la sección del servicio IoT.
2. En la subsección Certificados, busque el certificado que desee modificar.
3. Expanda el certificado y busque la política que desee desasociar.
4. Seleccione con el botón secundario (clic con el botón derecho) la política y elija Desasociar en el menú contextual.
5. Una vez completado este paso, la política dejará de ser un elemento accesible desde su certificado, pero estará disponible en la subsección Política.

Para eliminar un certificado

1. Desde el Explorador de AWS, expanda el encabezado del servicio IoT.
2. En la subsección Certificados, busque el certificado que desee eliminar.
3. Seleccione con el botón secundario (clic con el botón derecho del ratón) el certificado y elija Eliminar certificado en el menú contextual.

 Note

No puede eliminar un certificado si está asociado a un objeto o tiene un estado activo.

Puede eliminar un certificado que tenga políticas asociadas.

AWS IoTPolíticas de

Las políticas básicas de AWS IoT se definen mediante documentos JSON, cada uno de los cuales contiene una o más declaraciones de políticas. Las políticas definen cómo AWS IoT, AWS y su dispositivo pueden interactuar entre sí. Para obtener más información sobre cómo crear un documento de políticas, consulte [Políticas de IoT](#) en la guía para desarrolladores.

 Note

Las políticas con nombre tienen una versión para que pueda revertirlas. Desde el Explorador de AWS, sus políticas de IoT aparecen en la subsección Políticas del servicio IoT. Puede ver las versiones de una política si la expande. La versión predeterminada se indica con un asterisco.

Administración de políticas

El kit de herramientas para VS Code ofrece varias formas de administrar sus políticas del servicio AWS IoT. Estas son formas de administrar o modificar sus políticas directamente desde el Explorador de AWS en VS Code:

- [Create a policy](#)
- [Upload a new policy version](#)
- [Edit a policy version](#)

- [Change the policy version default](#)
- [Change the policy version default](#)

Creación de una política de AWS IoT

Note

Puede crear una política nueva desde el Explorador de AWS, pero el documento JSON que define la política ya debe estar en el sistema de archivos.

1. Desde el Explorador de AWS, expanda la sección del servicio IoT.
2. Seleccione con el botón secundario (clic con el botón derecho) la subsección Políticas y elija Crear política a partir de un documento para abrir el campo de entrada Nombre de la política.
3. Introduzca un nombre y siga las instrucciones para abrir un cuadro de diálogo que le pedirá que seleccione un documento JSON de su sistema de archivos.
4. Elija el archivo JSON que contiene las definiciones de su política. La política estará disponible en el Explorador de AWS cuando se complete esta operación.

Para cargar una nueva versión de política de AWS IoT

Se puede crear una nueva versión de una política cargando un documento JSON en la política.

Note

El nuevo documento JSON debe estar en el sistema de archivos para crear una nueva versión mediante el Explorador de AWS.

1. Desde el Explorador de AWS, expanda la sección del servicio IoT.
2. Expanda la subsección Políticas para ver sus políticas de AWS IoT.
3. Seleccione con el botón secundario (clic con el botón derecho) la política que desea actualizar y elija Crear versión nueva a partir de un documento.
4. Cuando se abra el cuadro de diálogo, elija el archivo JSON que contiene las actualizaciones de las definiciones de la política.

5. La nueva versión será accesible desde su política en el Explorador de AWS.

Para editar una versión de política de AWS IoT

Se puede abrir y editar un documento de política con VS Code. Cuando termine de editar el documento, puede guardarlo en el sistema de archivos. A continuación, puede cargarlo en su servicio AWS IoT desde el Explorador de AWS.

1. Desde el Explorador de AWS, expanda la sección del servicio IoT.
2. Amplíe la subsección Políticas y busque la política que desea actualizar. Seleccione Crear una política a partir del documento para abrir el campo de entrada Nombre de la política.
3. Expanda la política que desea actualizar y, a continuación, seleccione con el botón secundario (clic con el botón derecho) la versión de la política que desea editar.
4. Seleccione Ver en el menú contextual para abrir la versión de la política en VS Code
5. Cuando se abra el documento de la política, haga los cambios que desee y guárdelos.

Note

En este momento, los cambios realizados en la política solo se guardan en el sistema de archivos local. Para actualizar la versión y realizar un seguimiento con el Explorador de AWS, repita los pasos que se describen en el procedimiento [Upload a new policy version](#).

Para seleccionar una nueva versión de política (predeterminada)

1. Desde el Explorador de AWS, expanda la sección del servicio IoT.
2. Amplíe la subsección Policies (Políticas) y localice la política que desea actualizar.
3. Expanda la política que desea actualizar y, a continuación, seleccione con el botón secundario (clic con el botón derecho) la versión de la política que desea establecer y elija Establecer como predeterminada.
4. Cuando haya terminado, la nueva versión predeterminada que ha seleccionado tendrá una estrella junto a ella.

Para eliminar políticas

Note

Para poder eliminar una política o una versión de ella, es necesario que cumplan una serie de condiciones.

- No puede eliminar una política si está asociada a un certificado.
- No puede eliminar una política si tiene versiones distintas de las predeterminadas.
- No puede eliminar la versión predeterminada de una política a menos que seleccione una nueva versión predeterminada o elimine la política completa.
- Para poder eliminar una política completa, debe eliminar primero todas las versiones no predeterminadas de esa política.

1. Desde el Explorador de AWS, expanda la sección del servicio IoT.
2. Expanda la subsección Policies (Políticas) y localice la política que desea actualizar.
3. Expanda la política que desea actualizar y, a continuación, seleccione con el botón secundario (clic con el botón derecho) la versión de la política que desea eliminar y elija Eliminar.
4. Cuando se elimina una versión, deja de estar visible en el Explorador.
5. Si solo queda la versión predeterminada de una política, seleccione con el botón secundario (clic con el botón derecho) la política principal y elija Eliminar.

Funciones de AWS Lambda

El AWS Toolkit for Visual Studio Code ofrece soporte integral para las funciones de AWS Lambda, lo que permite crear, probar e implementar directamente desde VS Code.

Lambda es un servicio de computación totalmente administrado y basado en eventos que ejecuta el código de forma automática en respuesta a eventos de más de doscientas aplicaciones de software como servicio (SaaS) y servicios de AWS. Para obtener información detallada sobre el servicio AWS Lambda, consulte la Guía para desarrolladores de [AWS Lambda](#).

En los temas siguientes se describe cómo trabajar con AWS Lambda en el AWS Toolkit for Visual Studio Code:

Temas

- [Trabajar con AWS Lambda funciones](#)
- [AWS Lambda console a IDE](#)
- [AWS Lambda con LocalStack apoyo](#)
- [AWS Lambda depuración remota](#)

Trabajar con AWS Lambda funciones

AWS Toolkit for Visual Studio Code Esto le permite trabajar con sus AWS Lambda funciones en su entorno local de VS Code. Con el AWS kit de herramientas, puede crear, editar, probar, depurar e implementar sus funciones de Lambda sin tener que salir del IDE. Para obtener información detallada sobre el AWS Lambda servicio, consulte la [AWS Lambda](#)Guía para desarrolladores.

En las secciones siguientes se describe cómo empezar a trabajar con funciones de Lambda en el AWS Toolkit for Visual Studio Code.

 Note

Si ya ha creado funciones Lambda mediante el Consola de administración de AWS, puede invocarlas desde el kit de herramientas. Además, puede abrir sus funciones de Lambda en VS Code desde AWS Lambda console, para obtener información adicional, consulte el [AWS Lambda console a IDE](#) tema de esta guía del usuario. Para crear una función de Lambda en VS Code, siga los pasos que se describen en el tema [Creación de una nueva aplicación sin servidor \(local\)](#) de esta Guía del usuario.

Requisitos previos

Se deben cumplir las siguientes condiciones para trabajar con el AWS Lambda servicio del AWS kit de herramientas.

- La última versión de AWS Toolkit for Visual Studio Code se instala y configura con sus AWS credenciales.
- Sus políticas y permisos gestionados AWS Identity and Access Management (IAM) están configurados para funcionar con el AWS Lambda servicio. Para obtener información detallada sobre cómo configurar los permisos y crear una política AWS gestionada compatible, consulte el AWS Lambda tema [AWS Identity and Access Management correspondiente de la Guía para AWS Lambda](#) desarrolladores.

- Ya dispone de AWS Lambda funciones o está familiarizado con la creación de una. Para obtener instrucciones sobre cómo crear una función de Lambda, consulte el tema [Creación de su primera función de Lambda](#) de la Guía para desarrolladores de AWS Lambda .

Invocación de una función de Lambda

Para invocar una función Lambda desde AWS su cuenta en VS Code, complete los siguientes pasos.

1. Desde AWS Toolkit for Visual Studio Code, expanda el AWS explorador.
2. Desde el AWS explorador, expanda Lambda para ver sus recursos de Lambda.
3. Abra el menú contextual (clic con el botón derecho) de la función de Lambda que desee invocar y, a continuación, elija Invocar en la nube o el icono Invocar en la nube para abrir el menú Remote invoke configuration en VS Code.
4. En el menú Remote invoke configuration, especifique la configuración de Payload y agregue la información adicional que sea necesaria para el evento.

 Note

El primer proceso de invocación puede empezar a ejecutarse en cuanto elija Invocar en la nube en el explorador. El resultado se muestra en la pestaña Salida del terminal de VS Code.

5. Elija el botón Remote Invoke para invocar la función. El resultado se muestra en la pestaña Salida del terminal de VS Code.

Eliminación de una función de Lambda

Para eliminar una función de Lambda, realice el procedimiento siguiente.

 Warning

No utilice este procedimiento para eliminar las funciones de Lambda asociadas a [CloudFormation](#). Estas funciones deben eliminarse a través de la pila de CloudFormation .

1. Desde AWS Toolkit for Visual Studio Code, expande el AWS explorador.
2. Desde el AWS explorador, expanda Lambda para ver sus recursos de Lambda.

3. Haga clic con el botón derecho en la función de Lambda que desee eliminar y, a continuación, elija Eliminar.
4. Cuando se le solicite, confirme que desea eliminar la función.

Una vez eliminada la función, dejará de figurar en el AWS explorador.

Descarga de una función de Lambda

Puede descargar código de una función de Lambda remota en su espacio de trabajo de VS Code para editarla y depurarlo.

Note

Para descargar su función Lambda, debe trabajar en un espacio de trabajo de VS Code con una carpeta accesible y el AWS kit de herramientas solo admite esta función con funciones Lambda que utilicen tiempos de ejecución de Node.js y Python.

1. Desde, expanda el AWS Toolkit for Visual Studio Code explorador. AWS
2. Desde el AWS explorador, expanda Lambda para ver sus recursos de Lambda.
3. Haga clic con el botón derecho en la función de Lambda que desee descargar y, a continuación, elija Descargar.
4. La función Lambda se abre en el editor de VS Code y se muestra en el AWS explorador cuando se completa la descarga. El AWS kit de herramientas también crea una configuración de inicio en el panel de ejecución de VS Code que le permite ejecutar y depurar la función Lambda localmente con AWS Serverless Application Model. Para obtener más información sobre su uso AWS SAM, consulte. [the section called “Ejecutar y depurar una aplicación sin servidor desde una plantilla \(local\)”](#)

Implementación de actualizaciones para nuevas funciones de Lambda

Puede implementar actualizaciones en las nuevas funciones de Lambda desde una ubicación temporal no especificada de su máquina local.

Note

Cuando haya cambios sin implementar en sus archivos de Lambda, recibirá una notificación mediante el ícono M ubicado junto a los archivos modificados en el editor de VS Code y en el Explorador de AWS .

Implementación desde el editor de VS Code

1. Abra uno de los archivos de su función de Lambda en el editor de VS Code y, a continuación, realice un cambio en el archivo.
2. Guárdelo de forma manual desde el menú principal de VS Code, o bien pulse **option+s** (Mac) o **ctrl+s** (Windows).
3. VS Code le pregunta de forma automática si desea implementar los cambios en la nube. Elija el botón Implementar para confirmar la implementación.
4. VS Code le informa sobre el estado de la implementación y le envía una notificación una vez completado el proceso.

Implementación desde el AWS explorador

1. Abra uno de los archivos de su función de Lambda en el editor de VS Code y, a continuación, realice un cambio en el archivo.
2. Desde el AWS kit de herramientas, expanda el AWS explorador.
3. Desde el AWS explorador, expanda la AWS región con la función Lambda en la que desee implementar los cambios.
4. Desde la AWS región, expanda Lambda y navegue por la función para la que desee implementar los cambios.
5. En el menú rápido situado junto a la función, elija el ícono Guardar e implementar el código.
6. VS Code le informa sobre el estado de la implementación y le envía una notificación una vez completado el proceso.

Carga de actualizaciones para las funciones de Lambda existentes

En los procedimientos siguientes se describe cómo cargar los cambios locales realizados en las funciones de Lambda existentes. Esta característica admite cargas con cualquier tiempo de ejecución compatible con Lambda.

Warning

Antes de cargar la función de Lambda, tenga en cuenta lo siguiente:

- La actualización del código de esta manera no utiliza la AWS SAM CLI para la implementación ni crea una CloudFormation pila
- El AWS kit de herramientas no valida el código. Valide el código y pruebe las funciones antes de cargar los cambios en la nube.

Carga de un archivo Zip

1. Desde AWS Toolkit for Visual Studio Code, expande el AWS explorador.
2. Desde el AWS explorador, expanda Lambda para ver sus recursos de Lambda.
3. Haga clic con el botón derecho en la función de Lambda en la que desee cargar los cambios y, a continuación, elija Cargar Lambda... para abrir el menú Seleccionar tipo de carga.
4. Elija Archivo ZIP para localizar el ZIP Archive en el directorio local.
5. Cuando se le solicite, confirme la carga para iniciar la carga del ZIP Archive seleccionado.
6. El estado de la carga se muestra en VS Code y recibirá una notificación una vez completado el proceso de carga.

Carga de un directorio sin compilación

1. Desde AWS Toolkit for Visual Studio Code, expande el AWS explorador.
2. Desde el AWS explorador, expanda Lambda para ver sus recursos de Lambda.
3. Haga clic con el botón derecho en la función de Lambda en la que desee cargar los cambios y, a continuación, elija Cargar Lambda... para abrir el menú Seleccionar tipo de carga.
4. Elija Directorio para ir a la pantalla Directorio de compilación.
5. En la pantalla Directorio de compilación, seleccione No con el fin de elegir un directorio local para la carga.

6. Cuando se le solicite, confirme la carga para cargar el directorio seleccionado.
7. El estado de la carga se muestra en VS Code y recibirá una notificación una vez completado el proceso de carga.

Carga de un directorio con una compilación

Note

Tenga en cuenta lo siguiente:

- Este procedimiento requiere la AWS Serverless Application Model CLI.
- El AWS kit de herramientas le notifica que no se puede detectar un controlador coincidente antes de cargarlo.
- Para cambiar el controlador adjunto a la función Lambda, utilice AWS Lambda console el o el AWS Command Line Interface

1. Desde AWS Toolkit for Visual Studio Code, expanda el AWS explorador.
2. Desde el AWS explorador, expanda Lambda para ver sus recursos de Lambda.
3. Haga clic con el botón derecho en la función de Lambda en la que desee cargar los cambios y, a continuación, elija Cargar Lambda... para abrir el menú Seleccionar tipo de carga.
4. Elija Directorio para ir a la pantalla Directorio de compilación.
5. En la pantalla Directorio de compilación, elija Sí y, a continuación, seleccione un directorio local para la carga.
6. Cuando se le solicite, confirme la carga para empezar a compilar y cargar el directorio seleccionado.
7. El estado de la carga se muestra en VS Code y recibirá una notificación una vez completado el proceso de carga.

Conversión de la función Lambda en un proyecto AWS SAM

Para convertir la función Lambda en una AWS SAM pila, complete los siguientes pasos.

Warning

Actualmente, solo se admite un subconjunto de recursos al convertir una función de Lambda en un proyecto de AWS SAM . Para localizar los recursos que faltan después de una conversión, consulte la consola Lambda y agréguelos manualmente a la plantilla AWS SAM . Para obtener más detalles sobre los recursos compatibles y no compatibles, consulte el tema [Tipo de recurso compatible](#) de la Guía para desarrolladores de AWS CloudFormation .

1. En el AWS kit de herramientas, expanda el AWS explorador.
2. Desde el AWS explorador, expanda la AWS región con la función Lambda que desee convertir en un AWS SAM proyecto.
3. Desde la AWS región, expanda Lambda y navegue por la función que desee convertir en una AWS SAM pila.
4. En el menú rápido situado junto a la función de Lambda, elija el ícono Convertir a aplicación de SAM a fin de navegar por el sistema de archivos local y especificar una ubicación para el nuevo proyecto de AWS SAM .
5. Tras especificar una ubicación, el AWS kit de herramientas comienza a convertir la función Lambda en AWS SAM un proyecto, VS Code proporciona actualizaciones sobre el estado del proceso.

Note

Este proceso puede demorar unos minutos.

6. Cuando se lo solicite VS Code, ingrese un nombre de pila y, a continuación, presione la tecla **Enter** para continuar.
7. VS Code continúa informándole sobre el estado de su proyecto, luego le notifica cuando el proceso se ha completado y abre su nuevo AWS SAM proyecto como un espacio de trabajo de VS Code.

AWS Lambda console a IDE

La función AWS Lambda console al IDE le permite descargar sus AWS Lambda funciones de un código AWS Lambda console a VS Code. Al trabajar con sus funciones de Lambda en VS Code,

tendrá acceso a otras opciones de desarrollo local, como AWS Serverless Application Model (AWS SAM) y AWS Cloud Development Kit (AWS CDK).

Para obtener más información al respecto AWS Lambda, consulte la [AWS Lambda Guía para desarrolladores](#). Para empezar a trabajar con la función Lambda en el AWS kit de herramientas, consulte el tema [Trabajo con AWS Lambda funciones](#) de esta guía del usuario. En las secciones siguientes se describe cómo pasar el flujo de trabajo de la consola de Lambda a VS Code. Para obtener información detallada sobre cómo pasar las funciones de Lambda de la consola de Lambda a VS Code, incluida la forma de empezar a trabajar con la consola de Lambda, consulte el tema [Desarrollo local de funciones de Lambda con VS Code](#) de la Guía para desarrolladores de AWS Lambda .

Paso de la consola al desarrollo local

Para abrir una función de Lambda de la consola de Lambda en VS Code, complete los siguientes pasos:

1. En el navegador web, abra la [consola de Lambda](#).
2. En la consola de Lambda, elija la función que desea abrir en VS Code.
3. En la vista de la función, vaya a la pestaña Código fuente.
4. En la pestaña Código fuente, elija Abrir en VS Code.

Uso de la función de Lambda en VS Code

Cuando la función de Lambda se abre en VS Code a través de la consola de Lambda:

- VS Code se inicia automáticamente en la máquina local.
- La función de Lambda se abre como espacio de trabajo de VS Code.
- El `handler` file de Lambda se abre en el editor de VS Code.

 Note

Si no hay un `handler` file correctamente configurado en el espacio de trabajo, no se abrirá ningún archivo en el editor de VS Code.

Abrir la función Lambda en VS Code a través de la consola Lambda le permite acceder a todas las funciones AWS Lambda del kit de herramientas existentes, incluida la posibilidad de editar el código

de la función con compatibilidad lingüística completa, pruebas locales, depuración remota, soporte de implementación y administración de dependencias. Para obtener más información sobre las funciones de Lambda compatibles con el AWS kit de herramientas, consulte la tabla de contenido de [AWS Lambda](#) servicios de esta guía del usuario.

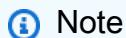
AWS Lambda con LocalStack apoyo

Cree, pruebe y depure sus aplicaciones sin servidor con el LocalStack soporte de. AWS Toolkit for Visual Studio Code LocalStack es un emulador AWS en la nube que permite realizar pruebas locales de aplicaciones sin servidor.

Para obtener más información al respecto AWS Lambda, consulta la Guía para [AWS Lambda](#) desarrolladores. Para obtener más información LocalStack, visite su sitio web [LocalStack](#).

Requisitos previos

Los siguientes son requisitos previos para trabajar con VS LocalStack Code.



Note

La LocalStack CLI se instala durante el proceso de configuración, pero si prefiere una versión diferente de la LocalStack CLI, la versión mínima requerida es la 4.8.0.

- Se requiere una cuenta de aplicación LocalStack web para acceder a todas las funciones disponibles en los niveles gratuitos y de pago LocalStack . LocalStack La edición comunitaria está disponible sin necesidad de una cuenta.
- Se requiere Docker para funcionar LocalStack en VS Code. Para obtener más información sobre LocalStack los requisitos de Docker, consulta el tema [Imágenes de LocalStack Docker](#) en la documentación. LocalStack
- Recomendado: El AWS Command Line Interface (AWS CLI) le ayuda a trabajar con los servicios de su entorno de nube simulado.

¿Instalando LocalStack

Para instalar versiones escalonadas LocalStack gratuitas y de pago, complete los siguientes pasos.

Note

Para obtener instrucciones sobre cómo configurar LocalStack la edición comunitaria, consulte el contenido de la LocalStack comunidad en la LocalStack sección de configuración de este tema.

1. En el AWS kit de herramientas, expanda el explorador de APPLICATION BUILDER.
2. Elija el botón Abrir tutorial para abrir la pestaña del tutorial Get started building your application en el editor de VS Code.
3. En el tutorial, elija Instalar LocalStack para iniciar el proceso de LocalStack instalación en VS Code.

Con LocalStack configuración

Después de instalar la LocalStack extensión para VS Code, es posible que veas uno de los siguientes indicadores cuando necesites configurarla:

- En la barra de estado de VS Code, ubicada de forma predeterminada en la esquina inferior izquierda del IDE, el LocalStack estado es rojo.
- VS Code te pide que lo configures. LocalStack

Hay dos tipos de configuración y configuración LocalStack, según la versión que LocalStack utilices. En las siguientes secciones con pestañas se describe cada proceso LocalStack de configuración.

Note

LocalStack Los tokens de autenticación son necesarios para las versiones de nivel gratuito y de pago de LocalStack. Para obtener información específica sobre LocalStack los precios, consulta su guía de precios [Elige tu plan](#).

LocalStack niveles gratuitos y de pago

Hay 2 formas de configurarlos LocalStack.

- En el mensaje Configuración de VS Code LocalStack para comenzar, elija el botón Configuración.

- En la barra de estado de VS Code, selecciona el icono de LocalStack estado para abrir el mensaje Configuración LocalStack para empezar y, a continuación, pulsa el botón Configuración.

Durante la configuración, el sistema realiza los siguientes pasos:

1. Instala la LocalStack CLI.
2. Comprueba si tiene una LocalStack cuenta.
3. Si tienes una LocalStack cuenta, el sistema te guía a través del proceso de autenticación en tu navegador web predeterminado. Del mismo modo, si no tiene una LocalStack cuenta, el sistema le guiará durante la configuración de la cuenta antes del proceso de autenticación.

Una vez LocalStack configurada, el LocalStack estado se actualiza en la barra de estado de VS Code.

 Note

Si no has creado un AWS perfil para el LocalStack, se creará automáticamente uno nuevo como parte del proceso de LocalStack configuración.

LocalStack Comunidad

La edición Community de LocalStack es de uso gratuito y no requiere que te registres para obtener una cuenta, sino que se ejecuta a partir de una imagen de Docker que no requiere licencia. Para obtener más información sobre LocalStack la Community Edition, consulta la documentación de [imágenes de la LocalStack comunidad](#). En las siguientes secciones se describen los requisitos previos y la configuración básica necesaria para trabajar con LocalStack la edición comunitaria en VS Code.

Lanzamiento de una nueva instancia

Para lanzar una nueva instancia de LocalStack Community, complete el siguiente procedimiento.

 Note

En el siguiente ejemplo, se inicia una instancia de contenedor LocalStack en el puerto 4566. Si especifica valores de puerto diferentes, debe actualizar el valor de puerto especificado

en el procedimiento que se encuentra en la sección Configuración del AWS CLI AWS kit de herramientas.

1. En VS Code, pulse **ctrl + `(backtick)** para abrir el terminal.
2. Una vez abierto, especifique lo siguiente.

Mac:

```
docker run -d --name localstack_main \
>> -p 4566:4566 \
>> -v /var/run/docker.sock:/var/run/docker.sock \
>> localstack/localstack
```

Windows:

```
docker run -d --name localstack_main \
>> -p 4566:4566 \
>> -v /var/run/docker.sock:/var/run/docker.sock \
>> localstack/localstack
```

3. El terminal se actualiza con el estado de la instancia de Docker una vez completado el proceso.

Esta instancia contenerizada de LocalStack le da acceso a los AWS servicios que especificó durante el proceso de descarga.

Configuración de la CLI para LocalStack y Docker.

Para configurar el kit de AWS herramientas AWS CLI y el kit de herramientas para que funcionen LocalStack en Docker, configure un nuevo perfil siguiendo estos pasos:

1. En VS Code, pulse **ctrl + `(backtick)** para abrir el terminal.
2. Una vez abierto, especifique lo siguiente.

```
~/.aws/credentials
[localstack]
aws_access_key_id = test
aws_secret_access_key = test
~/.aws/config
[profile localstack]
```

```
region = us-east-1
output = json
endpoint_url = http://localhost:4566 [default localstack endpoint]
```

3. El AWS kit de herramientas detecta su LocalStack perfil y actualiza el menú de estado de la conexión.

Tras la configuración, al elegir su LocalStack perfil en la sección de AWS perfiles de la barra de estado, sus LocalStack recursos estarán visibles en el AWS explorador. Además, puede ver sus LocalStack registros en la pestaña Salida del terminal VS Code.

Empezando LocalStack en VS Code

Puedes empezar a LocalStack usar cualquiera de los siguientes métodos:

Empezando LocalStack desde la barra de estado de VS Code

1. Desde VS Code, navega hasta la barra de estado y, a continuación, selecciona el LocalStack botón de inicio para iniciar LocalStack.
2. La barra de estado de VS Code se actualiza cuando LocalStack se inicia correctamente.

Empezando LocalStack desde la paleta de comandos de VS Code

1. En VS Code, pulse **Cmd + Shift + P** (Mac) o **Control + Shift + P** (Windows) para abrir la paleta de comandos.
2. En la paleta de comandos, escriba **Start LocalStack** en la barra de búsqueda y elija esta opción en la lista cuando aparezca en los resultados.
3. La barra de estado de VS Code se actualiza cuando LocalStack se inicia correctamente.

Empezando LocalStack desde el terminal VS Code

1. En VS Code, pulse **ctrl + `(backtick)** para abrir el terminal.
2. En el terminal de VS Code, escriba el comando **localstack start** de la CLI.
3. La barra de estado de VS Code se actualiza cuando LocalStack se inicia correctamente.

Creación de una aplicación sin servidor de muestra

Para empezar a trabajar con LocalStack VS Code, necesitas un ejemplo de aplicación sin servidor. Si ya tiene una aplicación existente en su AWS cuenta, puede implementarla localmente usando Serverless Land LocalStack o puede crear una nueva aplicación con AWS Serverless Land.

Para obtener más información sobre cómo crear una aplicación con Serverless Land en el kit de herramientas de AWS , consulte el tema [Uso de AWS Serverless Land](#) de esta Guía del usuario. Para obtener información detallada sobre Serverless Land, consulte la página de inicio principal de la aplicación web de [Serverless Land](#).

Prueba y depuración de funciones Lambda con LocalStack

Probar y depurar las funciones de Lambda en LocalStack la extensión VS Code es similar a trabajar con las funciones desplegadas en la nube. AWS La principal diferencia es que su instancia de AWS Toolkit debe estar autenticada con su LocalStack cuenta para poder implementar y depurar sus funciones. LocalStack

Note

Las funciones de prueba y depuración descritas en esta sección no están disponibles para la edición Community. LocalStack

Para trabajar con ellas LocalStack en VS Code, conéctate a tu LocalStack perfil en el AWS kit de herramientas. Cuando tu LocalStack perfil está activo, la barra de estado de VS Code muestra AWS: profile:localstack (punto final personalizado) con una marca de verificación.

Para obtener información detallada sobre cómo trabajar con las funciones de Lambda en el AWS kit de herramientas, consulte el tema [Trabajo con AWS Lambda funciones](#) de esta guía del usuario.

AWS Lambda depuración remota

AWS Toolkit for Visual Studio Code Esto le permite depurar AWS Lambda las funciones que se ejecutan en la nube, directamente en VS Code. Con la depuración AWS Lambda remota, puede inspeccionar las funciones en ejecución, establecer puntos de interrupción, examinar variables y realizar la depuración gradual sin modificar su flujo de trabajo de desarrollo actual.

En las secciones siguientes se describe cómo trabajar con la depuración remota de Lambda en el AWS Toolkit for Visual Studio Code.

Cómo funciona la depuración remota de Lambda

El AWS kit de herramientas permite la depuración remota modificando temporalmente las funciones de Lambda con una capa de depuración de Lambda adicional y ampliando el límite de tiempo de espera de invocación de Lambda a 900 segundos. Se establece una conexión segura entre el depurador local y el entorno de tiempo de ejecución Lambda mediante la opción de Túnel seguro de AWS IoT . Esta conexión permite utilizar los puntos de interrupción del código local para procesar la función a medida que se ejecuta de forma remota. Una vez finalizada la sesión de depuración, todas las modificaciones temporales se revierten automáticamente a su configuración original.

Introducción

Tiempos de ejecución admitidos

La depuración remota de Lambda es compatible con los siguientes tiempos de ejecución.

- Python (Amazon Linux 2023)
- Java
- JavaScript/Node.js (Amazon Linux 2023)

Requisitos previos

Antes de comenzar, se deben cumplir los siguientes requisitos previos:

- Debe tener AWS credenciales válidas configuradas en el AWS kit de herramientas. Para obtener información adicional sobre la instalación del AWS kit de herramientas y la configuración de sus credenciales, consulte el tema [Introducción](#) de esta guía del usuario.
- Se ha implementado una función Lambda en su AWS cuenta. Para obtener más información sobre la implementación de una función de Lambda, consulte el tema [Creación de su primera función de Lambda](#) de la Guía para desarrolladores de AWS Lambda.
- Debe tener la política AWS Identity and Access Management (de IAM) y los permisos adecuados para depurar su función. Para obtener más información sobre los permisos de Lambda, consulte el tema [Políticas administradas de AWS para AWS Lambda](#) de la Guía para desarrolladores de AWS Lambda. A continuación, se muestra un ejemplo de una política que contiene los permisos mínimos necesarios para trabajar con la depuración remota de Lambda en el kit de herramientas de AWS .

Note

La depuración remota se habilita mediante AWS AWS IoT Secure Tunneling. Esta opción permite que el depurador local establezca una conexión segura con el entorno de tiempo de ejecución Lambda.

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "lambda>ListFunctions",  
        "lambda>GetFunction",  
        "lambda>GetFunctionConfiguration",  
        "lambda>GetLayerVersion",  
        "lambda>UpdateFunctionConfiguration",  
        "lambda>InvokeFunction",  
        "lambda>PublishVersion",  
        "lambda>DeleteFunction",  
        "iot:CloseTunnel",  
        "iot:OpenTunnel",  
        "iot:RotateTunnelAccessToken",  
        "iot>ListTunnels"  
      ],  
      "Resource": "*"  
    }  
  ]  
}
```

Acceso a la depuración remota de Lambda

Existen dos rutas principales para acceder a la depuración remota de Lambda en el AWS kit de herramientas: el AWS explorador o el explorador de Application Builder. Desde el AWS explorador, puede acceder a la depuración remota de Lambda a través de sus nodos. AWS Lambda Desde el

explorador Desarrollador de aplicaciones, puede acceder a la depuración remota de Lambda a través de los proyectos de AWS SAM locales.

Acceso a la depuración remota de Lambda desde el explorador AWS

1. En VS Code, abra la extensión AWS Toolkit.
2. Desde el AWS kit de herramientas, expande el AWS explorador.
3. En el explorador, expanda el nodo de Lambda.
4. Vaya a la función que desee depurar y, a continuación, elija el icono Invocar remotamente en el menú contextual para abrir la pantalla Configuración de invocación remota.

Acceso a la depuración remota de Lambda desde el explorador Desarrollador de aplicaciones

1. En VS Code, abre la extensión AWS Toolkit.
2. Desde el AWS kit de herramientas, expande el explorador del generador de aplicaciones.
3. En el explorador, expanda el proyecto de AWS SAM que contiene el proyecto de Lambda que quiere depurar.
4. Expanda la función de Lambda implementada que desea depurar.
5. Vaya a la función remota y, a continuación, elija el icono Invocar remotamente en el menú contextual para abrir la pantalla Remote invoke configuration.

Uso de la depuración remota de Lambda

En las secciones siguientes se describe cómo trabajar con la depuración remota de Lambda en el AWS Toolkit for Visual Studio Code.

Note

Las funciones de Lambda tienen un límite de cinco capas y un límite combinado de 250 MB para el código de la función y todas las capas adjuntas. Para ejecutarse, la depuración remota de Lambda necesita disponer de al menos una capa libre.

Configuración de una sesión de depuración

Antes de empezar, configure la sesión de depuración mediante el procedimiento que se indica a continuación:

1. Abra el menú de configuración de invocación remota completando el procedimiento de depuración remota de Lambda desde AWS el explorador o el procedimiento de depuración remota de Acceso a Lambda desde el explorador de Application Builder, que se encuentra en la sección anterior.
2. En el menú Configuración de invocación remota, seleccione la casilla Depuración remota para mostrar las propiedades correspondientes a esta.
3. Especifique la Ruta raíz local al archivo de controlador local.

 Note

La ruta raíz local es la ubicación de su código fuente que coincide con la función de Lambda implementada. Si trabaja desde una función implementada en el explorador Desarrollador de aplicaciones, la ruta raíz local se detecta de forma automática. Si no tiene el código fuente almacenado de forma local, elija el botón Descargar código remoto para recuperar el código fuente de la función de Lambda. Al hacerlo, se abre el `handler file` en el editor de VS Code.

4. En la sección Payload, especifique dónde se obtienen los datos del evento de prueba.

Establecimiento de puntos de interrupción y depuración

Establezca puntos de interrupción e inicie la depuración mediante el siguiente procedimiento:

1. En el `handler file` del editor de VS Code, haga clic en el margen interior para establecer puntos de interrupción en los números de línea donde quiera poner la depuración en pausa.
2. Cuando esté conforme con los puntos de interrupción, vuelva al menú `Remote invoke configuration` para comprobar que la configuración de sus ajustes es correcta y, a continuación, elija el botón `Remote invoke` para iniciar la depuración.
3. El AWS kit de herramientas actualiza la función Lambda con capacidades de depuración, establece un túnel seguro para la sesión de depuración, invoca la función con la carga útil especificada y, a continuación, detiene el proceso cuando alcanza un punto de interrupción.
4. En una pausa de punto de interrupción use el panel Ejecución y depuración para ver sus Variables, Pila de llamadas y Puntos de interrupción.

Actualización y pruebas de la función

Para modificar el código y probar los cambios con una implementación rápida, realice el procedimiento siguiente:

1. Con la sesión de depuración activa, realice cambios en el `handler file` del editor de VS Code.
2. Guarde los cambios (**Command+S on macOS, Ctrl+S on Windows**).
3. Cuando se le solicite, confirme que desea continuar con la implementación de los cambios. El AWS kit de herramientas actualizará la función Lambda con el código modificado.
4. Siga depurando y probando los cambios mediante el establecimiento de nuevos puntos de interrupción y la selección del botón Invocación remota de nuevo.

 Note

Como alternativa, puede anular la selección de la opción Attach debugger en los controles de depuración de VS Code y elegir el botón Remote invoke para ejecutar la función sin depurar.

Finalización de una sesión de depuración

Cada una de las siguientes opciones finaliza la sesión de depuración remota y elimina la capa de depuración del proyecto:

- Seleccione la opción Remove Debug Setup en la pantalla Remote invoke configuration.
- Seleccione el ícono de desconexión en los controles de depuración de VS Code.
- Cierre el `handler file` en el editor de VS Code.

 Note

Tome nota de lo siguiente:

- La capa de depuración de Lambda se elimina automáticamente después de 60 segundos de inactividad. El tiempo empieza a contar tras la última invocación.
- Si realizó cambios de código en sus funciones gestionadas infrastructure-as-code (IaC) (AWS SAM, AWS CDK, Terraform) durante el proceso de depuración, guárdelos en su

proyecto local y considere la posibilidad de actualizar su repositorio de control de código fuente. Los cambios no guardados se sobrescriben cuando se vuelve a implementar la función laC.

- Si ha realizado cambios temporales solo con fines de depuración, puede que quiera volver a implementar la función desde el control de origen para asegurarse de que coincide con el código de producción.

Solución de problemas y casos de uso avanzados

Si se produce un error en la sesión de depuración, inicie el proceso de solución de problemas mediante estos pasos:

1. Actualice el AWS kit de herramientas a la última versión.
2. Actualice la vista web; para ello, cierre la vista web Configuración de invocación remota y vuelva a abrirla.
3. Cierre VS Code por completo y vuelva a abrirlo a fin de reiniciarlo.
4. Abra la paleta de comandos de VS Code, escriba el comando **AWS: Reset Lambda Remote Debugging Snapshot** y selecciónelo cuando aparezca en los resultados para restablecer la instantánea de depuración remota de Lambda.
5. [Si no puede solucionar el problema, envíe un problema a AWS Toolkit for Visual Studio Code GitHub Problemas.](#)

Caso de uso avanzado: configuración de firma de código

La depuración remota requiere adjuntar una capa de depuración a la función de Lambda. Si la función tiene habilitada y aplicada la configuración de firma de código, el AWS kit de herramientas no podrá adjuntar automáticamente la capa de depuración a la función.

Existen dos opciones para resolver el problema de configuración de firma de código.

- Elimine temporalmente la firma de código.
- Use una capa de depuración firmada.

Eliminación temporal de la firma de código

Actualice la configuración de firma de código mediante el establecimiento de `UntrustedArtifactOnDeployment : Warn` y, a continuación, vuelva a habilitarla como `Enforced` una vez completado el proceso de depuración.

Para obtener más información, consulta la [UpdateCodeSigningConfig](#) referencia en la Referencia de la API AWS Lambda

Uso de una capa de depuración firmada

1. Desde Depuración remota de Lambda en el AWS kit de herramientas, amplíe la sección de configuración adicional de Depuración remota.
2. En la sección Configuración adicional de depuración remota, copie el nombre de recurso de Amazon (ARN) de la capa de región del campo Anulación de capa.
3. Desde AWS CLI, utilice el siguiente comando para descargar la versión `aws lambda get-layer-version-by-arn --arn layer-arn` de la capa y sustituya `layer-arn` por el ARN de su capa. Para obtener instrucciones detalladas sobre cómo descargar la capa de depuración firmada, consulte la referencia [get-layer-version-by-arn en la Referencia de comandos AWS CLI](#).
4. Firme la capa con su configuración de firma de código y publíquela en su cuenta. Para obtener instrucciones sobre la firma y la publicación, consulta el tema [Cómo configurar la firma de código para tu AWS SAM aplicación](#) en la AWS Serverless Application Model Guía para desarrolladores.
5. Una vez que la capa se haya firmado y publicado en su cuenta, vuelva a la sección Configuración adicional de depuración remota de la depuración remota de Lambda y, a continuación, escriba el nuevo ARN de la capa en el campo Anulación de capa. Una vez completado el proceso, la depuración remota de Lambda utiliza la capa firmada en lugar de la predeterminada.

Regiones admitidas

El siguiente error se produce cuando una región no admite la depuración remota.

Region \${region} doesn't support remote debugging yet

A continuación se muestra una lista de las regiones admitidas:

- ap-east-1

- ap-northeast-1
- ap-northeast-2
- ap-south-1
- ap-southeast-1
- ap-southeast-2
- ca-central-1
- eu-central-1
- eu-north-1
- eu-west-1
- eu-west-2
- eu-west-3
- me-central-1
- me-south-1
- sa-east-1
- us-east-1
- us-east-2
- us-west-1
- us-west-2

Lambda RequestEntityTooLargeException

Las funciones de Lambda tienen un límite de cinco capas y un límite combinado de 250 MB para el código de la función y todas las capas adjuntas. La capa de depuración remota tiene aproximadamente 40 MB, lo que puede hacer que la función supere este límite si tiene un paquete de funciones de gran tamaño o varias capas. Para obtener más información, consulte la sección [Lambda: InvalidParameterValueException o RequestEntityTooLargeException](#) tema de la Guía para AWS Lambda desarrolladores.

En la lista siguiente se describen distintas formas de solucionar y corregir este error:

- Reducir el tamaño de la función: optimice el código de la función y elimine las dependencias innecesarias.

- Eliminar las capas no utilizadas: elimine de forma temporal las capas no esenciales durante la depuración.
- Utilizar dependencias externas: mueva las dependencias grandes a un almacenamiento externo, como Amazon S3, y cárguelas en tiempo de ejecución.

Superación de la cuota del túnel seguro de IoT

A continuación, se muestra un ejemplo del error de superación de la cuota de túnel que se produce cuando se alcanza el límite diario de conexiones de tunelización AWS IoT seguras en la depuración remota de Lambda.

```
Error creating/reusing tunnel: LimitExceededError: Exceeded quota of Lambda
debugging tunnels
```

AWS IoT Las conexiones de Secure Tunneling tienen las siguientes cuotas:

- El túnel seguro de IoT de capa gratuita tiene asignadas diez conexiones al día.
- Cada túnel admite una instancia de VS Code durante un máximo de doce horas.
- La cuota se aplica por AWS cuenta y por día.

Si se produce el error de tunelización AWS IoT segura, espera a que se restablezca la cuota diaria o ponte en contacto con el servicio de AWS asistencia para solicitar un aumento del límite de cuota.

[Para obtener la información AWS de contacto de soporte, consulta el portal de contacto de AWS soporte.](#) Para obtener información detallada sobre la tunelización AWS IoT segura, consulte el tema de la [tunelización AWS IoT segura](#) en la Guía para desarrolladores AWS IoT

Depuración de funciones TypeScript Lambda con mapas de origen

En las siguientes secciones se describe cómo depurar las funciones de TypeScript Lambda con mapas de origen.

Requisitos previos

Para depurar las funciones de TypeScript Lambda, se deben cumplir los siguientes requisitos previos.

- TypeScript Debe cumplir con la opción de mapa de origen habilitada. Para obtener información adicional, consulte el tema de [soporte del mapa JavaScript fuente](#) en la documentación de VS Code.

- Para trabajar con mapas de origen en línea, todos los cambios deben volver a implementarse en la nube.

Configuración

Para configurar la depuración remota de Lambda para las funciones de TypeScript Lambda en el AWS kit de herramientas, complete los siguientes pasos.

1. Desde el AWS kit de herramientas, expanda el explorador. AWS
2. En el explorador, expanda el nodo de Lambda.
3. Navegue hasta la función para la que desee configurar y, a continuación TypeScript, seleccione el ícono de invocación remota en el menú contextual para abrir la pantalla de configuración de la invocación remota.
4. Seleccione la casilla Depuración remota para habilitar la configuración remota.
5. Apunte al directorio que contiene su TypeScript `handler file` a fin de configurar la ruta raíz local.

 Note

En el TypeScript `handler file` se establecen los puntos de interrupción de la depuración.

6. Expanda la opción de Configuración adicional de depuración remota.
7. Seleccione la casilla Mapa de origen para habilitar los mapas de origen.
8. Establezca el campo Archivos de salida en el directorio local de la copia de la función de Lambda.

Example

Si `app.js` y `app.map` están en `.aws-sam/build/HelloWorldFunction`, establezca la ubicación de los archivos de salida en `/Users/user/project/aws-sam/build/HelloWorldFunction/*`.

 Note

La ruta del archivo de salida debe ser absoluta.

9. Cuando esté satisfecho con los ajustes, pulse el botón de invocación remota para empezar a depurar la función. TypeScript

Amazon Redshift en el kit de herramientas para VS Code

Amazon Redshift es un servicio de almacenamiento de datos administrado de varios petabytes en la nube. Para obtener información detallada sobre el servicio Amazon Redshift, consulte las tablas de contenido de la Guía del usuario de [Amazon Redshift](#).

En los temas siguientes se describe cómo trabajar con Amazon Redshift desde el AWS Toolkit for Visual Studio Code.

Temas

- [Uso de Amazon Redshift desde el kit de herramientas para VS Code](#)

Uso de Amazon Redshift desde el kit de herramientas para VS Code

En las secciones siguientes se describe cómo empezar a trabajar con Amazon Redshift desde el AWS Toolkit for Visual Studio Code.

Para obtener información detallada sobre el servicio Amazon Redshift, consulte los temas de la Guía del usuario de [Amazon Redshift](#).

Introducción

Antes de empezar a trabajar con Amazon Redshift desde el AWS Toolkit for Visual Studio Code, deben cumplirse los siguientes requisitos.

1. Debe estar conectado a sus cuentas de AWS desde el kit de herramientas. Para obtener información adicional sobre cómo conectarse a su cuenta de AWS desde el kit de herramientas, consulte el tema [Conexión a AWS](#) en esta Guía del usuario.
2. Ha creado un almacenamiento de datos aprovisionado o sin servidor.

Si aún no ha creado un clúster de Amazon Redshift sin servidor o un clúster aprovisionado de Amazon Redshift, en los siguientes procedimientos se describe cómo crear un almacenamiento de datos con un conjunto de datos de muestra desde la Consola de AWS.

Crear un almacenamiento de datos aprovisionado

Para obtener más información sobre la creación de almacenamientos de datos de clústeres aprovisionados de Amazon Redshift, consulte el tema [Crear un clúster de Amazon Redshift de muestra](#) en la Guía de introducción a Amazon Redshift.

1. En su navegador de Internet preferido, inicie sesión en la Consola de administración de AWS y abra la consola de Amazon Redshift en <https://console.aws.amazon.com/redshift/>.
2. En la consola de Amazon Redshift, seleccione Panel de clústeres aprovisionados.
3. En Panel de control de clústeres aprovisionados, pulse el botón Crear clúster para abrir el panel Crear el clúster.
4. Rellene los campos obligatorios en la sección Configuración del clúster.
5. En la sección Datos de muestra, seleccione la casilla Cargar datos de muestra para cargar el conjunto de datos de muestra **Tickit** en la base de datos predeterminada **Dev** con el esquema **public**.
6. En la sección Configuraciones de la base de datos, especifique valores en los campos Nombre de usuario del administrador y Contraseña de usuario administrador.
7. Elija Crear clúster para crear el almacenamiento de datos aprovisionado.

Crear un almacenamiento de datos sin servidor

Para obtener más información sobre la creación de almacenamientos de datos de Amazon Redshift sin servidor, consulte el tema [Crear un almacenamiento de datos de Amazon Redshift sin servidor](#) en la Guía de introducción a Amazon Redshift.

1. En su navegador de Internet preferido, inicie sesión en la Consola de administración de AWS y abra la consola de Amazon Redshift en <https://console.aws.amazon.com/redshift/>.
2. En la consola de Amazon Redshift, pulse el botón Pruebe Amazon Redshift sin servidor para abrir el panel Comience a utilizar Amazon Redshift sin servidor.
3. En la sección Configuraciones, elija el radial Usar la configuración predeterminada.
4. En la parte inferior del panel Comience a utilizar Amazon Redshift sin servidor, elija Guardar configuración para crear un almacenamiento de datos sin servidor con el grupo de trabajo, espacio de nombres, credencial y configuración de cifrado predeterminados.

Conexión a un almacenamiento de datos desde el kit de herramientas

Existen tres métodos para conectarse a una base de datos desde el kit de herramientas:

- Nombre de usuario y contraseña de la base de datos
- AWS Secrets Manager
- Credenciales temporales

Para conectarse a una base de datos ubicada en un clúster aprovisionado existente o en un almacenamiento de datos sin servidor desde el kit de herramientas, complete los siguientes pasos.

Important

Si ha completado los pasos de la sección Requisitos previos de este tema de la guía del usuario y su almacenamiento de datos no está visible en el Explorador del kit de herramientas, asegúrese de que está trabajando desde la región de AWS correcta en el explorador.

Conexión al almacenamiento de datos mediante el nombre de usuario y la contraseña de la base de datos

1. Desde el Explorador del kit de herramientas, expanda la Región de AWS donde se encuentra su almacenamiento de datos.
2. Expanda Redshift y elija su almacenamiento de datos para abrir el cuadro de diálogo Seleccione un tipo de conexión en VS Code.
3. En el cuadro de diálogo Seleccione un tipo de conexión, elija el nombre de usuario y la contraseña de la base de datos y proporcione la información requerida en cada una de las solicitudes.
4. Las bases de datos, tablas y esquemas disponibles están visibles en el Explorador del kit de herramientas cuando este último se conecta al almacenamiento de datos y se completa el procedimiento.

Conexión a su almacenamiento de datos con AWS Secrets Manager

Note

Este procedimiento requiere un secreto de base de datos de AWS Secrets Manager para completarse. Para obtener instrucciones sobre cómo configurar un secreto de base de datos, consulte [Crear un secreto de base de datos de AWS Secrets Manager](#) en la Guía del usuario de AWS Secrets Manager.

1. Desde el Explorador del kit de herramientas, expanda la Región de AWS donde se encuentra su almacenamiento de datos.
2. Expanda Redshift y elija su almacenamiento de datos para abrir el cuadro de diálogo Seleccione un tipo de conexión en VS Code.
3. En el cuadro de diálogo Seleccione un tipo de conexión, elija Secrets Manager y proporcione la información requerida en cada una de las solicitudes.
4. Las bases de datos, tablas y esquemas disponibles están visibles en el Explorador del kit de herramientas cuando este último se conecta al almacenamiento de datos y se completa el procedimiento.

Conexión a su almacenamiento de datos con credenciales temporales

1. Desde el Explorador del kit de herramientas, amplíe la región de AWS donde se encuentra su almacenamiento de datos.
2. Expanda Redshift y elija su almacenamiento de datos para abrir el cuadro de diálogo Seleccione un tipo de conexión en VS Code.
3. En el cuadro de diálogo Seleccione un tipo de conexión, elija Credenciales temporales y proporcione la información requerida en cada una de las solicitudes.
4. Las bases de datos, tablas y esquemas disponibles están visibles en el Explorador del kit de herramientas cuando este último se conecta al almacenamiento de datos y se completa el procedimiento.

Edición de la conexión a su almacenamiento de datos

Puede editar la conexión a su almacenamiento de datos para cambiar la base de datos a la que desea conectarse.

1. Desde el Explorador del kit de herramientas, expanda la Región de AWS donde se encuentra su almacenamiento de datos.
2. Expanda Redshift, haga clic con el botón derecho en el almacenamiento de datos al que está conectado, elija Editar conexión y escriba el nombre de la base de datos a la que desea conectarse.
3. Las bases de datos, tablas y esquemas disponibles están visibles en el Explorador del kit de herramientas cuando este último se conecta al almacenamiento de datos y se completa el procedimiento.

Eliminación de la conexión a su almacenamiento de datos

1. Desde el Explorador del kit de herramientas, expanda la Región de AWS donde se encuentra su almacenamiento de datos.
2. Expanda Redshift, haga clic con el botón derecho en el almacenamiento de datos con la conexión que deseé eliminar y seleccione Eliminar conexión. Al hacerlo, se eliminan las bases de datos, las tablas y los esquemas disponibles del Explorador del kit de herramientas.
3. Para volver a conectarse a su almacenamiento de datos, seleccione Haga clic para conectarse y proporcione la información requerida en cada una de las solicitudes. De forma predeterminada, la reconexión utiliza el método de autenticación anterior para conectarse al almacenamiento de datos. Para usar un método diferente, seleccione la flecha hacia atrás en el cuadro de diálogo hasta llegar a la solicitud de autenticación.

Ejecución de instrucciones SQL

En los siguientes procedimientos se describe cómo crear y ejecutar sentencias SQL en las bases de datos desde el AWS Toolkit for Visual Studio Code.

 Note

Para finalizar los pasos de cada uno de los siguientes procedimientos, primero debe completar la sección Conexión a un almacenamiento de datos desde el kit de herramientas, que se encuentra en este tema de la Guía del usuario.

1. Desde el Explorador del kit de herramientas, expanda Redshift y, a continuación, expanda el almacenamiento de datos que contiene la base de datos que desea consultar.

2. Seleccione Crear bloc de notas para especificar un nombre de archivo y una ubicación para almacenar su bloc de notas de forma local y, a continuación, haga clic en Aceptar para abrir el bloc de notas en su editor de VS Code.
3. En el editor de VS Code, introduzca las instrucciones SQL que desee almacenar en este bloc de notas.
4. Pulse el botón Ejecutar todo para ejecutar las instrucciones SQL que ha introducido.
5. El resultado de las instrucciones SQL se muestra debajo de las instrucciones que ha introducido.

Añadir marcado a un bloc de notas

1. En el bloc de notas del editor de VS Code, pulse el botón Marcado para añadir una celda de marcado a su bloc de notas.
2. Introduzca el marcado en la celda proporcionada.
3. La celda de marcado se puede editar con las herramientas de edición ubicadas en la esquina superior derecha de la celda de marcado.

Añadir código a un bloc de notas

1. En el bloc de notas del editor de VS Code, pulse el botón Código para añadir una celda de código a su bloc de notas.
2. Introduzca el código en la celda proporcionada.
3. Puede elegir ejecutar el código por encima o por debajo de la celda de código. Para ello, seleccione el botón correspondiente en las herramientas del editor de celdas, ubicadas en la esquina superior derecha de la celda de código.

Uso de Amazon S3

Amazon Simple Storage Service (Amazon S3) es un servicio de almacenamiento escalable. El AWS Toolkit for Visual Studio Code permite gestionar objetos y recursos de Amazon S3 directamente desde VS Code.

Para obtener información detallada sobre el servicio Amazon S3, consulte la Guía del usuario de [Amazon S3](#).

En las siguientes secciones se describe cómo trabajar con objetos y recursos de Amazon S3 desde el AWS Toolkit for Visual Studio Code.

Temas

- [Trabajo con recursos de Amazon S3](#)
- [Trabajo con objetos de Amazon S3](#)

Trabajo con recursos de Amazon S3

Puede utilizar Amazon S3 desde el AWS Toolkit for Visual Studio Code para ver, gestionar y editar sus buckets de Amazon S3 y otros recursos.

En las siguientes secciones se describe cómo trabajar con recursos de Amazon S3 desde el AWS Toolkit for Visual Studio Code. Para obtener información sobre cómo trabajar con objetos de Amazon S3, como carpetas y archivos, desde el AWS Toolkit for Visual Studio Code, consulte el tema [Trabajo con objetos de S3](#) de esta Guía del usuario.

Creación de un bucket de Amazon S3

1. Desde el Explorador del kit de herramientas, abra el menú contextual (clic con el botón derecho) del servicio S3 y seleccione Crear bucket.... También puede elegir el ícono Crear bucket para abrir el cuadro de diálogo Crear bucket.
2. En el campo Bucket Name (Nombre del bucket), ingrese un nombre para el bucket.

Pulse Intro para crear el bucket y cerrar el cuadro de diálogo. A continuación, el nuevo bucket se muestra en el servicio S3 del kit de herramientas.

Note

Dado que Amazon S3 permite que el bucket se utilice como una dirección URL con acceso público, el nombre del bucket que elija deberá ser único de forma global. Si otra cuenta ya ha creado un bucket con el nombre que quiere utilizar, deberá elegir otro.

Si no puede crear un bucket nuevo, revise los registros del kit de herramientas de AWS en la pestaña Salida. Si intenta usar un nombre de bucket no válido, se produce el error BucketAlreadyExists.

Para obtener más información, consulte [Restricciones y limitaciones de los buckets](#) en la Guía del usuario de Amazon Simple Storage Service.

Agregar una carpeta a un bucket de Amazon S3

Para organizar el contenido de un bucket de S3, agrupe los objetos en carpetas. También puede crear carpetas dentro de otras carpetas.

1. Desde el Explorador del kit de herramientas, expanda el servicio S3 para ver una lista de sus recursos de S3.
2. Seleccione el ícono Crear carpeta para abrir el cuadro de diálogo Crear carpeta. O abra el menú contextual (clic con el botón derecho) de un bucket o una carpeta y, a continuación, elija Crear carpeta.
3. Introduzca un valor en el campo Nombre de la carpeta y pulse Intro para crear la carpeta y cerrar el cuadro de diálogo. La nueva carpeta se muestra debajo del recurso S3 correspondiente en el menú del kit de herramientas.

Eliminación de un bucket de Amazon S3

Cuando elimina un bucket de S3, también elimina las carpetas y los objetos que contiene. Por lo tanto, cuando intente eliminar un bucket, se le pedirá que confirme que desea eliminarlo.

1. En el menú principal del kit de herramientas, expanda el servicio Amazon S3 para ver una lista de sus recursos de S3.
2. Abra el menú contextual (clic con el botón derecho) de un bucket o una carpeta y, a continuación, elija Eliminar bucket de S3.
3. Cuando se le pida, introduzca el nombre del bucket en el campo de texto y, a continuación, pulse Intro para eliminar el bucket y cerrar el mensaje de confirmación.

Note

Si su bucket contiene objetos, se vacía antes de que se elimine. Si intenta eliminar una gran cantidad de recursos u objetos a la vez, pueden tardar un tiempo en eliminarse.

Cuando se eliminan, recibirá una notificación en la que se indica que se han eliminado correctamente.

Trabajo con objetos de Amazon S3

Los archivos, carpetas y cualquier otro dato que esté almacenado en un bucket de recursos de S3 se conocen como objetos de S3.

En las siguientes secciones se describe cómo trabajar con objetos de Amazon S3 desde el AWS Toolkit for Visual Studio Code. Para obtener más información sobre cómo trabajar con los recursos de Amazon S3, como los buckets de S3, desde el AWS Toolkit for Visual Studio Code, consulte el tema [Trabajo con recursos de S3](#) en esta Guía del usuario.

Paginación de objetos

Si está trabajando con un gran número de objetos y carpetas de Amazon S3, la paginación le permite especificar el número de elementos que desea mostrar en una página.

1. Navegue hasta la barra de actividades de VS Code y seleccione Extensiones.
2. En la extensión del kit de herramientas de AWS, seleccione el ícono de configuración y, a continuación, seleccione Configuración de extensión.
3. En la página Configuración, desplácese hacia abajo hasta la configuración AWS > S3: máximo de elementos por página.
4. Cambie el valor predeterminado por el número de elementos de S3 que desea que se muestren antes de que aparezca “cargar más”.

 Note

Los valores válidos incluyen cualquier número entre 3 y 1000. Esta configuración solo se aplica al número de objetos o carpetas que se muestran a la vez. Todos los buckets que ha creado se muestran a la vez. De forma predeterminada, puede crear hasta 100 buckets en cada una de sus cuentas de AWS.

5. Cierre la página Configuración para confirmar los cambios.

También puede actualizar la configuración de un archivo con formato JSON si selecciona el ícono Abrir configuración (JSON) en la parte superior derecha de la página Configuración.

Cargar y descargar objetos de Amazon S3

Puede cargar archivos almacenados localmente en sus buckets de Amazon S3 o descargar objetos remotos de Amazon S3 en su sistema local desde el AWS Toolkit for Visual Studio Code.

Carga de un archivo mediante el kit de herramientas

1. Desde el Explorador del kit de herramientas, expanda el servicio Amazon S3 para ver una lista de sus recursos de S3.
2. Seleccione el ícono de carga de archivos que se encuentre junto a un bucket o una carpeta para abrir el cuadro de diálogo de Cargar archivo. O bien abra el menú contextual (clic con el botón derecho) del archivo y seleccione Cargar archivo.

 Note

Para cargar un archivo en la carpeta o recurso principal del objeto, abra el menú contextual (clic con el botón derecho) de cualquier objeto de S3 y elija Cargar en la cuenta principal.

3. Utilice el administrador de archivos del sistema para elegir un archivo y, a continuación, seleccione Cargar archivo para cerrar el cuadro de diálogo y cargar el archivo.

Carga de un archivo con la paleta de comandos

Puede utilizar la interfaz del kit de herramientas o la paleta de comandos para cargar un archivo en un bucket.

1. Para seleccionar un archivo para cargarlo, elija la pestaña del archivo en VS Code.
2. Pulse Ctrl+Mayús+P para mostrar la paleta de comandos.
3. En la paleta de comandos, escriba la frase `upload file` para ver una lista de los comandos recomendados.
4. Seleccione el comando AWS: Cargar archivo para abrir el cuadro de diálogo AWS: Cargar archivo.
5. Cuando se le solicite, seleccione el archivo que desea cargar y, a continuación, elija el bucket en el que desea cargar ese archivo.

6. Confirme la carga para cerrar el cuadro de diálogo y comenzar el proceso de carga. Una vez finalizada la carga, el objeto se muestra en el menú del kit de herramientas con metadatos que incluyen el tamaño del objeto, la fecha de la última modificación y la ruta.

Descarga de un objeto de Amazon S3

1. Desde el Explorador del kit de herramientas, expanda el servicio S3.
2. Desde un bucket o una carpeta, abra el menú contextual (clic con el botón derecho) del objeto que quiera descargar. A continuación, seleccione Descargar como para abrir el cuadro de diálogo Descargar como. O bien, seleccione el icono Descargar como situado junto al objeto.
3. En el administrador de archivos del sistema, seleccione una carpeta de destino, introduzca un nombre de archivo y, a continuación, seleccione Descargar para cerrar el cuadro de diálogo e iniciar la descarga.

Edición de objetos remotos

Puede usar el AWS Toolkit for Visual Studio Code para editar los objetos de Amazon S3 que estén almacenados en sus recursos remotos de Amazon S3.

1. Desde el Explorador del kit de herramientas, expanda el servicio S3.
2. Expanda el recurso S3 que contiene el archivo que desea editar.
3. Para editar el archivo, elija el icono del lápiz (Editar archivo).
4. Para editar un archivo que está abierto en modo de solo lectura, muéstrello en el editor de VS Code y, a continuación, haga clic en el icono del lápiz ubicado en la esquina superior derecha de la interfaz de usuario.

Note

- Si reinicia o sale de VS Code, el IDE se desconecta de sus recursos de S3. Si se está editando algún archivo remoto de S3 al desconectarse, la edición se detiene. Debe reiniciar VS Code y volver a abrir la pestaña de edición para reanudar la edición.
- El botón Editar archivo está en la esquina superior derecha de la interfaz de usuario. Solo está visible cuando está viendo activamente un archivo de solo lectura en el editor de VS Code.

- Los archivos que no son de texto no se pueden abrir en modo de solo lectura. Siempre se abren en modo de edición.
- No puede volver al modo de solo lectura desde el modo de solo edición, únicamente al contrario.

Copia de la ruta de un objeto de Amazon S3

En el procedimiento siguiente se describe cómo copiar la ruta de un objeto de Amazon S3 desde el AWS Toolkit for Visual Studio Code.

1. Desde el Explorador del kit de herramientas, expanda el servicio S3.
2. Expanda el bucket de recursos que contenga el objeto cuya ruta desea copiar.
3. Abra el menú contextual (clic con el botón derecho) del objeto cuya ruta quiera copiar y, a continuación, seleccione **Copiar ruta** para copiar la ruta del objeto en su portapapeles local.

Generación de una URL prefirmada para un objeto de Amazon S3

Puede compartir objetos privados de Amazon S3 con otros utilizando la característica de URL prefirmada para otorgar permisos de descarga por tiempo limitado. Para obtener más información, consulte [Compartir un objeto con una URL prefirmada](#).

1. Desde el Explorador del kit de herramientas, expanda el servicio S3.
2. Desde un bucket o una carpeta, abra el menú contextual (clic con el botón derecho) del objeto que quiera compartir. A continuación, seleccione **Generar URL prefirmada** para abrir la paleta de comandos.
3. En la paleta de comandos, introduzca el número de minutos durante los cuales se puede utilizar la URL para acceder al objeto. A continuación, pulse la tecla **Intro** para confirmar y cerrar el cuadro de diálogo.
4. Una vez generada la URL prefirmada, la barra de estado de VS Code muestra la URL prefirmada del objeto que se ha copiado en el portapapeles local.

Eliminación de un objeto de Amazon S3

Si un objeto si se encuentra en un bucket sin control de versiones, puede eliminarlo permanentemente. No obstante, en el caso de los buckets que tienen habilitado el control de

versiones, una solicitud de eliminación no elimina permanentemente ese objeto. En su lugar, Amazon S3 inserta un marcador de eliminación en el bucket. Para obtener más información, consulte [Eliminar versiones de objetos](#).

1. Desde el Explorador del kit de herramientas, expanda el servicio S3 para ver una lista de sus recursos de S3.
2. Abra el menú contextual (clic con el botón derecho) del objeto que quiera eliminar y, a continuación, seleccione Eliminar para abrir el diálogo de confirmación.
3. Seleccione Eliminar... para confirmar que desea eliminar el objeto S3. A continuación, cierre el cuadro de diálogo.

Estudio unificado de Amazon SageMaker para VS Code

Como parte de la próxima generación de Amazon SageMaker, Estudio unificado de Amazon SageMaker es una experiencia de desarrollo unificada que reúne servicios de datos, análisis, inteligencia artificial (IA) y machine learning (ML) de AWS. Proporciona un lugar para crear, implementar, ejecutar y supervisar los flujos de trabajo desde una única interfaz. Para obtener más información sobre cómo configurar la integración de Estudio unificado de Amazon SageMaker con el entorno de desarrollo integrado (IDE) de VS Code, consulte [Configuración de la integración de Estudio unificado de Amazon SageMaker en VS Code](#) en la Guía del usuario de Estudio unificado de Amazon SageMaker.

Uso de aplicaciones sin servidor

El AWS Toolkit for Visual Studio Code ofrece compatibilidad con [Aplicación sin servidor de AWS](#). En los siguientes temas se describe cómo empezar a crear aplicaciones de AWS Serverless Application Model (AWS SAM) y a trabajar con ellas desde el AWS Toolkit for Visual Studio Code.

Temas

- [Introducción a las aplicaciones sin servidor](#)
- [Trabajando con AWS Serverless Land](#)
- [Ejecución y depuración de funciones de Lambda directamente desde el código](#)
- [Ejecución y depuración de recursos locales de Amazon API Gateway](#)
- [Opciones de configuración para depurar aplicaciones sin servidor](#)
- [Solución de problemas de las aplicaciones sin servidor](#)

Introducción a las aplicaciones sin servidor

En las siguientes secciones se describe cómo empezar a crear una Aplicación sin servidor de AWS partir de AWS Toolkit for Visual Studio Code, usando AWS Serverless Application Model (AWS SAM) y CloudFormation stacks.

Requisitos previos

Antes de poder crear o trabajar con una Aplicación sin servidor de AWS, debe cumplir los siguientes requisitos previos.

 Note

Es posible que las siguientes operaciones requieran que cierre o reinicie VS Code para que se completen los cambios.

- Instale la interfaz de línea de AWS SAM comandos (CLI). Para obtener información e instrucciones adicionales sobre cómo instalar la AWS SAM CLI, consulte el tema [Instalación de la AWS SAM CLI](#) en esta Guía AWS Serverless Application Model del usuario.
- En el archivo de AWS configuración, identifique AWS la región predeterminada. Para obtener más información sobre el archivo de configuración, consulte el tema [Opciones de los archivos de configuración y credenciales](#) en la Guía del usuario de AWS Command Line Interface .
- Instale el SDK de su idioma y configure la cadena de herramientas. Para obtener información adicional sobre cómo configurar su cadena de herramientas, AWS Toolkit for Visual Studio Code consulte el tema sobre cómo [configurar su cadena de herramientas](#) en esta Guía del usuario.
- Instale la [extensión de compatibilidad con el lenguaje YAML](#) desde el Marketplace de VS Code. Esto es necesario para poder acceder a la CodeLens función de los archivos de AWS SAM plantilla. Para obtener información adicional al respecto CodeLens, consulte la [CodeLens](#) sección de la documentación de VS Code

Permisos de IAM para aplicaciones sin servidor

En el kit de herramientas para VS Code, debe tener un perfil de credenciales que contenga los permisos de AWS Identity and Access Management (IAM) necesarios para implementar y ejecutar aplicaciones sin servidor. Debe tener el read/write acceso adecuado a los siguientes servicios: IAM,

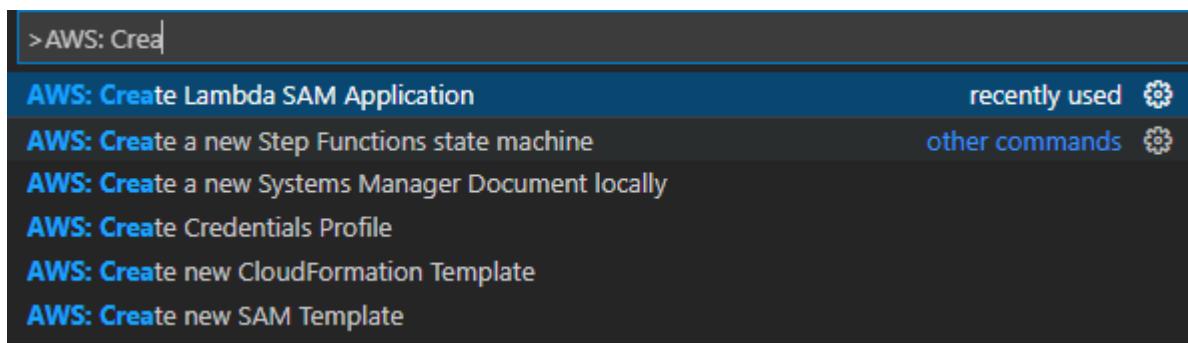
Lambda CloudFormation, Amazon API Gateway, Amazon Simple Storage Service (Amazon S3) (Simple Storage Service (Amazon S3) y Amazon Elastic Container Registry (Amazon ECR).

Para obtener más información sobre cómo configurar la autenticación necesaria para implementar y ejecutar aplicaciones sin servidor, consulte la sección [Administración del acceso a los recursos y los permisos](#) en la Guía para desarrolladores de AWS Serverless Application Model . Para obtener información acerca de cómo configurar sus credenciales, consulte la sección [AWS Credenciales de IAM](#) en esta Guía del usuario.

Crear una nueva aplicación sin servidor (local)

Este procedimiento muestra cómo crear una aplicación sin servidor con el Toolkit for VS Code mediante AWS SAM. El resultado de este procedimiento es un directorio local en su host de desarrollo que contiene un ejemplo de aplicación sin servidor, que puede crear, probar, modificar e implementar localmente en la nube. AWS

1. Para abrir la paleta de comandos, seleccione Ver, Paleta de comandos y, a continuación, introduzca AWS.
2. Elija Kit de herramientas de AWS : Crear aplicación de SAM de Lambda.



Note

Si la AWS SAM CLI no está instalada, aparece un error en la esquina inferior derecha del editor de VS Code. Si esto ocurre, verifique que cumple todas las [hipótesis y requisitos previos](#).

3. Elija el tiempo de ejecución de su AWS SAM aplicación.

 Note

Si selecciona uno de los tiempos de ejecución con “(Imagen)”, su aplicación es el tipo de paquete Image. Si selecciona uno de los tiempos de ejecución sin “(Imagen)”, su aplicación es el tipo Zip. Para obtener más información acerca de la diferencia entre los tipos de paquetes Image y Zip, consulte [Paquetes de implementación de Lambda](#) en la Guía para desarrolladores de AWS Lambda .

4. Según el tiempo de ejecución que seleccione, es posible que se le pida que seleccione un administrador de dependencias y una arquitectura de tiempo de ejecución para su aplicación de SAM.

Dependency Manager

Elija entre Gradle o Maven.

 Note

Esta posibilidad de selección de herramientas de automatización de compilaciones solo está disponible para los tiempos de ejecución de Java.

Architecture

Seleccione entre x86_64 o arm64.

La opción de ejecutar la aplicación sin servidor en un entorno emulado ARM64 basado en lugar del entorno predeterminado basado en x86_64 está disponible para los siguientes tiempos de ejecución:

- nodejs12.x (ZIP e imagen)
- nodejs14.x (ZIP e imagen)
- python3.8 (ZIP e imagen)
- python3.9 (ZIP e imagen)
- python3.10 (ZIP e imagen)
- python3.11 (ZIP e imagen)
- python3.12 (ZIP e imagen)

- java8.al2 con Gradle (ZIP e imagen)
- java8.al2 con Maven (solo ZIP)
- java11 con Gradle (ZIP e imagen)
- java11 con Maven (solo ZIP)

 **Important**

Debe instalar la AWS CLI versión 1.33.0 o posterior para permitir que las aplicaciones se ejecuten en entornos basados. ARM64 Para obtener más información, consulte [Requisitos previos](#).

5. Elija una ubicación para el nuevo proyecto. Puede utilizar una carpeta de espacio de trabajo existente si está abierta, seleccionar una existente en Select a different folder (Seleccionar una carpeta diferente) o crear una nueva y seleccionarla. Para este ejemplo, elija There are no workspace folders open (No hay carpetas de espacio de trabajo abiertas) para crear una carpeta denominada MY-SAM-APP.
6. Escriba un nombre para el proyecto. Para este ejemplo, use my-sam-app-nodejs. Después de pulsar la tecla Intro, el kit de herramientas para VS Code tarda unos momentos en crear el proyecto.

Cuando se crea el proyecto, la aplicación se añade al espacio de trabajo actual. Debería aparecer en la ventana Explorer.

Abrir una aplicación sin servidor (local)

Para abrir una aplicación sin servidor en el host de desarrollo local, abra la carpeta que contenga el archivo de plantilla de la aplicación.

1. En el menú Archivo, elija Abrir.
2. En el cuadro de diálogo Abrir carpeta, navegue hasta la carpeta de la aplicación sin servidor que desee abrir.
3. Pulse el botón Seleccionar carpeta.

Al abrir la carpeta de una aplicación, se añade a la ventana del explorador.

Ejecutar y depurar una aplicación sin servidor desde una plantilla (local)

Puede utilizar el kit de herramientas para VS Code para configurar cómo depurar las aplicaciones sin servidor y ejecutarlas localmente en su entorno de desarrollo.

Se empieza a configurar el comportamiento de depuración mediante la función VS Code para identificar una [CodeLens](#) función Lambda apta. CodeLens permite interacciones basadas en el contenido con el código fuente. Para obtener información sobre cómo garantizar el acceso a la CodeLens función, consulte la [Requisitos previos](#) sección que aparece anteriormente en este tema.

Note

En este ejemplo, depura una aplicación que utiliza JavaScript. No obstante, puede usar las características de depuración del kit de herramientas para VS Code con los siguientes lenguajes y tiempos de ejecución:

- C#: .NET Core 2.1, 3.1; .NET 5.0
- JavaScript/TypeScript — Node.js 12. x, 14. x
- Python: 3.6, 3.7, 3.8, 3.9, 3.10, 3.11, 3.12
- Java: 8, 8.al2, 11
- Go: 1.x

La elección del idioma también afecta a la forma en que CodeLens detecta los controladores Lambda aptos. Para obtener más información, consulte [Ejecución y depuración de funciones de Lambda directamente desde el código](#).

En este procedimiento, utilizará la aplicación de ejemplo creada en la sección [Crear una nueva aplicación sin servidor \(local\)](#) anterior en este tema.

1. Para ver los archivos de su aplicación en el explorador de archivos de VS Code, seleccione Ver, Explorador.
2. Desde la carpeta de la aplicación (por ejemplo, my-sample-app), abra el `template.yaml` archivo.

Note

Si utilizas una plantilla con un nombre diferente al `detemplate.yaml`, el CodeLens indicador no estará disponible automáticamente en el archivo YAML. Esto significa que debe agregar manualmente una configuración de depuración.

3. En el editor de `template.yaml`, vaya a la sección Resources de la plantilla que define los recursos sin servidor. En este caso, este es el recurso `HelloWorldFunction` de tipo `AWS::Serverless::Function`.

En el CodeLens indicador de este recurso, selecciona Añadir configuración de depuración. Utilice el CodeLens indicador del `template.yaml` archivo para añadir una configuración de depuración.

4. En la paleta de comandos, seleccione el tiempo de ejecución durante el cual se ejecutará la aplicación de AWS SAM .
5. En el editor del archivo `launch.json`, edite o confirme los valores de las siguientes propiedades de configuración:
 - "name": ingrese un nombre fácil de leer para que aparezca en el campo desplegable Configuration (Configuración) de la vista Run (Ejecución).
 - "target"— Asegúrese de que el valor sea "template" tal que la AWS SAM plantilla sea el punto de entrada a la sesión de depuración.
 - "templatePath": ingrese una ruta relativa o absoluta para el archivo `template.yaml`.
 - "logicalId"— Asegúrese de que el nombre coincida con el especificado en la sección Recursos de la AWS SAM plantilla. En este caso, es `HelloWorldFunction` de tipo `AWS::Serverless::Function`.

Configuración del archivo `launch.json` para la depuración basada en plantillas.

Para obtener más información sobre estas y otras entradas del archivo `launch.json`, consulte [Opciones de configuración para depurar aplicaciones sin servidor](#).

6. Si está satisfecho con la configuración de depuración, guarde el archivo `launch.json`. A continuación, pulse el botón de "reproducción" verde junto a EJECUTAR para iniciar la depuración.

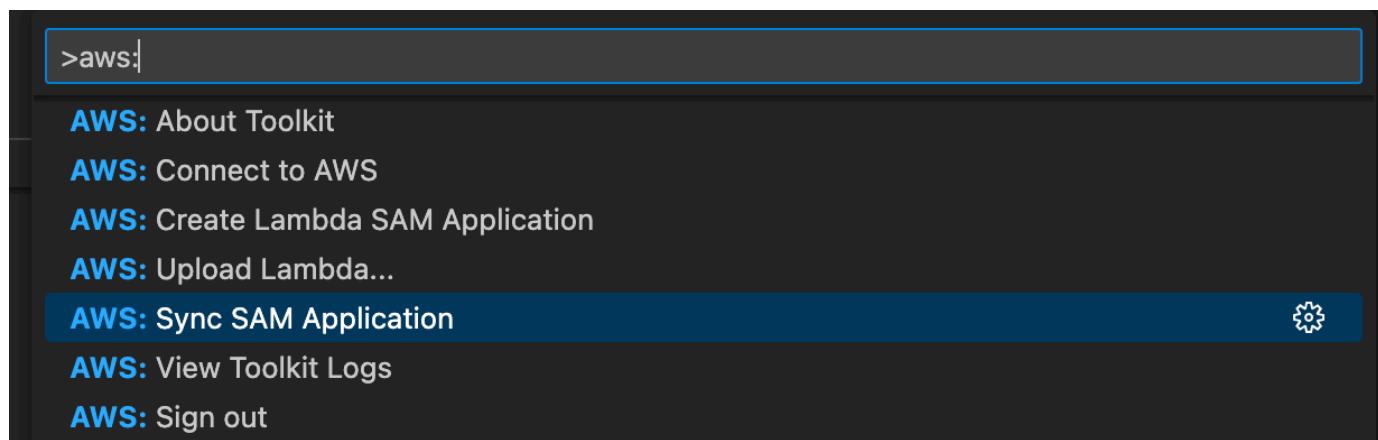
Cuando se inician las sesiones de depuración, el panel DEBUG CONSOLE (CONSOLA DE DEPURACIÓN) muestra la salida de la depuración y los valores devueltos por la función de Lambda. (Al depurar aplicaciones de AWS SAM , el kit de herramientas de AWS se selecciona como el canal de Salida en el panel Salida).

Sincronizar aplicaciones AWS SAM

AWS Toolkit for Visual Studio Code Ejecuta el comando AWS SAM CLI `sam sync` para implementar sus aplicaciones sin servidor en la Nube de AWS. Para obtener información adicional sobre la AWS SAM sincronización, consulte el tema de [referencia de comandos de la AWS SAM CLI](#) en la Guía para AWS Serverless Application Model desarrolladores

El siguiente procedimiento describe cómo implementar sus aplicaciones sin servidor `sam sync` desde el Nube de AWS Toolkit for VS Code.

1. En el menú principal de VS Code, abra la paleta de comandos expandiendo Ver y seleccionando Paleta de comandos.
2. En la paleta de comandos, busque AWS y seleccione Sincronizar aplicación de SAM para empezar a configurar la sincronización.



3. Elija la AWS región con la que desea sincronizar su aplicación sin servidor.
4. Elija el archivo `template.yaml` que desea utilizar para la implementación.
5. Seleccione un bucket de Amazon S3 existente o introduzca un nombre de bucket de Amazon S3 nuevo para implementar la aplicación.

Important

El bucket de Amazon S3 debe cumplir los siguientes requisitos:

- El bucket debe estar en la región en la que esté realizando la sincronización.
- El nombre del bucket de Amazon S3 debe ser único globalmente para todos los nombres de buckets existentes en Amazon S3.

6. Si su aplicación sin servidor incluye una función con el tipo de paquete Image, ingrese el nombre de un repositorio de Amazon ECR que se pueda utilizar en esta implementación. El repositorio debe estar en la región en la que está realizando la implementación.
7. Seleccione una pila de implementación de la lista de sus implementaciones anteriores o cree una nueva pila de implementación introduciendo un nombre de pila nuevo. A continuación, inicie el proceso de sincronización.

 Note

Las pilas utilizadas en implementaciones anteriores se recuperan por espacio de trabajo y región.

8. Durante el proceso de sincronización, el estado de la implementación se registra en la pestaña Terminal de VS Code. Compruebe que la sincronización se ha realizado correctamente desde la pestaña Terminal. Si se produce un error, recibirá una notificación.

 Failed to deploy SAM application.

 Note

Para obtener más información sobre la sincronización, puede acceder a los AWS Toolkit for Visual Studio Code registros desde la paleta de comandos.

Para acceder a los AWS Toolkit for Visual Studio Code registros desde la paleta de comandos, expanda Ver, elija Paleta de comandos y, a continuación **AWS: View AWS Toolkits Logs**, búsquela y selecciónela cuando aparezca en la lista.

Cuando finalice la implementación, verá la aplicación en el Explorador de AWS . Para obtener más información acerca de cómo invocar la función de Lambda creada como parte de la aplicación, consulte el tema [Trabajar con AWS Lambda funciones](#) de esta Guía del usuario.

Eliminar una aplicación sin servidor del Nube de AWS

Eliminar una aplicación sin servidor implica eliminar la CloudFormation pila que implementaste anteriormente en la AWS nube. Tenga en cuenta que este procedimiento no elimina el directorio de la aplicación del anfitrión local.

1. Abra la [Explorador de AWS](#).
2. En la ventana Explorador del kit de herramientas de AWS , expanda la región que contiene la aplicación implementada que desea eliminar y, a continuación, expanda CloudFormation.
3. Abre el menú contextual (haz clic con el botón derecho) para ver el nombre de la CloudFormation pila que corresponde a la aplicación sin servidor que deseas eliminar y, a continuación, selecciona Eliminar CloudFormation pila.
4. Para confirmar que desea eliminar la pila seleccionada, seleccione Sí.

Si la eliminación de la pila se realiza correctamente, el kit de herramientas para VS Code elimina el nombre de la pila de la lista de CloudFormation en el Explorador de AWS .

Trabajando con AWS Serverless Land

AWS Serverless Land in the AWS Toolkit for Visual Studio Code es un conjunto de funciones que le ayudan a crear arquitecturas basadas en eventos. En las siguientes secciones de temas, se describe cómo trabajar con Serverless Land en el kit de herramientas. Para obtener información detallada sobre Serverless Land, consulte la aplicación web de [Serverless Land](#).

Acceso a Serverless Land

Hay tres puntos de entrada principales para acceder a Serverless Land en el kit de herramientas: AWS

- La paleta de comandos de VS Code
- El explorador del kit de herramientas AWS
- El explorador de AWS Toolkit Application Builder

Apertura de Serverless Land desde la paleta de comandos de VS Code

Para abrir Serverless Land desde la paleta de comandos de VS Code, complete los siguientes pasos:

1. En VS Code, pulse **option+shift+p** (Mac) o **control+shift+p** (Windows) para abrir la paleta de comandos.
2. En la paleta de comandos de VS Code, escriba **AWS Create application with Serverless template** en la barra de búsqueda.
3. Elija AWS: Crear aplicación con plantilla de Serverless cuando aparezca en la lista.
4. El asistente de Serverless Land se abre en la pantalla Seleccionar un patrón para su aplicación (1/5) de VS Code una vez completado el proceso.

Abrir Serverless Land desde el AWS Toolkit Explorer.

Para abrir Serverless Land desde el explorador del AWS kit de herramientas, complete los siguientes pasos.

1. Desde el explorador del AWS kit de herramientas, expanda la región en la que desee abrir Serverless Land.
2. Abra el menú contextual (con el botón derecho) del nodo de Lambda.
3. Seleccione Crear aplicación con plantilla de Serverless en el menú contextual.
4. El asistente de Serverless Land se abre en la pantalla Seleccionar un patrón para su aplicación (1/5) de VS Code una vez completado el proceso.

Apertura de Serverless Land desde el explorador Desarrollador de aplicaciones

Para abrir Serverless Land desde el explorador de AWS Toolkit Application Builder, complete los siguientes pasos.

1. Desde el explorador del AWS kit de herramientas, navegue hasta el explorador de Application Builder.
2. Haga clic con el botón derecho en el explorador Desarrollador de aplicaciones y elija Crear aplicación con plantilla de Serverless en el menú contextual.
3. El asistente de Serverless Land se abre en la pantalla Seleccionar un patrón para su aplicación (1/5) de VS Code una vez completado el proceso.

Creación de una aplicación con una plantilla de Serverless

Para crear una aplicación con plantilla de Serverless, complete los siguientes pasos:

1. En la pantalla Seleccionar un patrón para su aplicación (1/5) del asistente de Serverless Land, elija un patrón como base de la aplicación.

 Note

Para obtener una vista previa y detalles adicionales de un patrón en particular, seleccione el ícono Abrir en Serverless Land situado junto al patrón que desea ver. El patrón de Serverless Land se abre en el navegador web predeterminado.

2. En la pantalla Seleccionar el tiempo de ejecución (2/5), elija un tiempo de ejecución para el proyecto.
3. En la pantalla Seleccionar IaC (3/5), elija una opción de infraestructura como código (IaC) para el proyecto.
4. En la pantalla Seleccionar una ubicación del proyecto (4/5), elija una ubicación para almacenar el proyecto.
5. En la pantalla Ingresar el nombre del proyecto (5/5), escriba un nombre para la nueva aplicación.
6. La nueva aplicación se muestra en el explorador de VS Code y el proyecto `readme.md` se abre en el editor de VS Code al completar el procedimiento.

 Note

Una vez creada la nueva aplicación, encontrará acciones adicionales específicas de su tipo de aplicación en el archivo `readme.md`. Además, sus aplicaciones AWS Serverless Application Model (AWS SAM) se pueden abrir con AWS Application Builder para realizar pruebas locales, depurarlas y mucho más.

Para obtener más información sobre cómo trabajar con Application Builder en el AWS kit de herramientas, consulte el tema [Cómo trabajar con el explorador de AWS Application Builder](#) en esta guía del usuario.

Ejecución y depuración de funciones de Lambda directamente desde el código

Al probar la AWS SAM aplicación, puede optar por ejecutar y depurar solo la función Lambda y excluir otros recursos que defina AWS SAM la plantilla. Este enfoque implica el uso de la

CodeLens función para identificar los controladores de funciones Lambda en el código fuente que se pueden invocar directamente.

Los controladores Lambda que detecta CodeLens dependen del lenguaje y del tiempo de ejecución que utilice para la aplicación.

Lenguaje/tiempo de ejecución	Criterios para identificar las funciones Lambda mediante indicadores CodeLens
C# (dotnetcore2.1, 3.1; .NET 5.0)	<p>La función tiene las siguientes características:</p> <ul style="list-style-type: none">• Es una función pública de clase pública.• Tiene uno o dos parámetros. Con dos parámetros, el segundo parámetro debe implementar la interfaz <code>ILambdaContext</code>.• Tiene un archivo <code>*.csproj</code> en su carpeta principal en la carpeta del área de trabajo de VS Code. <p>La extensión ms-dotnettools.csharp (o cualquier extensión que proporcione símbolos de lenguaje para C#) está instalada y habilitada.</p>
JavaScript/TypeScript (Node.js 12.x, 14.x)	<p>La función tiene las siguientes características:</p> <ul style="list-style-type: none">• Es una función exportada con hasta tres parámetros.• Tiene un archivo <code>package.json</code> en su carpeta principal en la carpeta del área de trabajo de VS Code.
Python (3.7, 3.8, 3.9, 3.10, 3.11, 3.12)	<p>La función tiene las siguientes características:</p> <ul style="list-style-type: none">• Es una función de alto nivel.• Tiene un archivo <code>requirements.txt</code> en su carpeta principal en la carpeta del área de trabajo de VS Code.

Lenguaje/tiempo de ejecución	Criterios para identificar las funciones Lambda mediante indicadores CodeLens
	La extensión ms-python.python (o cualquier extensión que proporcione símbolos de lenguaje para Python) está instalada y habilitada.

Lenguaje/tiempo de ejecución	Criterios para identificar las funciones Lambda mediante indicadores CodeLens
Java (8, 8.al2, 11)	<p>La función tiene las siguientes características:</p> <ul style="list-style-type: none"> • Es una función pública de clase pública no abstracta. • Tiene uno, dos o tres parámetros: <ul style="list-style-type: none"> • Un parámetro: el parámetro puede ser cualquier cosa. • Dos parámetros: los parámetros deben ser <code>java.io.InputStream</code> y <code>java.io.OutputStream</code>, O el último parámetro debe ser <code>com.amazonaws.services.lambda.runtime.Context</code>. • Dos parámetros: los parámetros deben ser <code>java.io.InputStream</code> y <code>java.io.OutputStream</code> Y el último parámetro debe ser <code>com.amazonaws.services.lambda.runtime.Context</code>. • Tiene un archivo <code>build.gradle</code> (Gradle) o <code>pom.xml</code> (Maven) en su carpeta principal dentro de la carpeta del área de trabajo de VS Code. <p>La extensión redhat.java (o cualquier extensión que proporcione símbolos de lenguaje para Java) está instalada y habilitada. Esta extensión requiere Java 11, independientemente del tiempo de ejecución de Java que utilice.</p> <p>El vscjava. vscode-javascript-debug La extensión (o cualquier extensión que proporcione un depurador de Java) está instalada y habilitada.</p>

Lenguaje/tiempo de ejecución	Criterios para identificar las funciones Lambda mediante indicadores CodeLens
Go (1.x)	<p>La función tiene las siguientes características:</p> <ul style="list-style-type: none">• Es una función de alto nivel.• Se necesitan entre 0 y 2 argumentos. Si hay dos argumentos, el primero debe implementar <code>context.Context</code>.• Devuelve entre 0 y 2 argumentos. Si hay más de 0 argumentos, el último debe implementar <code>error</code>.• Tiene un archivo <code>go.mod</code> dentro de la carpeta del área de trabajo de VS Code. <p>La extensión <code>golang.go</code> está instalada, configurada y habilitada.</p>

Para ejecutar y depurar una aplicación sin servidor directamente desde el código de la aplicación

1. Para ver los archivos de su aplicación en el explorador de archivos de VS Code, seleccione Ver, Explorador.
2. Desde la carpeta de aplicaciones (por ejemplo, `my-sample-app`), expanda la carpeta de funciones (en este caso, `hello-world`) y abra el archivo `app.js`
3. En el CodeLens indicador que identifica un controlador de funciones Lambda apto, elija `Add Debug Configuration`
Acceda a la opción `Añadir configuración de depuración` en el CodeLens indicador de un controlador de funciones Lambda.
4. En la paleta de comandos, seleccione el tiempo de ejecución durante el cual se ejecutará la aplicación de AWS SAM .
5. En el editor del archivo `launch.json`, edite o confirme los valores de las siguientes propiedades de configuración:

- "name": ingrese un nombre fácil de leer para que aparezca en el campo desplegable Configuration (Configuración) de la vista Run (Ejecución).
- "target": asegúrese de que el valor es "code" para que un controlador de funciones de Lambda se invoque directamente.
- "lambdaHandler": ingrese el nombre del método en el código al que Lambda llama para ejecutar su función. Por ejemplo, para las aplicaciones en JavaScript, el valor predeterminado es. app.lambdaHandler
- "projectRoot": ingrese la ruta al archivo de aplicación que contiene la función de Lambda.
- "runtime": ingrese o confirme un tiempo de ejecución válido para el entorno de ejecución de Lambda; por ejemplo, "nodejs.12x".
- "payload". Elija una de las siguientes opciones para definir la carga del evento que desea proporcionar a la función de Lambda como entrada:
 - "json": pares clave-valor con formato JSON que definen la carga del evento.
 - "path": ruta al archivo que se utiliza como carga del evento.

En el siguiente ejemplo, la opción "json" define la carga útil.

Configuración del archivo `launch.json` para invocar directamente las funciones de Lambda.

Para obtener más información sobre estas y otras entradas del archivo `launch.json`, consulte [Opciones de configuración para depurar aplicaciones sin servidor](#).

6.

Si está satisfecho con la configuración de depuración, elija la flecha de reproducción verde junto a EJECUTAR para iniciar la depuración.

Cuando se inician las sesiones de depuración, el panel CONSOLA DE DEPURACIÓN muestra la salida de la depuración y los valores que devuelve la función de Lambda. (Al depurar AWS SAM aplicaciones, se selecciona AWS Toolkit como canal de salida en el panel de salida).

Ejecución y depuración de recursos locales de Amazon API Gateway

Puede ejecutar o depurar recursos locales de AWS SAM API Gateway, especificados en `template.yaml`, mediante la ejecución de una configuración de lanzamiento de VS Code de `type=aws-sam` con `invokeTarget.target=api`.

Note

API Gateway admite dos tipos de API: REST y HTTP. Sin embargo, la característica API Gateway con el AWS Toolkit for Visual Studio Code solo admite las API de REST. A veces las API HTTP se llaman “API Gateway V2”.

Para ejecutar y depurar recursos de API Gateway locales

1. Elija uno de los siguientes enfoques para crear una configuración de lanzamiento para un recurso de AWS SAM API Gateway:

- Opción 1: visite el código fuente del controlador (archivo .js, .cs o .py) en su proyecto de AWS SAM, coloque el cursor sobre el controlador de Lambda y elija el CodeLens Agregar configuración de depuración. A continuación, en el menú, elija el elemento marcado como Evento de API.
- Opción 2: edite `launch.json` y cree una nueva configuración de lanzamiento mediante la sintaxis siguiente.

```
{  
  "type": "aws-sam",  
  "request": "direct-invoker",  
  "name": "myConfig",  
  "invokeTarget": {  
    "target": "api",  
    "templatePath": "n12/template.yaml",  
    "logicalId": "HelloWorldFunction"  
  },  
  "api": {  
    "path": "/hello",  
    "httpMethod": "post",  
    "payload": {  
      "json": {}  
    }  
  },  
  "sam": {},  
  "aws": {}  
}
```

2. En el panel Ejecutar de VS Code, elija la configuración de lanzamiento (denominada `myConfig` en el ejemplo anterior).
3. (Opcional) Agregue puntos de interrupción al código del proyecto de Lambda.
4. Escriba F5 o seleccione Reproducir en el panel Ejecutar.
5. En el panel de resultados, vea los resultados.

Configuración

Cuando utiliza el valor `api` de la propiedad `invokeTarget.target`, Toolkit cambia la validación y el comportamiento de la configuración de lanzamiento para admitir un campo `api`.

```
{  
  "type": "aws-sam",  
  "request": "direct-invoke",  
  "name": "myConfig",  
  "invokeTarget": {  
    "target": "api",  
    "templatePath": "n12/template.yaml",  
    "logicalId": "HelloWorldFunction"  
  },  
  "api": {  
    "path": "/hello",  
    "httpMethod": "post",  
    "payload": {  
      "json": {}  
    },  
    "queryString": "abc=def&qrs=tuv",  
    "headers": {  
      "cookie": "name=value; name2=value2; name3=value3"  
    }  
  },  
  "sam": {},  
  "aws": {}  
}
```

Reemplace los valores del ejemplo de la siguiente manera:

invokeTarget.logicalId

Un recurso de API.

path

La ruta de la API que solicita la configuración de lanzamiento; por ejemplo, "path": "/hello".

Debe ser una ruta de API válida resuelta desde el archivo `template.yaml` especificado por `invokeTarget.templatePath`.

httpMethod

Uno de los verbos siguientes: "delete", "get", "head", "options", "patch", "post", "put".

payload

La carga JSON (cuerpo HTTP) para enviar en la solicitud, con la misma estructura y las mismas reglas que el campo [lambda.payload](#).

`payload.path` apunta a un archivo que contiene la carga JSON.

`payload.json` especifica una carga JSON en línea.

headers

Mapa opcional de pares nombre-valor, que se utiliza para especificar encabezados HTTP que se incluirán en la solicitud, tal y como se muestra en el siguiente ejemplo.

```
"headers": {  
  "accept-encoding": "deflate, gzip;q=1.0, *;q=0.5",  
  "accept-language": "fr-CH, fr;q=0.9, en;q=0.8, de;q=0.7, *;q=0.5",  
  "cookie": "name=value; name2=value2; name3=value3",  
  "user-agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6)  
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/86.0.4240.198 Safari/537.36",  
}
```

queryString

Cadena opcional que establece la propiedad `queryString` de la solicitud; por ejemplo, `"queryString": "abc=def&ghi=jkl"`.

AWS

Cómo se proporciona la información de conexión de AWS. Para obtener más información, consulte la tabla de propiedades (“aws”) de conexión de AWS en la sección [Opciones de configuración para depurar aplicaciones sin servidor](#).

sam

Procedimiento que sigue AWS SAM CLI para compilar la aplicación. Para obtener más información, consulte la tabla de propiedades (“sam”) de conexión de AWS SAM CLI en la sección [Opciones de configuración para depurar aplicaciones sin servidor](#).

Opciones de configuración para depurar aplicaciones sin servidor

Al abrir el archivo `launch.json` para editar las configuraciones de depuración, puede usar la característica [IntelliSense](#) de VS Code para ver y completar automáticamente las propiedades válidas. Para activar IntelliSense en el editor, presione `Ctrl+barra espaciadora`.



The screenshot shows a code editor with a dark theme. A JSON configuration file is open, specifically the `lambda` section of a `launch.json` file. The code is as follows:

```
"lambda": [    "runtime": "nodejs12.x",    [      "event": {        "json": {}      }    }]
```

IntelliSense is active, with a tooltip or dropdown menu visible over the `event` property, showing suggestions for valid event types. The editor's interface includes standard VS Code elements like tabs, a status bar, and a sidebar.

Con IntelliSense puede encontrar y definir las propiedades para invocar funciones de Lambda directamente o con la plantilla de AWS SAM. También puede definir las propiedades de "lambda" (cómo se ejecuta la función), "sam" (cómo compila la aplicación AWS SAM CLI) y "aws" (cómo se proporciona la información de conexión de AWS).

AWS SAM: invocación directa del controlador de Lambda/invocación de Lambda basada en plantilla

Propiedad	Descripción
<code>type</code>	Especifica qué extensión administra la configuración de lanzamiento. Siempre se establece en <code>aws-sam</code> para usar AWS SAM CLI para realizar la compilación y depuración localmente.

Propiedad	Descripción
<code>name</code>	Especifica un nombre fácil de leer que aparecerá en la lista Debug launch configuration (Depurar configuración de lanzamiento).
<code>request</code>	Especifica el tipo de configuración que debe realizar la extensión designada (aws-sam). Siempre se establece en <code>direct-invoke</code> para lanzar la función de Lambda.

Propiedad	Descripción
invokeTarget	<p>Especifica el punto de entrada para la invocación del recurso.</p> <p>Para invocar directamente la función de Lambda, establezca los valores de los campos <code>invokeTarget</code> siguientes:</p> <ul style="list-style-type: none"> • <code>target</code> – se establece en <code>code</code>. • <code>lambdaHandler</code> : nombre del controlador de funciones de Lambda que se va a invocar. • <code>projectRoot</code> : ruta del archivo de aplicación que contiene el controlador de la función de Lambda. • <code>architecture</code> : arquitectura de procesador del entorno emulado en el que se ejecuta la aplicación de Lambda local para SAM. Para determinados tiempos de ejecución, puede elegir <code>arm64</code> en lugar de la arquitectura <code>x86_64</code> predeterminada. Para obtener más información, consulte Crear una nueva aplicación sin servidor (local). <p>Para invocar los recursos de Lambda con la plantilla de AWS SAM, establezca los valores de los campos <code>invokeTarget</code> siguientes:</p> <ul style="list-style-type: none"> • <code>target</code> – Establece en <code>template</code>. • <code>templatePath</code> : ruta al archivo de plantilla de AWS SAM. • <code>logicalId</code> : nombre del recurso de <code>AWS::Lambda::Function</code> o <code>AWS::Serverless::Function</code> que se va a invocar. Puede encontrar el nombre del recurso en la plantilla de AWS SAM con formato YAML. Tenga en cuenta que el AWS Toolkit reconoce implícitamente las funciones definidas con <code>PackageType: Image</code> en la plantilla de AWS SAM como funciones de Lambda basadas en imágenes. Para obtener más información, consulte Paquetes de implementación de Lambda en la Guía para desarrolladores de AWS Lambda.

Propiedades de Lambda ("lambda")

Propiedad	Descripción
environmentVariables	<p>Pasa parámetros operativos a la función de Lambda. Por ejemplo, si está escribiendo en un bucket de Amazon S3, en lugar de codificar de forma rígida el nombre del bucket, configúrelo como una variable de entorno.</p> <p>Note</p> <p>Al especificar variables de entorno para una aplicación sin servidor, debe añadir configuraciones tanto a la plantilla de AWS SAM (<code>template.yaml</code>) como al archivo <code>launch.json</code>.</p> <p>Ejemplo de formato para una variable de entorno en la plantilla de AWS SAM:</p> <pre>Resources: HelloWorldFunction: Type: AWS::Serverless::Function Properties: CodeUri: hello-world/ Handler: app.lambdaHandlerN10 Runtime: nodejs10.x Environment: Variables: SAMPLE1: Default Sample 1 Value</pre> <p>Ejemplo de formato para una variable de entorno en el archivo <code>launch.json</code>:</p> <pre>"environmentVariables": { "SAMPLE1": "My sample 1 value" }</pre>
payload	Proporciona dos opciones para la carga del evento que proporciona a la función de Lambda como entrada.

Propiedad	Descripción
	<ul style="list-style-type: none">• "json": pares clave-valor con formato JSON que definen la carga del evento.• "path": ruta al archivo que se utiliza como carga del evento.
memoryMB	Especifica los megabytes (MB) de memoria proporcionados para ejecutar una función de Lambda invocada.
runtime	Especifica el tiempo de ejecución que usa la función de Lambda. Para obtener más información, consulte Tiempos de ejecución de AWS Lambda .
timeoutSec	Establece el tiempo permitido, en segundos, antes de que se agote la sesión de depuración.

Propiedad	Descripción
pathMappings	<p>Especifica dónde se encuentra el código local en relación con el lugar en el que se ejecuta en el contenedor.</p> <p>De forma predeterminada, el kit de herramientas para VS Code establece <code>localRoot</code> en la raíz del código de la función de Lambda en el espacio de trabajo local y <code>remoteRoot</code> en <code>/var/task</code>, que es el directorio de trabajo predeterminado para ejecutar el código en Lambda. Si el directorio de trabajo se cambia en el Dockerfile o con el parámetro <code>WorkingDirectory</code> en el archivo de plantilla de CloudFormation, se debe especificar al menos una entrada de <code>pathMapping</code> para que el depurador pueda asignar correctamente los puntos de interrupción establecidos localmente al código que se ejecuta en el contenedor Lambda.</p> <p>Ejemplo de formato para <code>pathMappings</code> en el archivo <code>launch.json</code> :</p> <pre> "pathMappings": [{ "localRoot": "\${workspaceFolder}/sam-app/ HelloWorldFunction", "remoteRoot": "/var/task" }] </pre> <p>Advertencias:</p> <ul style="list-style-type: none"> • Con las funciones de Lambda basadas en imágenes .NET, la entrada <code>remoteRoot</code> debe ser el directorio de compilación. • Con las funciones de Lambda basadas en Node.js, puede especificar solo una entrada de asignación de rutas.

El kit de herramientas para VS Code utiliza la CLI de AWS SAM para compilar y depurar aplicaciones sin servidor localmente. Puede configurar el comportamiento de los comandos de AWS SAM CLI que utilizan las propiedades de la configuración de "sam" en el archivo `launch.json`.

Propiedades de AWS SAM CLI ("sam")

Propiedad	Descripción	Valor predeterminado
<code>buildArguments</code>	Configura el modo en que el comando <code>sam build</code> compila el código fuente de Lambda. Para ver las opciones de compilación, consulte la compilación de sam en la Guía para desarrolladores de AWS Serverless Application Model.	Cadena vacía
<code>containerBuild</code>	Indica si se debe compilar la función en un contenedor de Docker tipo Lambda.	<code>false</code>
<code>dockerNetwork</code>	Especifica el nombre o el ID de una red de Docker existente a la que deben conectarse los contenedores de Lambda y Docker, junto con la red en modo puente predeterminada. Si no se especifica, los contenedores de Lambda se conectan únicamente a la red Docker en modo puente predeterminada.	Cadena vacía
<code>localArguments</code>	Especifica argumentos adicionales de invocación local.	Cadena vacía

Propiedad	Descripción	Valor predeterminado
skipNewImageCheck	Especifica si el comando debe omitir la extracción de la última imagen de Docker para el tiempo de ejecución de Lambda.	false
template	Personaliza la plantilla de AWS SAM mediante parámetros para ingresar valores de cliente. Para obtener más información, consulte Parámetros en la Guía del usuario de AWS CloudFormation.	"parameters":{}

Propiedades de conexión de AWS ("aws")

Propiedad	Descripción	Valor predeterminado
credentials	Selecciona un perfil específico (por ejemplo, <code>profile:default</code>) de su archivo de credenciales para obtener las credenciales de AWS.	Las credenciales de AWS que su archivo de configuración de AWS compartido o el archivo de credenciales de AWS compartido existente s proporcionan al kit de herramientas de VS Code.
region	Establece la región de AWS del servicio (por ejemplo, <code>us-east-1</code>).	Región de AWS predeterminada asociada al perfil de credenciales activo.

Ejemplo: Configuración de lanzamiento de una plantilla

A continuación, se muestra un ejemplo de archivo de configuración de lanzamiento para un destino de plantilla de AWS SAM:

```
{  
  "configurations": [  
    {  
      "type": "aws-sam",  
      "request": "direct-invoker",  
      "name": "my-example:HelloWorldFunction",  
      "invokeTarget": {  
        "target": "template",  
        "templatePath": "template.yaml",  
        "logicalId": "HelloWorldFunction"  
      },  
      "lambda": {  
        "payload": {},  
        "environmentVariables": {}  
      }  
    }  
  ]  
}
```

Ejemplo: Configuración de lanzamiento de un código

A continuación, se muestra un ejemplo de archivo de configuración de lanzamiento para un destino de función de Lambda:

```
{  
  "configurations": [  
    {  
      "type": "aws-sam",  
      "request": "direct-invoker",  
      "name": "my-example:app.lambda_handler (python3.7)",  
      "invokeTarget": {  
        "target": "code",  
        "projectRoot": "hello_world",  
        "lambdaHandler": "app.lambda_handler"  
      },  
      "lambda": {  
        "runtime": "python3.7",  
        "payload": {},  
        "environmentVariables": {}  
      }  
    }  
  ]  
}
```

}

Solución de problemas de las aplicaciones sin servidor

En este tema se detallan los errores comunes que se pueden encontrar al crear aplicaciones sin servidor con el kit de herramientas para VS Code y cómo resolverlos.

Temas

- [¿Cómo puedo usar samconfig.toml con una configuración de lanzamiento de SAM?](#)
- [Error: “RuntimeError: Container does not exist”](#)
- [Error: “docker.errors.APIError: 500 Server Error... You have reached your pull rate limit”.](#)
- [Error: “500 Server Error: Mounting C:\Users\...”](#)
- [Al usar WSL, las vistas web \(por ejemplo, el formulario “Invoke on AWS”\) no funcionan](#)
- [Depuración de una aplicación de TypeScript, pero los puntos de interrupción no funcionan](#)

¿Cómo puedo usar samconfig.toml con una configuración de lanzamiento de SAM?

Especifique la ubicación de su archivo [SAMconfig.toml](#) de la CLI de SAM configurando el argumento `--config-file` en la propiedad `sam.localArguments` de su configuración de lanzamiento. Por ejemplo, si el archivo `samconfig.toml` se encuentra en el nivel superior de su espacio de trabajo:

```
"sam": {  
  "localArguments": ["--config-file", "${workspaceFolder}/samconfig.toml"],  
}
```

Error: “RuntimeError: Container does not exist”

El comando `sam build` puede mostrar este error si el sistema no tiene suficiente espacio en el disco para el contenedor de Docker. Si el almacenamiento del sistema solo tiene 1 o 2 GB de espacio disponible, es posible que se produzca un error en `sam build` durante el procesamiento, incluso si el almacenamiento del sistema no está completamente lleno antes de que comience la compilación. Para obtener más información, consulte [este problema de GitHub](#).

Error: “`docker.errors.APIError: 500 Server Error... You have reached your pull rate limit`”.

Docker Hub limita las solicitudes que pueden realizar los usuarios anónimos. Si el sistema alcanza el límite, Docker falla y aparece este error en la vista SALIDA de VS Code:

```
docker.errors.APIError: 500 Server Error: Internal Server Error ("toomanyrequests: You
have
reached your pull rate limit. You may increase the limit by authenticating and
upgrading:
https://www.docker.com/increase-rate-limit")
```

Asegúrese de que el servicio Docker del sistema se haya autenticado con sus credenciales de Docker Hub.

Error: “500 Server Error: Mounting C:\Users\...”

Los usuarios de Windows pueden ver este error de montaje de Docker al depurar aplicaciones de AWS SAM:

```
Fetching lambci/lambda:nodejs10.x Docker container image.....
2019-07-12 13:36:58 Mounting C:\Users\<username>\AppData\Local\Temp\ ... as /var/
task:ro,delegated inside runtime container
Traceback (most recent call last):
...
requests.exceptions.HTTPError: 500 Server Error: Internal Server Error ...
```

Pruebe a actualizar las credenciales de sus unidades compartidas (en la configuración de Docker).

Al usar WSL, las vistas web (por ejemplo, el formulario “Invoke on AWS”) no funcionan

Se trata de un problema de VS Code conocido para los usuarios de Cisco VPN. Para obtener más información, consulte [este problema de GitHub](#).

Se sugiere una solución alternativa en [este problema de seguimiento de WSL](#).

Depuración de una aplicación de TypeScript, pero los puntos de interrupción no funcionan

Esto sucede si no hay un mapa fuente para vincular el archivo JavaScript compilado al archivo TypeScript fuente. Para solucionarlo, abra el archivo `tsconfig.json` y asegúrese de que estén configurados la siguiente opción y valor: `"inlineSourceMap": true`.

Uso de documentos de Automatización de Systems Manager

AWS Systems Manager le proporciona visibilidad y control de su infraestructura AWS. Systems Manager proporciona una interfaz de usuario unificada para que pueda ver los datos operativos de varios AWS servicios y automatizar las tareas operativas en todos sus AWS recursos.

Un [documento de Systems Manager](#) define las acciones que Systems Manager lleva a cabo en las instancias administradas. Un documento de Automatización es un tipo de documento de Systems Manager que usa para llevar a cabo tareas de implementación y mantenimiento comunes como, por ejemplo, crear o actualizar una imagen de máquina de Amazon (AMI). En este tema se describe cómo crear, editar, publicar y eliminar documentos de Automation con AWS Toolkit for Visual Studio Code.

Temas

- [Hipótesis y requisitos previos](#)
- [Permisos de IAM para documentos de Systems Manager Automation](#)
- [Creación de un nuevo documento de Automatización de Systems Manager](#)
- [Apertura de un documento de automatización de Systems Manager existente](#)
- [Edición de un documento de Automatización de Systems Manager](#)
- [Publicación de un documento de Automatización de Systems Manager](#)
- [Eliminación de un documento de Automatización de Systems Manager](#)
- [Ejecución de un documento de Automatización de Systems Manager](#)
- [Solución de problemas de documentos de Automatización de Systems Manager en el kit de herramientas para VS Code](#)

Hipótesis y requisitos previos

Antes de comenzar, asegúrese de que:

- Ha instalado Visual Studio Code y la última versión del AWS Toolkit for Visual Studio Code. Para obtener más información, consulte [Instalación del AWS Toolkit for Visual Studio Code](#).
- Está familiarizado con Systems Manager. Para obtener más información, consulte la [Guía del usuario de AWS Systems Manager](#).
- Está familiarizado con los casos de uso de Automatización de Systems Manager. Para obtener más información, consulte [AWS Systems Manager Automation](#) en la Guía del usuario de AWS Systems Manager .

Permisos de IAM para documentos de Systems Manager Automation

En el kit de herramientas para VS Code, debe tener un perfil de credenciales que contenga los permisos de AWS Identity and Access Management (IAM) necesarios para crear, editar, publicar y eliminar documentos de Automatización de Systems Manager. En el siguiente documento de política se definen los permisos de IAM necesarios que se pueden utilizar en una política maestra:

JSON

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "ssm>ListDocuments",  
        "ssm>ListDocumentVersions",  
        "ssm>DescribeDocument",  
        "ssm>GetDocument",  
        "ssm>CreateDocument",  
        "ssm>UpdateDocument",  
        "ssm>UpdateDocumentDefaultVersion",  
        "ssm>DeleteDocument"  
      ],  
      "Resource": "*"  
    }  
  ]  
}
```

Para obtener más información sobre cómo crear una política de IAM, consulte [Creación de políticas de IAM](#) en la Guía del usuario de IAM. Para obtener información sobre cómo configurar su perfil de credenciales, consulte [AWS Credenciales de IAM](#).

Creación de un nuevo documento de Automatización de Systems Manager

Puede crear un nuevo documento de Automatización en formato JSON o YAML utilizando Visual Studio Code. Cuando crea un documento de Automatización nuevo, se presenta en un archivo sin título. Puede asignar un nombre a su archivo y guardarla en VS Code; sin embargo, el nombre del archivo no estará visible AWS.

Para crear un documento de Automatización nuevo

1. Abra VS Code.
2. En el menú Ver, seleccione Paleta de comandos para abrir la paleta de comandos.
3. En la paleta de comandos, escriba Kit de herramientas de AWS : crear un nuevo documento de Systems Manager local.
4. Elija una de las plantillas iniciales para un ejemplo de Hello World.
5. Elija JSON o YAML.

Se crea un nuevo documento de Automatización.

 Note

El nuevo documento de Automatización en VS Code no aparece automáticamente en AWS. Debe publicarlo AWS antes de poder ejecutarlo.

Apertura de un documento de automatización de Systems Manager existente

Utilice el AWS Explorador para buscar los documentos existentes de Systems Manager Automation. Al abrir un documento de Automatización existente, aparece un archivo sin título en VS Code.

Para abrir el documento de Automatización

1. Abra VS Code.

2. En la barra de navegación de la izquierda, seleccione AWS para abrir el Explorador de AWS .
3. En el AWS Explorador, en Systems Manager, elija el icono de descarga en el documento que desee abrir y, a continuación, elija la versión del documento. El archivo se abrirá en el formato de esa versión. De lo contrario, seleccione Descargar como JSON o Descargar como YAML.

 Note

Aunque se guarde localmente un documento de Automatización como un archivo en VS Code, no implica que aparezca en AWS. Debe publicarse AWS antes de ejecutarse.

Edición de un documento de Automatización de Systems Manager

Si es propietario de algún documento de automatización, aparecerá en la categoría Owned by Me de documentos de Systems Manager en el AWS Explorador. Puede ser propietario de los documentos de automatización que ya existen y puede ser propietario de los documentos nuevos o actualizados que haya publicado anteriormente AWS desde VS Code. AWS

Cuando abre un documento de Automatización para editarlo en VS Code, puede hacer más con él que en la Consola de administración de AWS. Por ejemplo:

- Hay una validación de esquemas tanto en formato JSON como YAML.
- Hay fragmentos disponibles en el editor de documentos para que pueda crear cualquiera de los diferentes tipos de paso de automatización.
- La característica de autocompletar se admite en varias opciones de JSON y YAML.

Uso de versiones

Los documentos de Automatización de Systems Manager utilizan versiones para la administración de cambios. Puede elegir la versión predeterminada para un documento de Automatización en VS Code.

Para establecer una versión predeterminada

- En el AWS explorador, navegue hasta el documento en el que desee establecer la versión predeterminada, abra el menú contextual (haga clic con el botón derecho) del documento y seleccione Establecer versión predeterminada.

Note

Si el documento elegido solo tiene una versión, no podrá cambiar el valor predeterminado.

Publicación de un documento de Automatización de Systems Manager

Después de editar su documento de automatización en VS Code, puede publicarlo en AWS.

Para publicar el documento de Automatización

1. Abra el documento de Automatización que quiere publicar siguiendo el procedimiento descrito en [Apertura de un documento de automatización de Systems Manager existente](#).
2. Haga los cambios que quiera que se publiquen. Para obtener más información, consulte [Edición de un documento de Automatización de Systems Manager](#).
3. En la parte superior derecha del archivo abierto, seleccione el ícono de carga.
4. En el cuadro de diálogo del flujo de trabajo de publicación, elija la AWS región en la que desea publicar el documento de automatización.
5. Si va a publicar un documento nuevo, elija Creación rápida. De lo contrario, seleccione Actualización rápida para actualizar un documento de automatización existente en esa AWS región.
6. Introduzca el nombre del documento de Automatización.

Al publicar una actualización de un documento de automatización existente en AWS, se agrega una nueva versión al documento.

Eliminación de un documento de Automatización de Systems Manager

Puede eliminar documentos de Automatización en VS Code. Al eliminar un documento de automatización, se elimina el documento y todas sus versiones.

Important

- La eliminación es una acción destructiva que no se puede deshacer.

- Al eliminar un documento de Automatización que ya se ha ejecutado, no se eliminan los recursos de AWS que se crearon o modificaron cuando se inició.

Para eliminar el documento de Automatización

1. Abra VS Code.
2. En la barra de navegación de la izquierda, seleccione AWS para abrir el Explorador de AWS .
3. En el AWS Explorador, en Systems Manager, abra el menú contextual (haga clic con el botón derecho) del documento que deseé eliminar y seleccione Eliminar documento.

Ejecución de un documento de Automatización de Systems Manager

Una vez publicado el documento de automatización AWS, podrá ejecutarlo para realizar tareas en su nombre en su AWS cuenta. Para ejecutar su documento de automatización, utilice el Consola de administración de AWS, el Systems Manager APIs AWS CLI, el o el Herramientas de AWS para PowerShell. Para obtener instrucciones sobre cómo ejecutar un documento de Automatización, consulte [Ejecución de una automatización sencilla](#) en la Guía del usuario de AWS Systems Manager

Como alternativa, si desea utilizar uno de los AWS SDKs junto con el Systems Manager APIs para ejecutar su documento de automatización, consulte las [referencias del AWS SDK](#).

Note

La ejecución de un documento de automatización puede crear nuevos recursos AWS e incurrir en costes de facturación. Le recomendamos encarecidamente que comprenda lo que creará su documento de Automatización en su cuenta antes de iniciar lo.

Solución de problemas de documentos de Automatización de Systems Manager en el kit de herramientas para VS Code

He guardado mi documento de automatización en VS Code, pero no lo veo en la Consola de administración de AWS.

Al guardar un documento de Automatización en VS Code, no se publica en AWS. Para obtener más información sobre la publicación del documento de automatización, consulte [Publicación de un documento de Automatización de Systems Manager](#).

Al publicar el documento de Automatización tuve un error de permisos.

Asegúrese de que su perfil de AWS credenciales tenga los permisos necesarios para publicar documentos de automatización. Si desea ver un ejemplo de política de permisos, consulte [Permisos de IAM para documentos de Systems Manager Automation](#).

He publicado mi documento de automatización en AWS, pero no lo veo en Consola de administración de AWS.

Asegúrese de haber publicado el documento en la misma AWS región en la que está navegando Consola de administración de AWS.

He eliminado mi documento de Automatización, pero se me siguen facturando los recursos que se crearon.

Al eliminar un documento de Automatización, no se eliminan los recursos que se crearon o modificaron. Puedes identificar los AWS recursos que has creado desde la [consola de administración de AWS facturación](#), explorar tus cargos y elegir qué recursos deseas eliminar desde allí.

AWS Step Functions

Con AWS Step Functions, puede crear flujos de trabajo (también denominados máquinas de estados) para crear aplicaciones distribuidas, automatizar procesos, organizar microservicios y crear canalizaciones de datos y aprendizaje automático. En los temas siguientes se describe cómo trabajar con AWS Step Functions . AWS Toolkit for Visual Studio Code Para obtener información detallada sobre el AWS Step Functions servicio, consulte la Guía para [AWS Step Functions](#) desarrolladores.

Temas

- [Trabajando con AWS Step Functions](#)
- [Trabajando con AWS Step Functions Workflow Studio](#)

Trabajando con AWS Step Functions

En las siguientes secciones se describe cómo trabajar con AWS Step Functions Amazon State Language (ASL) archivos que contienen definiciones de máquinas de estados en el AWS kit

de herramientas. Para obtener información detallada sobre las máquinas de AWS Step Functions estado, consulte el tema [Más información sobre las máquinas de estado en Step Functions](#) de la Guía para AWS Step Functionsdesarrolladores.

Visualización de máquinas de estado de Step Functions

Para ver los ASL archivos existentes que contienen definiciones de máquinas de estado en el explorador del AWS kit de herramientas, complete los siguientes pasos.

1. En el explorador del AWS kit de herramientas, expanda la región que contiene el ASL archivo que desea ver.
2. Expanda el encabezado Step Functions.
3. Los ASL archivos se muestran en el AWS Explorador.

Creación de una máquina de estado de Step Functions

En el AWS kit de herramientas, puede crear una nueva máquina de estados de Step Functions a partir de un archivo o puede usar una plantilla. En el procedimiento siguiente se describe cómo crear una máquina de estado de Step Functions a partir de un archivo. Para obtener más información sobre cómo crear una máquina de estado de SFN; a partir de una plantilla, consulte la sección Plantillas de máquina de estado que se encuentra más abajo, en este tema de la Guía del usuario.

Note

Para trabajar con Step Functions en VS Code, la extensión del archivo Amazon State Language(ASL) que contiene la definición de máquina de estado debe terminar en `asl.json`, `asl.yml` o `.asl.yaml`.

De forma predeterminada, los archivos de Step Functions pertinentes se abren en el Diseñador de flujo de trabajo. Para obtener información detallada sobre cómo trabajar en Workflow Studio mediante el AWS kit de herramientas, consulte el tema Cómo [trabajar con Workflow Studio](#) de esta Guía del usuario.

1. En el espacio de trabajo de VS Code, cree un archivo.
2. Asígnele un nombre y especifique la extensión de archivo como `asl.json`, `asl.yml` o `.asl.yaml`.

3. Tras la creación, el AWS kit de herramientas abre el nuevo archivo en AWS Step Functions Workflow Studio.
4. En Diseñador de flujo de trabajo, elija el botón Guardar del menú de utilidades para guardar el nuevo archivo de ASL.

Creación de una máquina de estado de Step Functions a partir de una plantilla

En el AWS kit de herramientas, puede crear una máquina de estados de Step Functions a partir de una plantilla. El proceso de plantilla crea un archivo de ASL que contiene una definición de máquina de estado, lo que proporciona un punto de partida para el proyecto. El siguiente procedimiento describe cómo crear una máquina de estados de Step Functions a partir de una plantilla del AWS kit de herramientas.

1. Desde el explorador del AWS kit de herramientas, expanda la región en la que desee crear una máquina de estados de Step Functions.
2. Abra el menú contextual (con el botón derecho) de Step Functions y elija Crear una máquina de estado de Step Functions para abrir el asistente Select a starter template (1/2) en VS Code.
3. En el asistente Select a starter template (1/2), elija el tipo de plantilla para la máquina de estado de Step Functions a fin de continuar.
4. En la pantalla Select template format (2/2), elija YAML o JSON como formato de plantilla.
5. Se abrirá un nuevo archivo de ASL que contiene la definición de máquina de estado en el editor de VS Code.

Descarga de una máquina de estado de Step Functions

Para descargar una máquina de estado de Step Functions almacenada de forma remota en su instancia local de VS Code, complete los siguientes pasos:

1. Desde el explorador de AWS kits de herramientas, expanda la región que contiene la máquina de estados Step Functions que desee descargar.
2. Expanda Step Functions y, a continuación, haga clic con el botón derecho en la máquina de estado de Step Functions que desea descargar y elija Descargar definición....
3. Especifique una ubicación donde almacenar la máquina de estado de Step Functions de forma local para continuar.

4. La máquina de estado de Step Functions se abre en el Diseñador de flujo de trabajo una vez completado el procedimiento.

Cómo guardar cambios en una máquina de estado de Step Functions

En el procedimiento siguiente se describe cómo guardar los cambios realizados en la máquina de estado de Step Functions.

Note

Las ediciones realizadas en el Diseñador de flujo de trabajo se sincronizan con el archivo local, pero no se guardan hasta que el trabajo se guarde en el editor de VS Code o el Diseñador de flujo de trabajo. Si el archivo local se modifica y se guarda mientras el Diseñador de flujo de trabajo está abierto y no se detectan errores en el archivo de ASL, recibirá la notificación Correcto en el Diseñador de flujo de trabajo cuando se haya completado el proceso de guardado. Sin embargo, si el archivo local contiene JSON o YAML no válido e intenta guardarlo, dicho archivo no se sincroniza y recibirá la notificación Advertencia en el Diseñador de flujo de trabajo.

1. En un archivo de ASL abierto que contenga una definición de máquina de estado en el Diseñador de flujo de trabajo, diríjase a los botones de utilidades.
2. Seleccione el botón Guardar.
3. VS Code le avisará una vez que se haya guardado el archivo.

Ejecución de una máquina de estado de Step Functions

El siguiente procedimiento describe cómo ejecutar una máquina de estados Step Functions en el AWS kit de herramientas.

1. En el explorador del AWS kit de herramientas, expanda la región que contiene la máquina de estados Step Functions que deseé ejecutar.
2. Expanda Step Functions y, a continuación, haga clic con el botón derecho en la máquina de estado de Step Functions que desea ejecutar.
3. En el menú contextual, elija Iniciar la ejecución para iniciar el proceso de lanzamiento.

4. El estado del lanzamiento se muestra en la ventana de Salida del kit de herramientas de AWS en VS Code.

Uso de fragmentos de código

Los fragmentos de código son sugerencias automatizadas que se generan en función del código en el que está trabajando. Para trabajar con fragmentos de código con Step Functions en el kit de herramientas, complete los siguientes pasos.

Note

Para trabajar con fragmentos de código de Step Functions en VS Code, la extensión del archivo de ASL que contiene la definición de máquina de estado debe terminar en `.asl.json`, `.asl.yml` o `.asl.yaml`.

De forma predeterminada, los archivos de Step Functions pertinentes se abren en el Diseñador de flujo de trabajo.

1. En VS Code, abra un archivo de ASL que contenga la definición de máquina de estado que desea modificar o bien cree un archivo de ASL.
2. En el Diseñador de flujo de trabajo, cambie al modo Código si está en el modo Diseño.
3. En el editor de código del Diseñador de flujo de trabajo, coloque el cursor en la propiedad "States".
4. Pulse **control + space** para abrir el menú de fragmentos de código; para acceder a propiedades adicionales basadas en los valores "State" "Type", pulse **control + space**.
5. Elija el fragmento de código que desee de la lista.

Validación de código

Mientras trabaja en Step Functions en el Diseñador de flujo de trabajo, la validación de código identifica los errores de forma activa y hace sugerencias respecto a lo siguiente:

- Propiedades que faltan
- Valores incorrectos
- Estado no terminal

- Estados inexistentes a los que se señala

Trabajando con AWS Step Functions Workflow Studio

En las siguientes secciones se describe cómo trabajar con AWS Step Functions Workflow Studio en AWS Toolkit for Visual Studio Code. Para obtener información detallada sobre AWS Step Functions Workflow Studio, consulte el tema [Desarrollo de flujos](#) de trabajo en la Guía para AWS Step Functionsdesarrolladores

Apertura del Diseñador de flujo de trabajo

En la siguiente lista se describen las diferentes rutas disponibles para abrir el Diseñador de flujo de trabajo en VS Code.

Note

Para trabajar con el Diseñador de flujo de trabajo en VS Code, la extensión del archivo Amazon State Language(ASL) que contiene la definición de máquina de estado debe terminar en `asl.json`, `asl.yml` o `asl.yaml`. Para obtener más información sobre cómo descargar o crear una nueva definición de máquina de estados en el AWS kit de herramientas, consulte las secciones Descarga de máquinas de estados y Creación de una máquina de estados en el AWS Step Functions tema Cómo [trabajar con](#) esta guía del usuario.

- En el AWS explorador, abra el menú contextual (haga clic con el botón derecho) en un ASL archivo que contenga una definición de máquina de estados y, a continuación, seleccione Abrir en Workflow Studio.
- En un archivo de ASL abierto que contenga una definición de máquina de estado, elija el ícono Abrir con el Diseñador de flujo de trabajo situado junto a las pestañas de la ventana del editor de VS Code.
- En un ASL archivo abierto que contenga una definición de máquina de estados, seleccione el CodeLens comando Abrir con Workflow Studio, situado en la parte superior del archivo.
- Al cerrar y volver a abrir un archivo de ASL con una definición de máquina de estado, se volverá a abrir automáticamente el archivo en el Diseñador de flujo de trabajo, a menos que el Diseñador de flujo de trabajo predeterminado se des habilite de forma manual.

Modo Diseño y modo Código

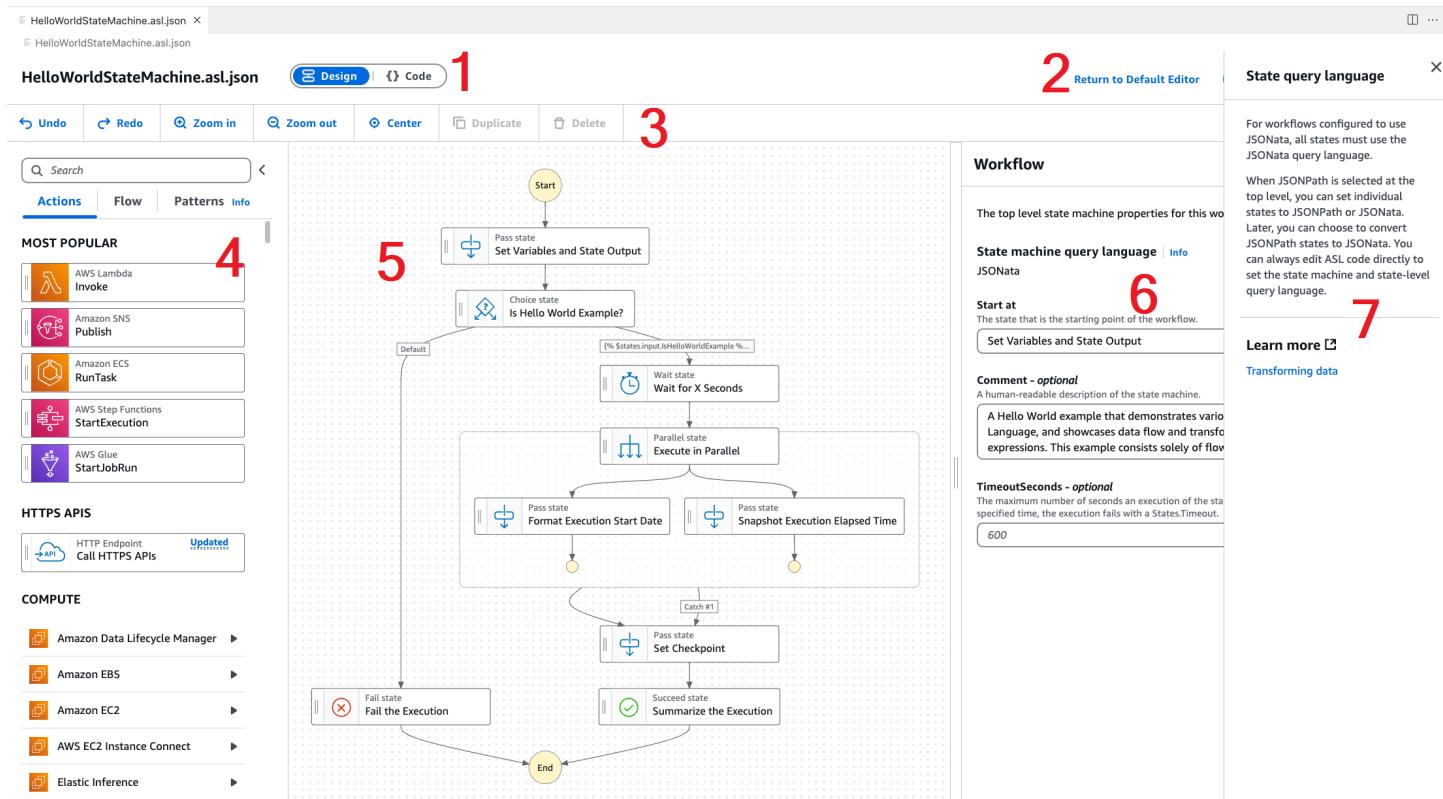
El Diseñador de flujo de trabajo tiene dos modos para trabajar con los archivos de ASL que contienen una definición de máquina de estado: el modo Diseño y el modo Código. El modo Diseño proporciona una interfaz gráfica para visualizar los flujos de trabajo a medida que crea prototipos. El modo Código tiene un editor de código integrado en el que puede ver, escribir y editar las definiciones de ASL de los flujos de trabajo.

Note

Para obtener información detallada sobre cada una de las secciones de la interfaz de usuario en los modos Diseño y Código, consulte el tema [Uso del Diseñador de flujo de trabajo](#) de la Guía para desarrolladores de AWS Step Functions. No todas las funciones de Workflow Studio están disponibles en el AWS kit de herramientas, como el modo Config, por ejemplo.

La interfaz de usuario del modo Diseño tiene siete secciones principales, tal como aparecen etiquetadas y descritas en la imagen siguiente.

1. Botones de modo: botones para cambiar entre los modos Diseño y Código.
2. Botones de utilidad: conjunto de botones para realizar tareas, como salir del Diseñador de flujo de trabajo, guardar los flujos de trabajo o exportar las definiciones de ASL a un archivo JSON o YAML.
3. Barra de herramientas de diseño: barra de herramientas que contiene un conjunto de botones que realizan acciones comunes, como deshacer, eliminar y ajustar el zoom.
4. Navegador de estados: navegador que contiene drag-and-drop los estados de su lienzo de flujo de trabajo. Los estados se organizan en pestañas y se definen como Acciones, Flujo y Patrones.
5. Lienzo y gráfico del flujo de trabajo: representación visual del flujo de trabajo en la que puede eliminar, reorganizar y seleccionar los estados para su configuración.
6. Panel inspector: vea y edite las propiedades de cualquier estado seleccionado en el lienzo. En función del estado seleccionado en el gráfico del flujo de trabajo del lienzo, las pestañas se rellenan con opciones específicas del estado para Configuración, Entrada/salida, Variables y Gestión de errores.
7. Enlaces de información: abren un panel con información contextual cuando necesita ayuda. Estos paneles también incluyen enlaces a temas relacionados en la Guía para desarrolladores de AWS Step Functions.



Uso de pruebas de estado único durante el diseño

En la interfaz de usuario de estado de prueba del Diseñador de flujo de trabajo, puede probar los estados individuales de la máquina de estado. Esto incluye la capacidad de proporcionar entradas de estado, establecer variables y realizar sustituciones tanto por CloudFormation definición como por ambas AWS SAM .

Para obtener más información sobre la infraestructura como código (IaC), las definiciones de recursos y la transformación de datos, consulte los temas [Uso AWS SAM para crear flujos de trabajo de Step Functions](#) y [Transformación de datos con JSONata Step Functions](#) en la Guía para AWS Step Functionsdesarrolladores.

En el siguiente procedimiento se describe cómo abrir la interfaz de usuario de estado de prueba en el Diseñador de flujo de trabajo.

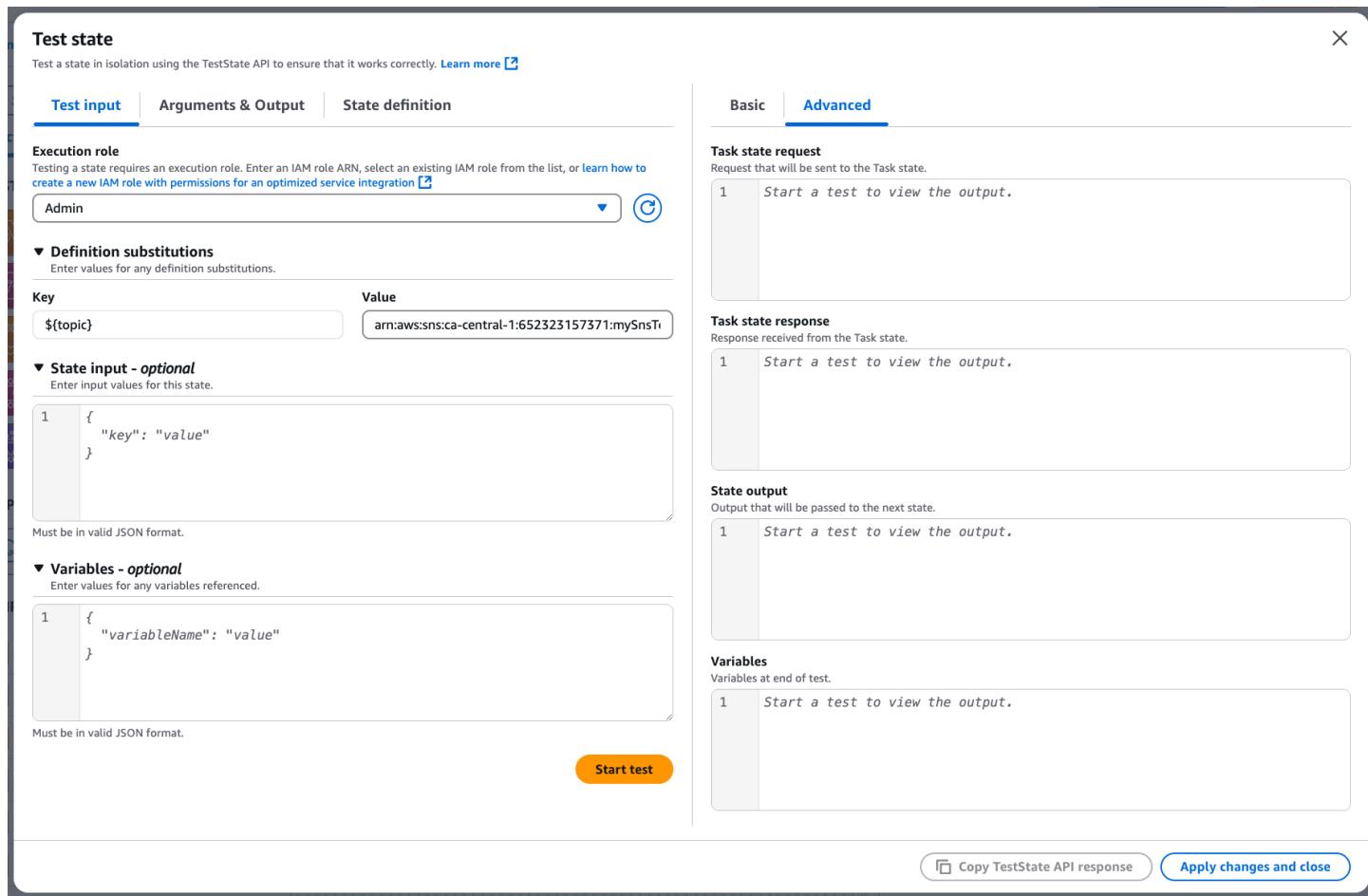
Apertura de la interfaz de usuario de estado de prueba

1. En la pestaña del modo Diseño del Diseñador de flujo de trabajo, vaya al lienzo y elija un estado para abrirlo en el panel Inspector.
2. En el panel Inspector, elija el botón Probar estado.

3. La interfaz de usuario de Probar estado se abre en VS Code.

La interfaz de usuario de estado de prueba tiene tres pestañas principales: Entrada de prueba, Argumentos y resultado y Definición de estado. La pestaña Probar entradas tiene 3 campos adicionales que le permiten introducir datos de estado, establecer variables y especificar sustituciones de definiciones a partir de sus plantillas AWS SAM o CloudFormation plantillas. En la pestaña Definición de estado, puede ajustar el flujo de trabajo y volver a probarlo. Cuando termine de ejecutar las pruebas, podrá aplicar y guardar los cambios en la definición de máquina de estado.

En la siguiente captura de pantalla se muestra la interfaz de usuario de estado de prueba, que incluye una definición de recursos del tema.



Deshabilitación del Diseñador de flujo de trabajo de forma predeterminada

De forma predeterminada, el Diseñador de flujo de trabajo es el editor predeterminado para los archivos de ASL que contienen una definición de máquina de estado. Para deshabilitar la configuración predeterminada, modifique el archivo de `settings.json` en el directorio `.vscode`

local. Si desactiva el Diseñador de flujo de trabajo de forma predeterminada, podrá seguir accediendo a él mediante los métodos que se indican en la sección Apertura del Diseñador de flujo de trabajo, que se encuentra en este tema.

Para editar el archivo de `settings.json` desde VS Code, complete los siguientes pasos.

1. En VS Code, pulse **option+shift+p** (Mac) o **ctrl+shift+p** (Windows) para abrir la paleta de comandos.
2. En la paleta de comandos de VS Code, escriba **Open User Settings (JSON)** en la barra de búsqueda y elija esta opción cuando aparezca en la lista.
3. En `settings.json` del editor, agregue la siguiente modificación al archivo:

```
{  
  "workbench.editorAssociations": {  
    // Use all the following overrides or a specific one for a  
    certain file type  
    "*.asl.json": "default",  
    "*.asl.yaml": "default",  
    "*.asl.yml": "default"  
  }  
}
```

4. Guarde los cambios en `settings.json` y actualice o reinicie VS Code.

Uso de Threat Composer

Puede usar el AWS Toolkit for Visual Studio Code para trabajar con la herramienta Threat Composer. Threat Composer es una herramienta de modelado de amenazas que puede simplificar dicho proceso.

Para obtener información detallada sobre la herramienta Threat Composer, consulte el [repositorio de GitHub de Threat Composer](#).

En los temas siguientes se describe cómo trabajar con Threat Composer en el AWS Toolkit for Visual Studio Code.

Temas

- [Uso de Threat Composer desde el kit de herramientas](#)

Uso de Threat Composer desde el kit de herramientas

Con Threat Composer, puede crear, ver y editar modelos de amenazas de Threat Composer directamente en VS Code. Para obtener información detallada sobre la herramienta Threat Composer, consulte el [repositorio de GitHub de Threat Composer](#).

En las secciones siguientes se describe cómo acceder a las herramientas de Threat Composer desde el AWS Toolkit for Visual Studio Code.

Acceso a Threat Composer desde el kit de herramientas

Hay tres formas principales de acceder a Threat Composer desde el kit de herramientas:

Acceso a Threat Composer a través de un modelo de amenazas existente

Para abrir Threat Composer, abra un archivo de modelo de amenazas (extensión `.tc.json`) existente en VS Code. Threat Composer se abre automáticamente y representa una visualización del archivo de modelo de amenazas en la ventana del editor de VS Code.

Creación de un modelo de amenazas de Threat Composer

1. En el menú principal de VS Code, expanda Archivo y, a continuación, elija Nuevo archivo.
2. En el cuadro de diálogo Nuevo archivo, elija Threat Composer File....
3. Cuando se le solicite, escriba un valor para `file name` y a continuación, pulse la tecla **enter** para abrir Threat Composer y crear una visualización del archivo de modelo de amenazas vacío en una nueva ventana del editor de VS Code.

Creación de un nuevo modelo de amenazas de Threat Composer desde la paleta de comandos

1. En VS Code, pulse **Cmd + Shift + P** o **Ctrl + Shift + P** (Windows) para abrir la paleta de comandos.
2. En el campo de búsqueda, escriba **Threat Composer** y elija Create New Threat Composer File cuando aparezca en los resultados.
3. Cuando se le solicite, escriba un valor para `file name` y a continuación, pulse la tecla **enter** para abrir Threat Composer y crear una visualización del archivo de modelo de amenazas vacío en una nueva ventana del editor de VS Code.

Uso de recursos

Además de acceder a AWS los servicios que aparecen de forma predeterminada en el AWS Explorador, también puede ir a Recursos y elegir entre cientos de recursos para añadirlos a la interfaz. En AWS, un recurso es una entidad con la que puede trabajar. Algunos de los recursos que se pueden añadir son Amazon AppFlow, Amazon Kinesis Data Streams, las funciones de IAM AWS , Amazon VPC y las distribuciones de Amazon. CloudFront

Después de hacer la selección, puede ir a Recursos y ampliar el tipo de recurso para enumerar los recursos disponibles para ese tipo. Por ejemplo, si selecciona el tipo de recurso AWS Toolkit:Lambda:Function, puede acceder a los recursos que definen distintas funciones, sus propiedades y atributos.

Después de agregar un tipo de recurso a Resources (Recursos), puede interactuar con él y sus recursos de las siguientes maneras:

- Consulte una lista de los recursos existentes que están disponibles en la AWS región actual para este tipo de recurso.
- Ver una versión de solo lectura del archivo JSON que describe un recurso.
- Copiar el identificador de recurso para el recurso.
- Consulta la AWS documentación que explica el propósito del tipo de recurso y el esquema (en formatos JSON y YAML) para modelar un recurso.
- Crear un nuevo recurso editando y guardando una plantilla con formato JSON que se ajuste a un esquema.*
- Actualizar o eliminar un recurso existente.*

Important

* En la versión actual, AWS Toolkit for Visual Studio Code la opción de crear, editar y eliminar recursos es una función experimental. Dado que las características experimentales se siguen evaluando y actualizando, es posible que tengan problemas de usabilidad, Además, es posible que las funciones experimentales se eliminen AWS Toolkit for Visual Studio Code sin previo aviso.

Para permitir el uso de características experimentales para los recursos, abra el panel Configuración en su IDE de VS Code, amplíe Extensiones y elija Kit de herramientas de AWS

En el AWS kit de herramientas Experimentos, seleccione jsonResourceModificationesta opción para poder crear, actualizar y eliminar recursos. Para obtener más información, consulte [Uso de características experimentales](#).

Permisos de IAM para acceder a los recursos

Necesita AWS Identity and Access Management permisos específicos para acceder a los recursos asociados a los AWS servicios. Por ejemplo, una entidad de IAM, como un usuario o un rol, requiere permisos de Lambda para acceder a los recursos de AWS Toolkit:Lambda::Function.

Además de los permisos para los recursos de servicio, una entidad de IAM necesita permisos para permitir que Toolkit for VS Code AWS llame a las operaciones de la API de Cloud Control en su nombre. Las operaciones de la API de Cloud Control permiten al usuario o rol de IAM acceder y actualizar los recursos remotos.

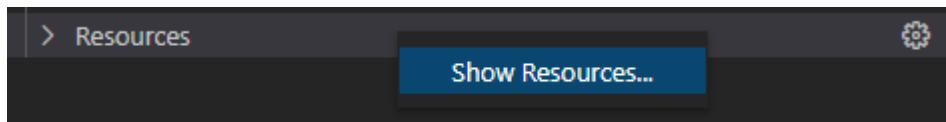
La forma más sencilla de conceder permisos es adjuntar la política AWS gestionada, PowerUserAccess, a la entidad de IAM que realiza estas operaciones de API mediante la interfaz del kit de herramientas. Esta [política administrada](#) otorga una serie de permisos para realizar tareas de desarrollo de aplicaciones, incluido llamar a las operaciones de la API.

Para obtener permisos específicos que definen las operaciones de API permitidas en recursos remotos, consulte la [Guía del usuario de la API de Cloud Control de AWS](#).

Añadir recursos existentes e interactuar con ellos

1. Desde el Explorador de AWS , haga clic con el botón derecho en Recursos y seleccione Mostrar recursos.

Un panel muestra una lista de los tipos de recurso que están disponibles para seleccionar.



2. En el panel de selección, elija los tipos de recurso que deseé añadir al Explorador de AWS y pulse la tecla Intro o seleccione Aceptar para confirmar.

Los tipos de recursos que ha seleccionado se muestran en Recursos.

 Note

Si ya ha agregado un tipo de recurso al Explorador de AWS y, a continuación, desactiva la casilla de verificación de ese tipo, ya no aparece en Recursos después de que elija Aceptar. Solo los tipos de recursos que están actualmente seleccionados están visibles en el Explorador de AWS.

3. Para ver los recursos que ya existen para un tipo de recurso, expanda la entrada de ese tipo.

En el tipo de recursos, se muestra una lista de los recursos disponibles.

4. Para interactuar con un recurso específico, haga clic con el botón derecho en el nombre y elija una de las siguientes opciones:

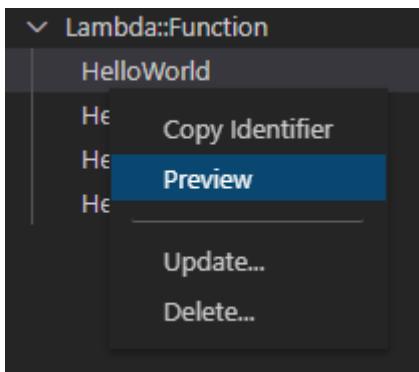
- Copiar identificador del recurso: copie el identificador del recurso específico en el portapapeles. (Por ejemplo, el recurso AWS Toolkit:DynamoDB::Table se puede identificar mediante la propiedad `TableName`).
- Preview (Vista previa): vea una versión de solo lectura de la plantilla con formato JSON que describe el recurso.

Cuando aparezca la plantilla de recursos, puede modificarla pulsando el icono de actualización situado a la derecha de la pestaña del editor. Para actualizar un recurso, debe tener activada la [???](#) necesaria.

- Actualizar: edite la plantilla con formato JSON del recurso en un editor de VS Code. Para obtener más información, consulte [Creación y edición de recursos](#).
- Eliminar: para eliminar el recurso, confirme la eliminación en el cuadro de diálogo que aparece. (La eliminación de recursos es actualmente una opción [???](#) en esta versión de AWS Toolkit for Visual Studio Code.)

 Warning

Si eliminas un recurso, cualquier AWS CloudFormation pila que utilice ese recurso no se actualizará. Para corregir este error de actualización, debes volver a crear el recurso o eliminar la referencia a él en la CloudFormation plantilla de la pila. Para obtener más información, consulte este [artículo del Centro de conocimientos](#).



Creación y edición de recursos

⚠ Important

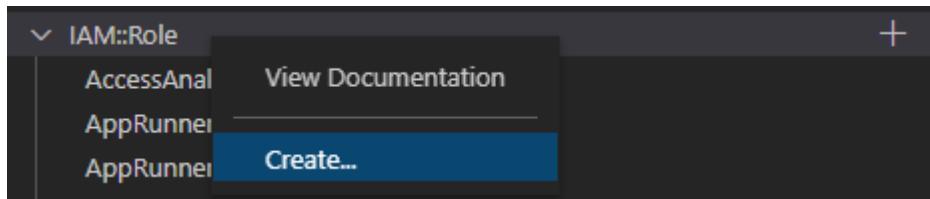
La creación y actualización de recursos es actualmente una [??? en esta versión de AWS Toolkit for Visual Studio Code.](#)

La creación de un nuevo recurso implica agregar un tipo de recurso a la lista de Recursos y, a continuación, editar una plantilla con formato JSON que defina el recurso, sus propiedades y sus atributos.

Por ejemplo, un recurso que pertenece al tipo de AWS Toolkit:SageMaker::UserProfile recurso se define con una plantilla que crea un perfil de usuario para Amazon SageMaker AI Studio. La plantilla que define este recurso de perfil de usuario debe ajustarse al esquema de tipos de recursos de AWS Toolkit:SageMaker::UserProfile. Si, por ejemplo, la plantilla no cumple con el esquema porque le faltan propiedades o porque son incorrectas, el recurso no se podrá crear ni actualizar.

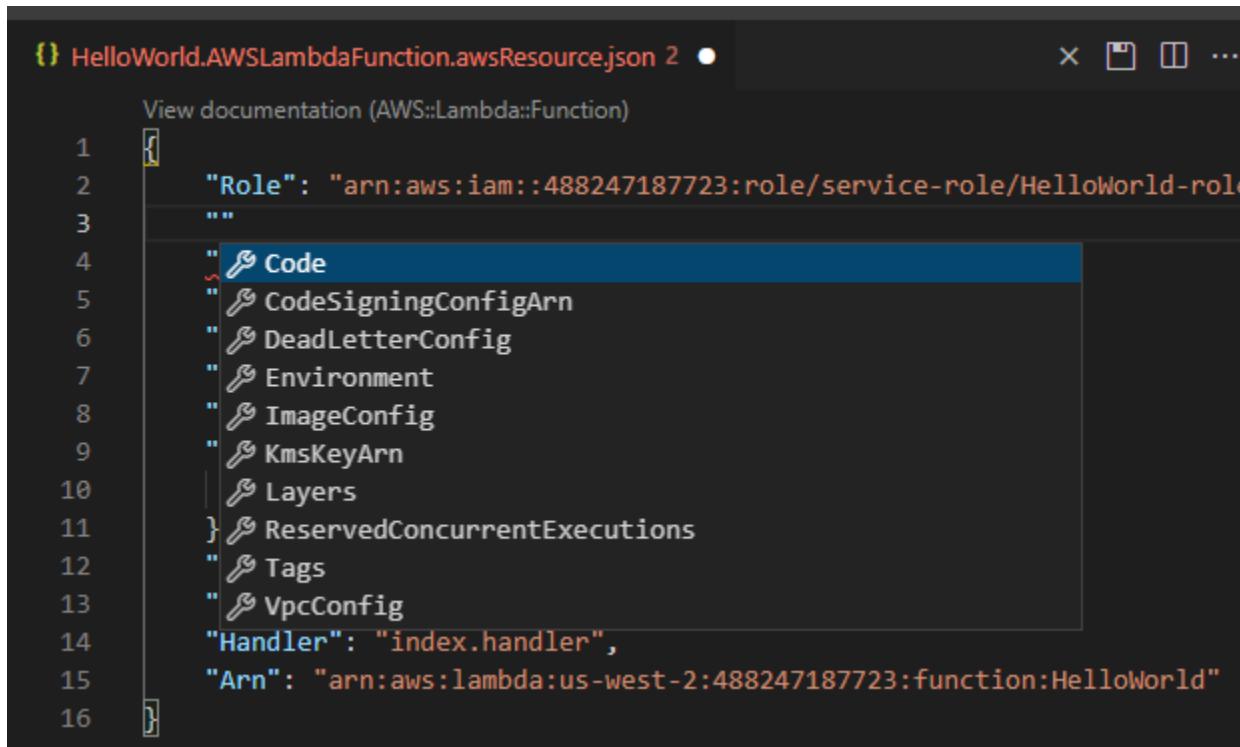
1. Agregue el tipo de recurso del recurso que desea crear haciendo clic con el botón derecho en Recursos y, luego, elija Mostrar recursos.
2. Despues de añadir el tipo de recurso en Recursos, seleccione el icono con el signo más (“+”) para abrir el archivo de plantilla en un editor nuevo.

También puede hacer clic con el botón derecho en el nombre del tipo de recurso y seleccionar Crear. También puede acceder a la información sobre cómo modelar el recurso seleccionando si elige Ver documentación.



3. En el editor, comience a definir las propiedades que componen la plantilla de recursos. La característica de autocompletar sugiere nombres de propiedades que se ajusten al esquema de la plantilla. Al pasar el ratón sobre un tipo de propiedad, un panel muestra una descripción de para qué se utiliza. Para obtener información detallada sobre el esquema, elija Ver la documentación.

Cualquier texto que no se ajuste al esquema de recursos se indica con un subrayado ondulado de color rojo.



4. Cuando termine de declarar el recurso, elija el icono Guardar para validar la plantilla y guardar el recurso en la AWS nube remota.

Si su plantilla define el recurso de acuerdo con su esquema, aparecerá un mensaje para confirmar que se ha creado el recurso. (Si el recurso ya existe, el mensaje confirma que este se actualizó).

Una vez creado el recurso, se agrega a la lista bajo el encabezado de tipo de recurso.

5. Si el archivo contiene errores, aparecerá un mensaje que explica que el recurso no se ha podido crear o actualizar. Seleccione Registro de eventos para identificar los elementos de la plantilla que necesita corregir.

Solución de problemas del AWS Toolkit for Visual Studio Code

Las siguientes secciones contienen información general sobre la solución de problemas relacionados con los AWS servicios del kit de herramientas AWS Toolkit for Visual Studio Code y su uso. Para conocer los problemas relacionados específicamente con la solución de problemas de SAM en el AWS kit de herramientas, consulte el tema [Solución de problemas de aplicaciones sin servidor](#) de esta Guía del usuario.

Temas

- [Solución de problemas y prácticas recomendadas](#)
- [El perfil ... no se ha podido encontrar en el archivo de configuración](#)
- [Esquema JSON de SAM: no se puede cambiar el esquema en el archivo template.yaml](#)

Solución de problemas y prácticas recomendadas

Las siguientes son las mejores prácticas recomendadas para solucionar problemas AWS Toolkit for Visual Studio Code . Para obtener información detallada sobre cómo puede contribuir a la AWS Toolkit for Visual Studio Code, consulte el AWS Toolkit for Visual Studio Code tema [Contribuir a](#) en el AWS Toolkit for Visual Studio Code GitHub repositorio.

- Intente recrear el problema o error antes de enviar un informe.
- Tome notas detalladas de cada paso, configuración y mensaje de error durante el proceso de recreación.
- Recopile los registros de AWS depuración de su kit de herramientas. Para obtener una descripción detallada de cómo localizar los registros de depuración del AWS kit de herramientas, consulte el procedimiento Cómo localizar los registros, que se encuentra en este tema de la guía del usuario. AWS
- Comprueba si hay solicitudes pendientes o soluciones conocidas o informa del problema no resuelto en la sección [AWS Toolkit for Visual Studio Code Problemas](#) del repositorio. AWS Toolkit for Visual Studio Code GitHub

i Note

El siguiente procedimiento describe cómo ver los registros de depuración del AWS kit de herramientas. El proceso para ver los registros de depuración de Amazon Q es idéntico, excepto que elige Amazon Q: View Logs de la paleta de comandos de VS Code.

¿Cómo localizar sus AWS Toolkit for Visual Studio Code registros de depuración?

1. En VS Code, pulse **Cmd + Shift + P** o **Ctrl + Shift + P** (Windows) para abrir la paleta de comandos y escriba **AWS View Logs** en el campo de búsqueda.
2. Elija AWS Ver registros para abrir los registros de su AWS kit de herramientas en la ventana de salida del terminal VS Code.
3. En la ventana de salida del terminal de VS Code, expanda el menú del ícono de engranaje y elija Depurar.
4. Vuelva a expandir el menú del ícono de engranaje y elija Establecer como predeterminado.
5. Pulse **Cmd + Shift + P** o **Ctrl + Shift + P** (Windows) para volver a abrir la paleta de comandos y busque **Reload Window**. A continuación, elija Desarrollador: Recargar ventana.
6. VS Code se vuelve a cargar y la ventana de salida del terminal de VS Code muestra los registros de depuración actualizados del AWS kit de herramientas.

El perfil ... no se ha podido encontrar en el archivo de configuración

Problema

i Note

Este problema solo se aplica al archivo `~/.aws/config` y no al archivo `~/.aws/credentials`. Para obtener información detallada sobre los archivos de AWS configuración y AWS credenciales, consulta el tema sobre los [archivos de configuración y credenciales compartidos](#) de la guía de referencia del AWS SDK y las herramientas.

Al elegir las credenciales, los registros del AWS kit de herramientas muestran un mensaje con esta estructura: `Profile name could not be found in shared credentials file.`

El siguiente es un ejemplo del aspecto de este error en los registros de AWS Toolkit:

```
2023-08-08 18:20:45 [ERROR]: _aws.auth.reauthenticate: Error: Unable to
authenticate connection
-> CredentialsProviderError: Profile vscode-prod-readonly could not be found
in shared credentials file.
```

Solución

Si su perfil ya existe en `~/.aws/config`, compruebe que empieza por `[profile]`. A continuación, se muestra un ejemplo de un perfil de usuario estructurado de forma correcta:

```
[profile example]
region=us-west-2
credential_process=...
```

A continuación, se muestra un ejemplo de un perfil de usuario estructurado de forma incorrecta:

```
[example]
region=us-west-2
credential_process=...
```

Esquema JSON de SAM: no se puede cambiar el esquema en el archivo `template.yaml`

Problema

No puede seleccionar de forma manual un esquema JSON distinto en SAM `template.yaml`

Solución

Tras actualizar a la versión 1.11 o a una versión posterior de `vscode-yaml`, puede añadir una línea de modo **`yaml-language-server`** en la parte superior de un archivo YAML para forzar el uso de un esquema por parte del URI. Para obtener información adicional sobre el [uso de esquemas](#)

[integrados](#), en el tema sobre el servidor de lenguaje yaml del repositorio para desarrolladores de Redhat. GitHub A continuación, se muestra un ejemplo de una línea de modo **yaml-language-server**:

```
# yaml-language-server: $schema=https://raw.githubusercontent.com/aws/
serverless-application-model/main/samtranslator/schema/schema.json
```

Seguridad de AWS Toolkit for Visual Studio Code

Temas

- [Protección de los datos en AWS Toolkit for Visual Studio Code](#)

Protección de los datos en AWS Toolkit for Visual Studio Code

El [modelo de responsabilidad compartida](#) de AWS se aplica a la protección de datos del kit de herramientas de AWS para Visual Studio Code. Como se describe en este modelo, AWS es responsable de proteger la infraestructura global que ejecuta toda la Nube de AWS. Eres responsable de mantener el control sobre el contenido alojado en esta infraestructura. También eres responsable de las tareas de administración y configuración de seguridad para los Servicios de AWS que utiliza. Para obtener más información sobre la privacidad de los datos, consulta las [Preguntas frecuentes sobre la privacidad de datos](#). Para obtener información sobre la protección de datos en Europa, consulta la publicación de blog sobre el [Modelo de responsabilidad compartida de AWS y GDPR](#) en el Blog de seguridad de AWS.

Con fines de protección de datos, recomendamos proteger las credenciales de la Cuenta de AWS y configurar cuentas de usuario individuales con AWS IAM Identity Center o AWS Identity and Access Management (IAM). De esta manera, solo se otorgan a cada usuario los permisos necesarios para cumplir sus obligaciones laborales. También recomendamos proteger sus datos de la siguiente manera:

- Utiliza la autenticación multifactor (MFA) en cada cuenta.
- Utiliza SSL/TLS para comunicarse con los recursos de AWS. Se recomienda el uso de TLS 1.2 y recomendamos TLS 1.3.
- Configure los registros de API y de actividad de los usuarios con AWS CloudTrail. Para obtener información sobre cómo utilizar registros de seguimiento de CloudTrail para capturar actividades de AWS, consulta [Working with CloudTrail trails](#) en la Guía del usuario de AWS CloudTrail.
- Utiliza las soluciones de cifrado de AWS, junto con todos los controles de seguridad predeterminados dentro de los servicios de Servicios de AWS.
- Utiliza servicios de seguridad administrados avanzados, como Amazon Macie, que lo ayuden a detectar y proteger los datos confidenciales almacenados en Amazon S3.
- Si necesita módulos criptográficos validados FIPS 140-3 al acceder a AWS a través de una interfaz de línea de comandos o una API, utiliza un punto de conexión de FIPS. Para obtener

más información sobre los puntos de conexión de FIPS disponibles, consulta [Estándar de procesamiento de la información federal \(FIPS\) 140-3](#).

Se recomienda encarecidamente no introducir nunca información confidencial o sensible, como por ejemplo, direcciones de correo electrónico de clientes, en etiquetas o campos de formato libre, tales como el campo Nombre. Esto incluye el uso del kit de herramientas de AWS para Visual Studio Code u otros Servicios de AWS con la consola, la API, la AWS CLI o los SDK de AWS. Cualquier dato que ingrese en etiquetas o campos de texto de formato libre utilizados para nombres se puede emplear para los registros de facturación o diagnóstico. Si proporciona una URL a un servidor externo, recomendamos encarecidamente que no incluya información de credenciales en la URL a fin de validar la solicitud para ese servidor.

Historial de documentos para la guía del usuario de AWS Toolkit for Visual Studio Code

En la siguiente tabla se describen los cambios importantes en cada versión del AWS Toolkit for Visual Studio Code. Para obtener notificaciones sobre las actualizaciones de esta documentación, puede suscribirse a una [fuente RSS](#).

Cambio	Descripción	Fecha
Estudio unificado de Amazon SageMaker	Integración con el servicio Estudio unificado de Amazon SageMaker.	18 de septiembre de 2025
LocalStack	Se ha agregado un tema nuevo a la Guía del usuario para respaldar el lanzamiento de LocalStack.	11 de septiembre de 2025
Trabajo con funciones de AWS Lambda	Actualización del tema de la Guía del usuario para incluir características de Lambda actualizadas en el kit de herramientas.	17 de julio de 2025
Depuración remota de AWS Lambda	Se ha agregado el tema nuevo Depuración remota de AWS Lambda a la Guía del usuario de AWS Toolkit for Visual Studio Code.	17 de julio de 2025
De la consola de AWS Lambda al IDE	Se ha agregado el tema nuevo De la AWS Lambda console al IDE a la Guía del usuario del AWS Toolkit for Visual Studio Code.	17 de julio de 2025

<u>Actualizaciones del contenido de AWS Step Functions y adición de compatibilidad con el Diseñador de flujo de trabajo</u>	Se han agregado actualizaciones al contenido existente de AWS Step Functions y al tema de la Guía del usuario del Diseñador de flujo de trabajo de AWS Step Functions con el fin de respaldar el lanzamiento de la característica.	6 de marzo de 2025
<u>AWS Serverless Land</u>	Se ha agregado el tema nuevo AWS Serverless Land a la tabla de contenido del Desarrollador de aplicaciones de AWS.	6 de marzo de 2025
<u>Actualización de firewalls y puertas de enlace para permitir el acceso</u>	Listas de puntos de conexión y recursos que deben figurar en la lista de permitidos para acceder a todos los servicios y características del AWS Toolkit for Visual Studio Code y las extensiones de Amazon Q para VS Code.	28 de febrero de 2025
<u>Compatibilidad con Amazon ECR App Runner</u>	Se ha agregado documentación de ayuda para lanzar un servicio AWS App Runner desde el nodo Amazon Elastic Container Registry, en el kit de herramientas de AWS.	6 de febrero de 2025
<u>Amazon DocumentDB</u>	Se ha agregado un tema nuevo de Amazon DocumentDB a la Guía del usuario de AWS Toolkit for Visual Studio Code.	6 de febrero de 2025

<u>Compatibilidad con EC2</u>	Se ha agregado compatibilidad con el servicio Amazon Elastic Compute Cloud al kit de herramientas.	31 de enero de 2025
<u>AWS Documentos de</u>	Se ha agregado un tema nuevo a la Guía del usuario para los documentos de AWS.	20 de enero de 2025
<u>Live Tail de Registros de Amazon CloudWatch</u>	Se ha agregado un nuevo subtema de soporte para la característica Live Tail de Registros de Amazon CloudWatch en el AWS Toolkit for Visual Studio Code.	15 de diciembre de 2024
<u>Desarrollador de aplicaciones de AWS</u>	Se ha agregado el tema nuevo Desarrollador de aplicaciones de AWS a la Guía del usuario de AWS Toolkit for Visual Studio Code.	30 de octubre de 2024
<u>Infrastructure Composer</u>	AWS Application Composer es ahora AWS Infrastructure Composer.	3 de octubre de 2024
<u>Actualizaciones del Analizador de acceso de AWS Identity and Access Management (AWS IAM)</u>	Se ha actualizado el contenido del Analizador de acceso de AWS IAM para incluir nuevas referencias de API.	10 de julio de 2024
<u>Analizador de acceso de AWS Identity and Access Management (AWS IAM)</u>	Se ha agregado un tema nuevo a la Guía del usuario para el Analizador de acceso de AWS IAM.	23 de mayo de 2024

<u>Actualización de la conexión al flujo de autorización de AWS</u>	El flujo de autorización se ha actualizado para reflejar los cambios en el proceso de autenticación y la separación de Amazon Q del AWS Toolkit for Visual Studio Code.	30 de abril de 2024
<u>Extensión de Amazon Q para VS Code</u>	A partir del 30 de abril de 2024, CodeWhisperer forma parte de Amazon Q y este último está disponible como extensión para VS Code.	30 de abril de 2024
<u>Compatibilidad con la nube privada virtual en entornos de desarrollo</u>	Contenido actualizado que cubre los cambios en la interfaz de usuario para admitir la nube privada virtual (VPC) en entornos de desarrollo.	21 de enero de 2024
<u>Infrastructure Composer</u>	Se ha agregado el tema nuevo de Infrastructure Composer a la Guía del usuario de AWS Toolkit for Visual Studio Code.	28 de noviembre de 2023
<u>Compatibilidad de SSO con CodeCatalyst</u>	Contenido actualizado para cubrir la compatibilidad de IAM Identity Center con CodeCatalyst y los entornos de desarrollo.	17 de noviembre de 2023
<u>Adición de enlaces de descarga de VS Code y el kit de herramientas</u>	Contenido actualizado con enlaces de descarga para VS Code y el AWS Toolkit for Visual Studio Code.	1 de noviembre de 2023

<u>Tema de Amazon Redshift</u>	Se ha agregado un tema nuevo de Amazon Redshift a la Guía del usuario de AWS Toolkit for Visual Studio Code.	17 de octubre de 2023
<u>Actualización de la conexión al flujo de autorización de AWS</u>	El flujo de autorización se ha actualizado para centrarse en métodos de autenticación específicos del servicio.	29 de septiembre de 2023
<u>Guía de usuario creada: Crear una plantilla de CloudFormation</u>	Se ha creado una nueva guía del usuario que describe cómo crear una plantilla de CloudFormation con el kit de herramientas para VS Code	17 de diciembre de 2021
<u>Actualización menor de la IU</u>	Se cambiado el texto “Vista previa del estado de la máquina” por “Gráfico de representación” porque se adapta mejor a la interfaz de usuario.	14 de diciembre de 2021
<u>Guía del usuario creada para Amazon Elastic Container Service Exec</u>	Esta es una descripción general de Amazon ECS Exec.	13 de diciembre de 2021
<u>Guía de usuario creada para el servicio AWS IoT del kit de herramientas para VS Code</u>	El objetivo de esta guía del usuario es ayudarle a empezar a utilizar el servicio AWS IoT del kit de herramientas para VS Code.	22 de noviembre de 2021
<u>Compatibilidad con características experimentales</u>	Se ha añadido compatibilidad para activar características experimentales en los servicios de AWS.	14 de octubre de 2021

<u>Compatibilidad con recursos de AWS</u>	Se ha añadido compatibilidad para acceder a tipos de recursos junto con opciones de interfaz para crear, editar y eliminar recursos.	14 de octubre de 2021
<u>Información general sobre el servicio Amazon ECR para el AWS Toolkit for Visual Studio Code</u>	Se ha agregado información general y un recorrido por las características y funciones del servicio Amazon ECR que son accesibles en VS Code.	14 de octubre de 2021
<u>Compatibilidad con entornos ARM</u>	Ahora puede ejecutar aplicaciones sin servidor en entornos emulados basados en ARM64, así como en entornos basados en x86_64.	1 de octubre de 2021
<u>Aplicación sin servidor de AWS</u>	Se ha añadido compatibilidad para ejecutar aplicaciones de AWS SAM en una Plataforma ARM64	30 de septiembre de 2021
<u>Actualización del formato de la sección Node.js</u>	Se ha actualizado el formato de Node.js/ TypeScript según los comentarios de los clientes.	12 de agosto de 2021
<u>Compatibilidad con App Runner</u>	Se ha agregado soporte para AWS App Runner para AWS Toolkit for Visual Studio Code.	11 de agosto de 2021
<u>Depuración de las funciones de Go</u>	Se ha agregado compatibilidad para depurar las funciones locales de Go.	10 de mayo de 2021

<u>Depuración de las funciones de Java</u>	Se ha agregado compatibilidad para depurar las funciones locales de Java.	22 de abril de 2021
<u>Compatibilidad de YAML con AWS Step Functions</u>	Se ha agregado compatibilidad de YAML con AWS Step Functions.	4 de marzo de 2021
<u>Depuración de recursos de Amazon API Gateway</u>	Se ha añadido compatibilidad con la depuración de recursos locales de Amazon API Gateway.	1 de diciembre de 2020
<u>Amazon API Gateway</u>	Se ha agregado compatibilidad con Amazon API Gateway.	1 de diciembre de 2020
<u>Aplicación sin servidor de AWS</u>	Se ha añadido compatibilidad de las imágenes de contenedor de Lambda con las aplicaciones sin servidor.	1 de diciembre de 2020
<u>AWS Systems Manager Compatibilidad con</u>	Se ha añadido compatibilidad con los documentos de Automatización de Systems Manager.	30 de septiembre de 2020
<u>Registros de CloudWatch</u>	Se ha agregado compatibilidad para Registros de CloudWatch.	24 de agosto de 2020
<u>Amazon S3</u>	Se ha agregado compatibilidad con Amazon S3.	30 de julio de 2020
<u>AWS Step Functions Compatibilidad con</u>	Se agregó compatibilidad con la de para AWS Step Functions.	31 de marzo de 2020

<u>Contenido de seguridad</u>	Se ha añadido contenido de seguridad.	6 de febrero de 2020
<u>Trabajo con esquemas de Amazon EventBridge</u>	Se ha agregado compatibilidad con esquemas de Amazon EventBridge	1 de diciembre de 2019
<u>AWS CDK</u>	Versión de prueba del servicio AWS CDK.	25 de noviembre de 2019
<u>Uso de un proceso de credenciales externo</u>	Se ha añadido información sobre el uso de un proceso de credenciales externo para obtener credenciales de AWS.	25 de septiembre de 2019
<u>Uso de IntelliSense para archivos de definición de tareas</u>	Se ha agregado compatibilidad con IntelliSense para trabajar con archivos de definición de tareas de Amazon ECS.	24 de septiembre de 2019
<u>Guía del usuario del AWS Toolkit for Visual Studio Code</u>	Versión de disponibilidad general.	11 de julio de 2019
<u>Guía del usuario del AWS Toolkit for Visual Studio Code</u>	Se actualizó la estructura del documento para mayor claridad y facilidad de uso.	3 de julio de 2019
<u>Instalación del AWS Toolkit for Visual Studio Code</u>	Se agregó información sobre la instalación de los SDK de lenguaje para admitir varias cadenas de herramientas.	12 de junio de 2019
<u>Configuración de la cadena de herramientas</u>	Se ha añadido información sobre cómo configurar diferentes cadenas de herramientas.	12 de junio de 2019

Versión inicial

Versión inicial de la Guía del usuario del AWS Toolkit for Visual Studio Code.

28 de marzo de 2019

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