



Guía del usuario

AWS Kit de herramientas para VS Code



AWS Kit de herramientas para VS Code: Guía del usuario

Copyright © 2024 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 el 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
Abrir el panel de inicio de sesión	7
Conectarse a AWS desde el kit de herramientas	7
Autenticación para Amazon CodeCatalyst	8
AWS Regiones cambiantes	9
Añadir una región al AWS explorador	10
Oculta una región del AWS explorador	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	11
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

AWS ID de constructor	17
Uso de un proceso de credenciales externo	17
Trabajando con AWS	18
Características experimentales	19
AWS Explorador	19
Amazon CodeCatalyst	20
¿Qué es Amazon CodeCatalyst?	20
Introducción a CodeCatalyst	21
Uso de recursos de Amazon CodeCatalyst	24
Uso de entornos de desarrollo	28
Solución de problemas	32
Amazon API Gateway	33
AWS App Runner	33
Requisitos previos	34
Precios	37
Creación de servicios de App Runner	38
Administración de los servicios de App Runner	41
AWS Application Composer	44
Trabajo con componentes de la aplicación AWS	44
AWS CDK	45
Aplicaciones de AWS CDK	46
Pilas de AWS CloudFormation	48
Eliminación de una pila de AWS CloudFormation	48
Cree una CloudFormation plantilla	49
Registros de Amazon CloudWatch	51
Visualización de grupos de registros y flujos de registro de CloudWatch	51
Uso de eventos de CloudWatch Logs	53
Buscar grupos de registro	55
Amazon ECR	57
Requisitos previos	58
Uso de Amazon ECR con el kit de herramientas para VS Code	60
Amazon ECS	69
Uso de IntelliSense para archivos de definición de tareas	69
Amazon ECS Exec	70
Amazon EventBridge	73
Trabajar con Amazon EventBridge Schemas	73

AWS Analizador de acceso IAM	75
Trabajando con AWS IAM Access Analyzer	76
AWS IoT	80
Requisitos previos de AWS IoT	80
Objetos de AWS IoT	81
AWS IoT Certificados de	82
Políticas de AWS IoT	86
Funciones AWS Lambda	89
Interacción con funciones de Lambda remotas	89
Amazon Redshift	96
Uso de Amazon Redshift	96
Amazon S3	101
Trabajo con recursos de S3	102
Trabajo con objetos de S3	104
Aplicación sin servidor de AWS	108
Introducción	108
Ejecución y depuración de funciones de Lambda directamente desde el código	117
Ejecución y depuración de recursos locales de Amazon API Gateway	121
Opciones de configuración para depurar aplicaciones sin servidor	125
Solución de problemas	134
AWS Systems Manager	136
Hipótesis y requisitos previos	136
Permisos de IAM para documentos de Systems Manager Automation	137
Creación de un nuevo documento de Automatización de Systems Manager	138
Apertura de un documento de automatización de Systems Manager existente	138
Edición de un documento de Automatización de Systems Manager	139
Publicación de un documento de Automatización de Systems Manager	140
Eliminación de un documento de Automatización de Systems Manager	140
Ejecución de un documento de Automatización de Systems Manager	141
Solución de problemas	141
AWS Step Functions	142
AWS Step Functions y VS Code	142
Threat Composer	153
Trabajando con Threat Composer	153
Recursos	154
Permisos de IAM para acceder a los recursos	155

Añadir recursos existentes e interactuar con ellos	156
Creación y edición de recursos	157
Seguridad	160
Protección de los datos	160
Historial de documentos	162
.....	clxvi

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 el 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. Descarga la [extensión Amazon Q Developer IDE desde VS Code Marketplace](#) para empezar.

Para obtener más información sobre el servicio Amazon Q Developer, consulte la Guía del usuario para [desarrolladores de Amazon Q](#). 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 de AWS Toolkit for Visual Studio Code](#)
- [Conectarse a AWS](#)
- [AWS Regiones cambiantes](#)
- [Configuración de la cadena de herramientas](#)

Instalación de 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 1.42.0 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.

Abrir el panel de inicio de sesión

Complete uno de los siguientes procedimientos para abrir el panel de inicio de sesión del AWS kit de herramientas.

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

1. Desde el AWS Toolkit for Visual Studio Code, expande EXPLORER.
2. Amplíe Más acciones... menú seleccionando el... icono.
3. De Más acciones... menú, seleccione Conectar AWS a para abrir el panel de inicio de sesión del AWS kit de herramientas.

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

1. Abre la paleta de comandos pulsando **Shift+Command+P** (**Ctrl+Shift+P**Windows).
2. Entra **AWS: Add a New Connection** en el campo de búsqueda.
3. Seleccione esta opción **AWS: Add a New Connection** para abrir el panel de inicio AWS de sesión del kit de herramientas.

Conectarse a AWS desde el kit de herramientas

Autenticar y conectarse con el SSO

Para autenticarse y conectarse mediante el AWS uso 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 de inicio de sesión del AWS kit de herramientas, seleccione la pestaña Workforce y, a continuación, pulse el botón Continuar para continuar.
2. En el panel Iniciar sesión con IAM Identity Center, introduzca la URL de inicio de su organización. Un administrador o el servicio de asistencia de su empresa le proporcionará esta URL.
3. Seleccione 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 AWS autorización 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 de inicio de sesión del AWS kit de herramientas, elija la credencial de IAM y, a continuación, pulse el botón Continuar para continuar.
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, autentícate 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 .

Autentícase y conéctate con un Builder ID AWS

1. En el panel de inicio de sesión del AWS kit de herramientas, selecciona la pestaña Workforce y, a continuación, selecciona el botón Continuar para continuar.
2. En la parte superior del panel Iniciar sesión con SSO, selecciona el enlace Saltar 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.

Autentíquese y conéctese con el Centro de identidades de IAM

1. En el panel de inicio de sesión del AWS kit de herramientas, selecciona la pestaña Workforce y, a continuación, selecciona el botón Continuar para continuar.
2. En el panel Iniciar sesión con IAM Identity Center, introduzca la URL de inicio de su organización. Un administrador o el servicio de asistencia de su empresa le proporcionará esta URL.
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 AWS autorización 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.

AWS Regiones cambiantes

Una AWS región específica dónde se administran sus AWS recursos. Tu AWS región predeterminada se detecta cuando te conectas a tu AWS cuenta desde el AWS Toolkit for Visual Studio Code, y se muestra automáticamente en el AWS Explorador.

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

Añadir una región al AWS explorador

Complete el siguiente procedimiento para añadir una región al AWS explorador.

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 las AWS regiones que desee añadir al AWS explorador.
4. Pulse el botón Aceptar para confirmar sus elecciones y actualizar el AWS Explorador.

Ocultar una región del AWS explorador

Para ocultar una región de la vista del AWS explorador, complete el siguiente procedimiento.

1. En el AWS Explorador, localice la AWS región que desee ocultar.
2. Abra el menú contextual (clic con el botón derecho) de la región que desea eliminar.
3. Seleccione Mostrar u ocultar regiones para abrir las opciones AWS: Mostrar u ocultar regiones en VS Code.
4. Deseleccione las regiones que desee ocultar en la vista del AWS explorador.

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 `gopl`.
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 necesita autenticarse para empezar AWS a trabajar con. AWS Toolkit for Visual Studio Code Sin embargo, la mayoría de AWS los recursos se administran a través de una AWS cuenta. Para acceder a todos los AWS Toolkit for Visual Studio Code servicios y funciones, tendrás que autenticarte con AWS IAM Identity Center credenciales de AWS Builder ID o IAM.

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

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

Temas

- [AWS Centro de identidad de IAM](#)
- [AWS Credenciales de IAM](#)
- [AWS Builder ID para desarrolladores](#)
- [Uso de un proceso de credenciales externo](#)

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 IAM Identity Center para los kits de desarrollo de software (SDK), consulte la sección [Autenticación de IAM Identity Center](#) en la Guía de referencia de los SDK y herramientas de AWS .

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 de herramientas y AWS SDK.

Note

El paso 3: Actualizar el archivo de credenciales compartidas de la Guía de referencia de los SDK y las herramientas de AWS es opcional.

Si completas el paso 3, AWS Toolkit for Visual Studio Code detectará automáticamente tus credenciales según se indica a continuación. [the section called “Crear un archivo de credenciales compartido a partir del AWS Toolkit for Visual Studio Code”](#)

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. Abra la paleta de comandos pulsando **Shift+Command+P (Ctrl+Shift+P)**Windows).
2. Entra **AWS: Add a New Connection** en el campo de búsqueda.
3. Seleccione esta opción **AWS: Add a New Connection** para abrir el panel de inicio AWS de sesión del kit de herramientas.
4. En el panel de inicio de sesión del AWS kit de herramientas, elija la credencial de IAM y, a continuación, seleccione el botón Continuar para continuar.
5. Introduzca las letras **Profile Name** **Access 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
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, se presupone que `[Profile_Name]` contiene errores de sintaxis y origina un error de autenticación.

```
[Profile_Name]  
xaws_access_key_id= AKIAI44QH8DHBEXAMPLE
```

```
xaws_secret_access_key= wJa1rXUtnFEMI/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 = wJa1rXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
[Profile2_Name]
aws_access_key_id = AKIAI44QH8DHBEXAMPLE
```

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

AWS Builder ID para desarrolladores

Un AWS Builder ID es una AWS cuenta adicional que es opcional o necesaria para determinados AWS servicios. Para obtener información detallada sobre el método de autenticación de AWS Builder ID, consulte el tema [Iniciar sesión con AWS Builder ID](#) en la Guía del usuario de AWS inicio de sesión.

Para obtener más información sobre cómo autenticar y conectar el AWS kit de herramientas con tu ID de AWS Builder existente, consulta el AWS tema [Conectar a](#) de esta Guía del usuario.

Uso de un proceso de credenciales externo

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

La modificación del `shared config file` para los procesos de obtención de credenciales es la misma para el AWS Toolkit for Visual Studio Code y la 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 .

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. La siguiente es una lista de temas de guía que cubren cada servicio del Toolkit for VS Code y sus características. Elija un servicio o herramienta para obtener más información sobre lo que hace, cómo configurarlo y cómo trabajar con las funciones básicas.

Temas

- [Uso de características experimentales](#)
- [Trabajar con AWS los servicios en el AWS explorador](#)
- [Amazon CodeCatalyst para VS Code](#)
- [Uso de Amazon API Gateway](#)
- [Utilizándolo AWS App Runner con AWS Toolkit for Visual Studio Code](#)
- [AWS Application 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](#)
- [Uso del servicio Amazon Elastic Container Registry](#)
- [Trabajar con Amazon Elastic Container Service](#)
- [Trabajo con Amazon EventBridge](#)
- [AWS Analizador de acceso IAM](#)
- [Uso de AWS IoT en AWS Toolkit for Visual Studio Code](#)
- [Trabajo con funciones de AWS Lambda](#)
- [Amazon Redshift en el kit de herramientas para VS Code](#)
- [Uso de Amazon S3](#)
- [Uso de aplicaciones sin servidor](#)
- [Uso de documentos de Automatización de Systems Manager](#)
- [Trabajando con AWS Step Functions](#)
- [Trabajando con Threat Composer](#)

- [Usar recursos de](#)

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.

Trabajar con AWS los servicios en el AWS explorador

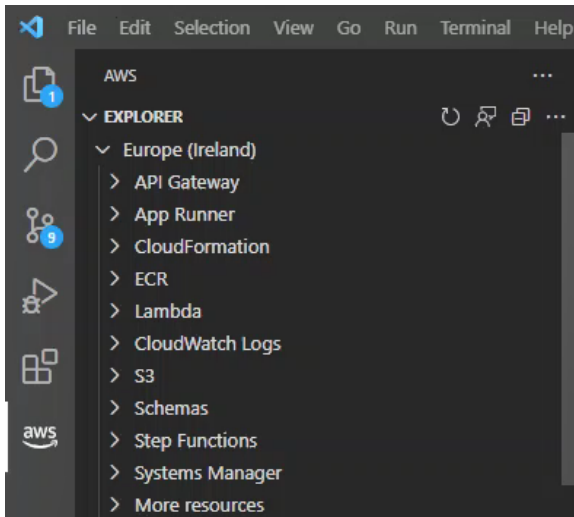
El AWS Explorador le ofrece una vista de algunos de los AWS servicios con los que puede trabajar al utilizar el AWS Toolkit for Visual Studio Code.

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:



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

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

AWS

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. Con el AWS Toolkit for Visual Studio Code, puede ver y administrar sus recursos de CodeCatalyst directamente desde VS Code. También puede trabajar directamente en la nube utilizando el kit de herramientas de AWS para lanzar entornos de computación virtuales en entornos

de desarrollo que ejecuten VS Code. Para obtener más información sobre el servicio CodeCatalyst, consulte la Guía del usuario de [Amazon CodeCatalyst](#).

En los temas siguientes se describe cómo conectar VS Code con CodeCatalyst y cómo trabajar con CodeCatalyst desde el kit de herramientas para VS Code.

Temas

- [Introducción a CodeCatalyst y al kit de herramientas para VS Code](#)
- [Uso de recursos de Amazon CodeCatalyst en VS Code](#)
- [Trabajo en entornos de desarrollo y VS Code](#)
- [Solución de problemas de Amazon CodeCatalyst y VS Code](#)

Introducción a CodeCatalyst y al kit de herramientas para VS Code

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

Temas

- [Instalación de VS Code](#)
- [Instalación del kit de herramientas para VS Code](#)
- [Creación de una cuenta de CodeCatalyst](#)
- [Conexión del kit de herramientas de AWS con CodeCatalyst](#)

Instalación de VS Code

Antes de comenzar, asegúrese de que utiliza la última versión de VS Code. Para descargar la última versión de VS Code, visite el sitio web de [descargas de VS Code](#).

Instalación del kit de herramientas para VS Code

Instale la última versión de la extensión del kit de herramientas para VS Code para conectar el kit de herramientas con su cuenta de CodeCatalyst. Puede encontrar la versión más reciente y completar la instalación del kit de herramientas de AWS en Extensiones: Marketplace en VS Code.

Para abrir Extensiones: Marketplace en VS Code e instalar el kit de herramientas para VS Code, complete los siguientes pasos:

1. En la barra de actividades de VS Code, situada en el extremo izquierdo de la pantalla principal, seleccione el icono de Extensiones para abrir la vista EXTENSIONES: MARKETPLACE.

Note

También puede usar la paleta de comandos de VS Code para abrir la vista Extensiones: Marketplace. Para ello, abra la paleta de comandos pulsando Command+Shift+P (macOS) o Ctrl+Shift+P (Windows). A continuación, introduzca `Extensions: Focus on Marketplace View` en el campo de búsqueda proporcionado y seleccione `Extensiones: Foco en vista Marketplace` para abrir la vista de Marketplace.

2. En la barra de búsqueda de EXTENSIONES: MARKETPLACE, introduzca `AWS Toolkit`.
3. Los resultados de la búsqueda se filtran a medida que escribe. Cuando la extensión del kit de herramientas de AWS esté disponible, seleccione `Instalar` para empezar a instalar la versión más reciente del kit de herramientas de AWS.

Creación de una cuenta de CodeCatalyst

Además de instalar la versión más reciente del kit de herramientas de AWS, debe tener un ID de creador de AWS activo para conectar CodeCatalyst con el kit de herramientas para VS Code. Si no tiene un ID de creador de AWS o cuenta de CodeCatalyst, consulte la sección [Configuración con CodeCatalyst](#) de la Guía del usuario de CodeCatalyst.

Note

Un ID de creador de AWS es diferente de sus credenciales de IAM. Las credenciales de IAM son necesarias para la mayoría de Servicios de AWS accesibles desde el kit de herramientas de AWS. Por el contrario, se necesita un ID de creador de AWS para crear una nueva cuenta de CodeCatalyst y conectarla al kit de herramientas de AWS.

Conexión del kit de herramientas de AWS con CodeCatalyst

Para conectar el kit de herramientas de AWS con su cuenta de CodeCatalyst, complete los siguientes pasos:


1. En el kit de herramientas para VS Code, expanda el explorador de HERRAMIENTAS PARA DESARROLLADORES.

2. En el explorador de HERRAMIENTAS PARA DESARROLLADORES, expanda CodeCatalyst y seleccione Iniciar.
3. Cuando se lo solicite VS Code, abra el portal del ID de creador de AWS en su navegador web preferido para continuar.

 Note

Si no dispone de un ID de creador de AWS, utilice el enlace proporcionado para crear uno. A continuación, complete los siguientes pasos para autenticarse y conectar las credenciales de su ID de creador de AWS al kit de herramientas para VS Code.

4. Desde el portal del ID de creador de AWS, siga los pasos para autenticar su ID de creador de AWS. Cuando se le solicite, seleccione Permitir para que el kit de herramientas para VS Code pueda acceder a sus datos.
5. El navegador se actualiza automáticamente cuando las credenciales del ID de creador de AWS se conectan correctamente al kit de herramientas para VS Code. Esto indica que es seguro cerrar el navegador y volver al kit de herramientas para VS Code.

 Note

Si el kit de herramientas para VS Code ya se ha autenticado con sus credenciales de IAM, VS Code mostrará el siguiente mensaje:

Algunas herramientas que ha estado usando no funcionan con el ID de creador de AWS. ¿Seguir usando el ID de creador de AWS en segundo plano mientras usa Credenciales AWS?

- Para continuar utilizando su ID de creador de AWS para autenticar el servicio CodeCatalyst y sus credenciales de IAM para autenticar sus otros servicios del kit de herramientas para VS Code, seleccione Sí, seguir usando el ID de creador de AWS con CodeCatalyst y usar las credenciales de IAM con otros servicios.
- Para continuar usando solo su ID de AWS Builder para autenticar todos los servicios del kit de herramientas para VS Code, elija No, cambiar todo para autenticarse con el ID de creador de AWS.

6. Cuando su ID de creador de AWS se haya conectado correctamente con el kit de herramientas para VS Code, el explorador de HERRAMIENTAS PARA DESARROLLADORES actualizará las funciones de CodeCatalyst y mostrará el estado AWSID de creador de conectado.

Uso de recursos de Amazon CodeCatalyst en VS Code

En las siguientes secciones encontrará una descripción general de las características de administración de recursos de Amazon CodeCatalyst que están disponibles con el kit de herramientas para VS Code.

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

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

Clonación de un repositorio

CodeCatalyst es un servicio basado en la nube que requiere estar conectado a la nube para trabajar en los proyectos de CodeCatalyst. Si prefiere trabajar en un proyecto de forma local, puede clonar los repositorios de CodeCatalyst en su máquina local y sincronizarlos con su proyecto de CodeCatalyst en línea la próxima vez que se conecte a la nube.

Para clonar un repositorio de su cuenta de CodeCatalyst con VS Code mediante el kit de herramientas de AWS, complete los siguientes pasos:

1. En el kit de herramientas para VS Code, expanda el explorador de HERRAMIENTAS PARA DESARROLLADORES.
2. Amplíe CodeCatalyst y seleccione Clonar repositorio.
3. En el cuadro de diálogo Seleccione un repositorio de CodeCatalyst, 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.

 **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?*.

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

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

 **Note**

Al seleccionar el entorno de desarrollo, este se abre para iniciar el proceso de conexión de VS Code con CodeCatalyst. Durante el proceso, VS Code muestra el progreso en una ventana de estado de CodeCatalyst. 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.

Creación de un entorno de desarrollo en CodeCatalyst

Para crear un nuevo entorno de desarrollo:

1. En el kit de herramientas para VS Code, expanda el explorador de HERRAMIENTAS PARA DESARROLLADORES.
2. Expanda CodeCatalyst y elija Crear entorno de desarrollo para abrir el cuadro de diálogo Cree un entorno de desarrollo de CodeCatalyst en VS Code.
3. En la sección Código fuente, elija una de las siguientes opciones:
 - Utilizar un repositorio de CodeCatalyst existente: crea un entorno de desarrollo a partir de un repositorio de CodeCatalyst existente. Debe seleccionar el proyecto y la ramificación de CodeCatalyst.
 - 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.

Note

Aunque no es obligatorio, le recomendamos que especifique un alias; de esa manera, se facilitará la organización y la búsqueda de sus entornos 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.

Note

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.

- Almacenamiento: seleccione Editar tamaño de almacenamiento para cambiar la cantidad de espacio de almacenamiento asignado a su sistema.

Note

El almacenamiento es la única configuración que no se puede cambiar después de haber creado su entorno de desarrollo.

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

Note

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.

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 origen se gestiona a nivel de proyecto en CodeCatalyst. Para obtener instrucciones y detalles adicionales sobre cómo conectar un repositorio de terceros a su entorno de desarrollo, consulte el tema [Vincular un repositorio de origen](#) en la Guía del usuario de Amazon CodeCatalyst.

Comandos de CodeCatalyst en VS Code

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

Para ver una lista de comandos asignados a CodeCatalyst desde la paleta de comandos:

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

Trabajo en entornos de desarrollo y VS Code

Puede lanzar un entorno de computación virtual de VS Code, conocido como entorno de desarrollo, desde el kit de herramientas para VS Code. Los entornos de desarrollo son entornos personalizables de desarrollo en la nube que puede copiar y compartir entre los distintos miembros del equipo de su espacio.

En las siguientes secciones se describe cómo configurar los entornos de desarrollo de forma local y remota, así como las funciones básicas a las que se puede acceder desde el kit de herramientas para VS Code cuando se lanza en un entorno de desarrollo. Para obtener información sobre los entornos de desarrollo, consulte [Entornos de desarrollo](#) en la guía de CodeCatalyst.

Temas

- [Configuración del entorno de desarrollo con archivos de desarrollo](#)
- [Conexión de las credenciales de AWS IAM con su entorno de desarrollo](#)
- [Uso del kit de herramientas para VS Code en entornos de desarrollo](#)

Configuración del entorno de desarrollo con archivos de desarrollo

Puede definir las herramientas y bibliotecas de aplicaciones específicas del proyecto que se instalan en su entorno de desarrollo en el `devfile` de su propio entorno de desarrollo. La especificación de `devfile` es un estándar abierto que se escribe en un documento YAML.

Puede editar su `devfile` de manera local y remotamente en VS Code. El proceso es similar al de crear o editar cualquier archivo compatible con VS Code. También puede editar el `Devfile`

directamente en CodeCatalyst desde el repositorio de origen de su proyecto. Para obtener instrucciones sobre cómo editar su archivo de desarrollo en CodeCatalyst, consulte la sección [Configuración del entorno de desarrollo](#) en la guía de CodeCatalyst.

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.

Para editar el `Devfile` en una instancia local de VS Code:

1. Desde el Explorador de VS Code, localice el archivo `devfile.yaml` y ábralo en el editor de VS Code.
2. Haga los cambios en el documento y guárdelo.
3. Confirme los cambios y envíelos al repositorio de origen.
4. La próxima vez que inicie su entorno de desarrollo, la configuración se actualizará para que coincida con las especificaciones definidas en su `Devfile`.

Conexión de las credenciales de AWS IAM con su entorno de desarrollo

Para acceder a todos los servicios del kit de herramientas para VS Code en un entorno de desarrollo, debe conectar las credenciales de AWS IAM con su entorno de desarrollo. Para conectar las credenciales de IAM con su entorno de desarrollo, complete los siguientes pasos:

1. En el EXPLORADOR del kit de herramientas de VS Code de su entorno de desarrollo, elija **Seleccionar credenciales de IAM** para ver los recursos para abrir el cuadro de diálogo **Cambiar conexión**.

2. Elija las credenciales de IAM que desee utilizar de la lista o seleccione **Añadir nueva conexión** para añadir nuevas credenciales.
3. Elija una de estas opciones y siga las instrucciones para crear una nueva conexión:
 - Usar un correo electrónico personal para registrarse e iniciar sesión con el ID de creador de AWS: crea un ID de creador de AWS nuevo para uso personal. Esta opción solo admite el uso de servicios exclusivos del ID de creador de AWS, como CodeCatalyst.
 - Conectarse mediante el Centro de identidad de AWS IAM: le permite iniciar sesión en el portal de acceso de IAM de su empresa. Esta opción solo está disponible a través de una empresa que tenga acceso al Centro de identidad de IAM. Los recursos y servicios de AWS a los que se puede acceder desde esta cuenta los administra la empresa propietaria de la cuenta.
 - Utilizar las credenciales de IAM: le permite trabajar con sus servicios y recursos AWS desde el kit de herramientas para VS Code. El uso únicamente de esta opción no permite utilizar servicios exclusivos del ID de creador de AWS, como CodeCatalyst.
4. Una vez que sus credenciales de IAM se hayan conectado correctamente al kit de herramientas para VS Code en su entorno de desarrollo, el EXPLORADOR de AWS se actualizará con sus servicios y recursos de AWS.

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 kit de herramientas de AWS y conectarse con su ID de creador de AWS.

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 y elija **Detener entorno de desarrollo**.
3. Cuando VS Code se lo solicite, confirme que desea detener su entorno de desarrollo. Cuando su entorno de desarrollo se haya detenido correctamente, 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:

1. En el kit de herramientas para VS Code, expanda el explorador de HERRAMIENTAS PARA DESARROLLADORES.
2. Expanda CodeCatalyst y elija Abrir configuración para abrir la vista de configuración del entorno de desarrollo que corresponda 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.

Note

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

Configuración de su entorno de desarrollo

Para abrir el devfile de su entorno de desarrollo y configurar su entorno de desarrollo actual, complete los siguientes pasos. Para obtener información más detallada sobre cómo editar los archivos de desarrollo, consulte los recursos que se enumeran en la sección [Configuración de su entorno de desarrollo con archivos de desarrollo](#) de esta guía.

1. En el kit de herramientas para VS Code, expanda el explorador de HERRAMIENTAS PARA DESARROLLADORES.
2. Expanda CodeCatalyst y seleccione Abrir archivo de desarrollo para abrir `devfile.yaml` en una nueva ventana del editor, dentro de su entorno de desarrollo actual.

Solución de problemas de Amazon CodeCatalyst y VS Code

Los siguientes temas abordan los problemas técnicos que pueden surgir 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 para los URI `vscode://` en su sistema. Sin este controlador, no puede acceder a todas las funciones de CodeCatalyst desde el kit de herramientas de AWS. Por ejemplo, se produce un error al iniciar un entorno de desarrollo desde VS Code Insiders. Esto se debe a que VS Code Insiders gestiona los URI `vscode-insiders://`, pero no los URI `vscode://`.

Permisos para Amazon CodeCatalyst

A continuación, se muestran los requisitos de los permisos de archivo para trabajar con CodeCatalyst desde el 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 AWS Management Console 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 utiliza la consola de Amazon ECR en <https://console.aws.amazon.com/ecr/repositories>.
- 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 expande el nodo ECR para ver la lista de repositorios de esa AWS región.
- 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 se enlace GitHub a AWS. Puede seleccionar las conexiones que están disponibles en la página de GitHubconexiones 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 icono 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, AppRunnerECR AccessRole, 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.

AWS Application Composer

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

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

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

Temas

- [Trabajar con AWS Application Composer desde el kit de herramientas](#)

Trabajar con AWS Application Composer desde el kit de herramientas

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

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

En las secciones siguientes se describe cómo acceder al servicio AWS Application Composer desde el AWS Toolkit for Visual Studio Code.

Acceso a AWS Application Composer desde el kit de herramientas

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

Acceso a AWS Application Composer desde una plantilla existente

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

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

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

Acceso a AWS Application 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, introduzca **AWS Application Composer** y seleccione AWSApplication Composer cuando aparezca en los resultados.
3. Elija el archivo de plantilla que desee abrir, AWS Application 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 AWS 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 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 AWS CloudFormation](#)
- [Cree una AWS CloudFormation plantilla utilizando el AWS Toolkit for Visual Studio Code](#)

Eliminación de una pila de AWS CloudFormation

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

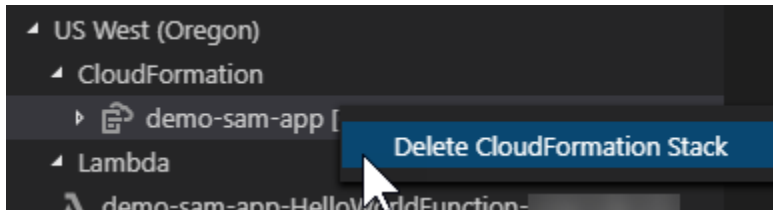
Requisitos previos

- 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 AWS CloudFormation. Si en AWS Explorer, en CloudFormation, aparece el mensaje “Error loading CloudFormation resources” (Error al cargar

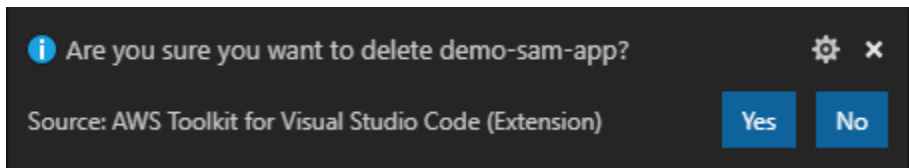
recursos de CloudFormation) o un mensaje similar, verifique los permisos adjuntos a esas credenciales. Los cambios que lleve a cabo en los permisos tardarán unos minutos en afectar al Explorador de AWS en VS Code.

Eliminar una pila de CloudFormation

1. En AWS Explorer, abra el menú contextual de la pila de AWS 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 haya creado [Autenticación y acceso](#) deben incluir el acceso de lectura/escritura 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

Amazon CloudWatch Logs 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](#)

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 registros es un grupo de flujos de registros que comparten la misma configuración de retención, monitorización y control de acceso. Puede definir grupos de registros 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

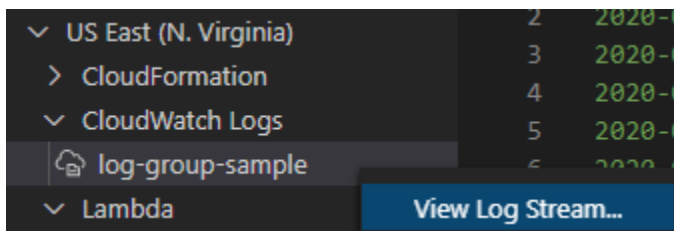
- [Visualización de grupos de registros y flujos de registro con el nodo de CloudWatch Logs](#)

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 icono 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 icono 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 icono **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.

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 AWS CLI 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.

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

Para poder acceder a un servicio de AWS, como Amazon ECR, debe proporcionar las credenciales. Esto permite al servicio determinar si tiene permisos para obtener acceso a sus recursos. Sin embargo, no es recomendable acceder a AWS con las credenciales de su cuenta raíz de AWS. En su lugar, utilice AWS Identity and Access Management (IAM) para crear un usuario de IAM y añada el usuario a un grupo de IAM con permisos administrativos. De este modo, podrá tener acceso a AWS mediante una URL especial y las credenciales del usuario de IAM.

Si se registró en AWS pero no creó un usuario de IAM para usted, puede crear uno con la consola de IAM.

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

Elegir una forma de administrar el administrador	Para	B	También puede
En IAM Identity Center (Recomendado)	Use credenciales a corto plazo para acceder a AWS. Esto se alinea con las prácticas	Siga las instrucciones en Introducción en la Guía del usuario de AWS IAM Identity Center.	Configure el acceso programático mediante Configuración de la AWS CLI para usar AWS IAM Identity Center en la Guía del usuario

Elegir una forma de administrar el administrador	Para	B	También puede
	recomendadas de seguridad. Para obtener información sobre las prácticas recomendadas, consulte Prácticas recomendadas de seguridad en IAM en la Guía del usuario de IAM.		de AWS Command Line Interface.
En IAM (No recomendado)	Use credenciales a largo plazo para acceder a AWS.	Siga las instrucciones en Creación del primer grupo de usuarios y usuario de administrador de IAM en la Guía del usuario de IAM.	Configure el acceso programático mediante Administración de 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 la sesión en la consola de AWS y, a continuación, utilice la siguiente URL. En la siguiente dirección URL, donde `su_id_de_cuenta_de_aws` es su número de cuenta de AWS sin los guiones (por ejemplo, si su número de cuenta de AWS es 1234-5678-9012, su ID de cuenta de AWS será 123456789012):

```
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 desea que la dirección URL de la página de inicio de sesión contenga el ID de su cuenta de AWS, puede 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, consulte [Su ID de cuenta de AWS y el 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/
```

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.

Instalar y configurar la versión 2 de AWS CLI

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

Temas

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

Uso del servicio Amazon Elastic Container Registry en VS Code

Puede acceder al servicio Amazon Elastic Container Registry (Amazon ECR) directamente desde el Explorador de AWS en VS Code y usarlo 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.

Secciones

- [Requisitos previos](#)
- [1. Creación de un Dockerfile](#)
- [2. Creación de la imagen a partir del Dockerfile](#)
- [3. Creación de un repositorio nuevo](#)
- [4. Inserción, extracción y eliminación de imágenes](#)

Requisitos previos

Para poder utilizar la característica del servicio Amazon ECR del kit de herramientas para VS Code, asegúrese de cumplir estos [requisitos previos](#).

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
SIZE			
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. Expanda el menú del Explorador de AWS.
3. Busque la región de AWS predeterminada que esté asociada con su cuenta de AWS. A continuación, selecciónela 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 completado el proceso, puede acceder al nuevo repositorio desde la sección ECR del menú del Explorador de AWS.

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, con el Explorador de AWS 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 una línea de comandos del sistema operativo que esté conectado a su instancia de AWS CLI.
2. Utilice el método `get-login-password` para autenticarse en su registro de 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 su instancia de AWS, inserte una imagen en su 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
SIZE			
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
```

Después de que la imagen etiquetada se haya cargado correctamente en el repositorio, estará visible en el menú del Explorador de AWS.

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 Explorador de AWS.

1. Desde el Explorador de AWS, 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.

Eliminación de una imagen mediante la AWS CLI

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

```
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 Explorador de AWS.

1. Desde el Explorador de AWS, 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.

Eliminación de un repositorio de Amazon ECR desde 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"  
    }  
  ]  
}
```

```
]
}
```

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](#).

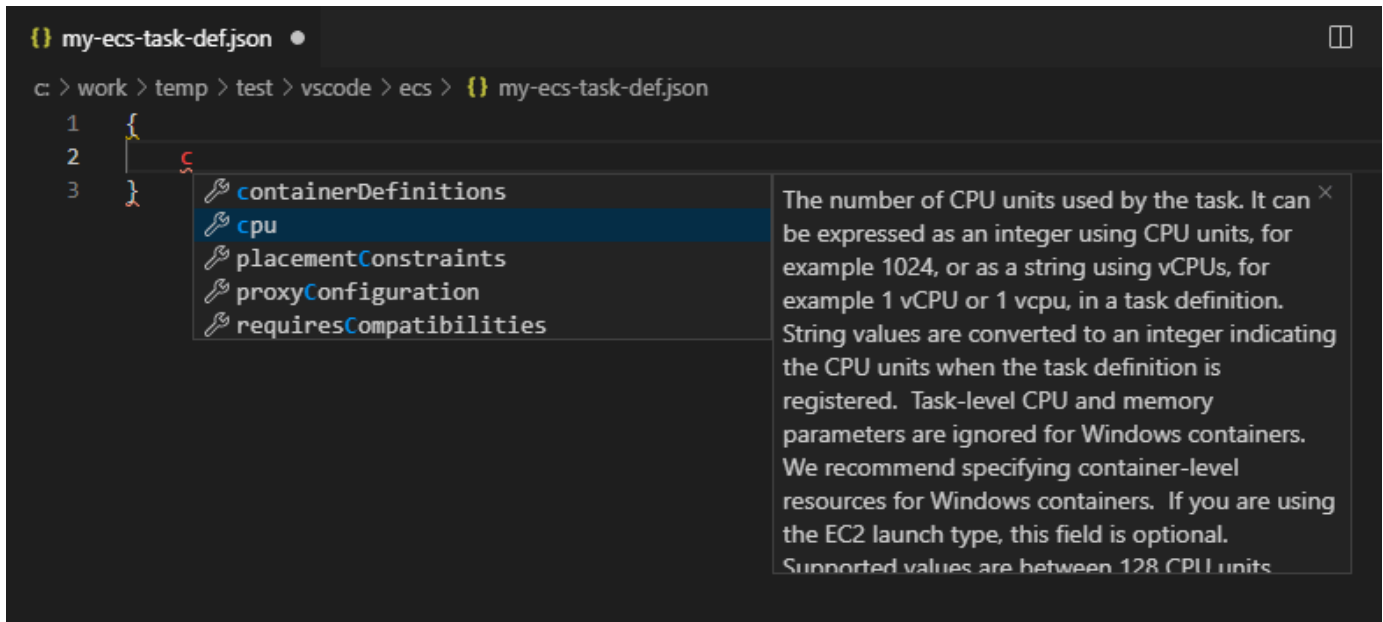
Uso de IntelliSense en archivos de definición de tareas de Amazon ECS

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.

⚠ Important

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 (Fargate) Amazon ECS Exec tiene diferentes requisitos de versión.

- Si utiliza Amazon EC2, debe usar una AMI optimizada para Amazon ECS que se publicó después del 20 de enero de 2021, con un agente versión 1.50.2 o superior. Encontrará información adicional en [AMI optimizadas para Amazon ECS](#) en la Guía para desarrolladores.
- 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 desee.

ℹ 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

- [Trabajar con Amazon EventBridge Schemas](#)

Trabajar con Amazon EventBridge Schemas

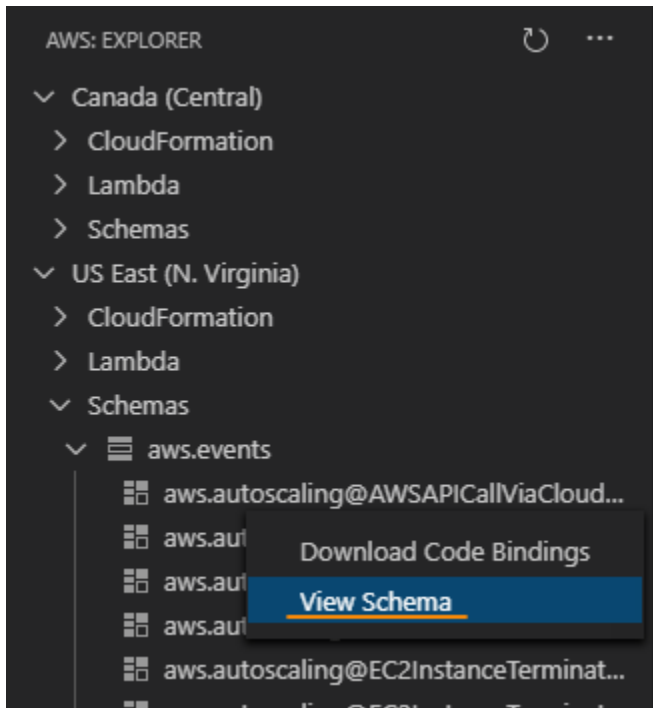
Puede usar el AWS Toolkit for Visual Studio Code código VS para realizar diversas operaciones en los [EventBridge esquemas de Amazon](#).

Requisitos previos

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

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 suministra 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).

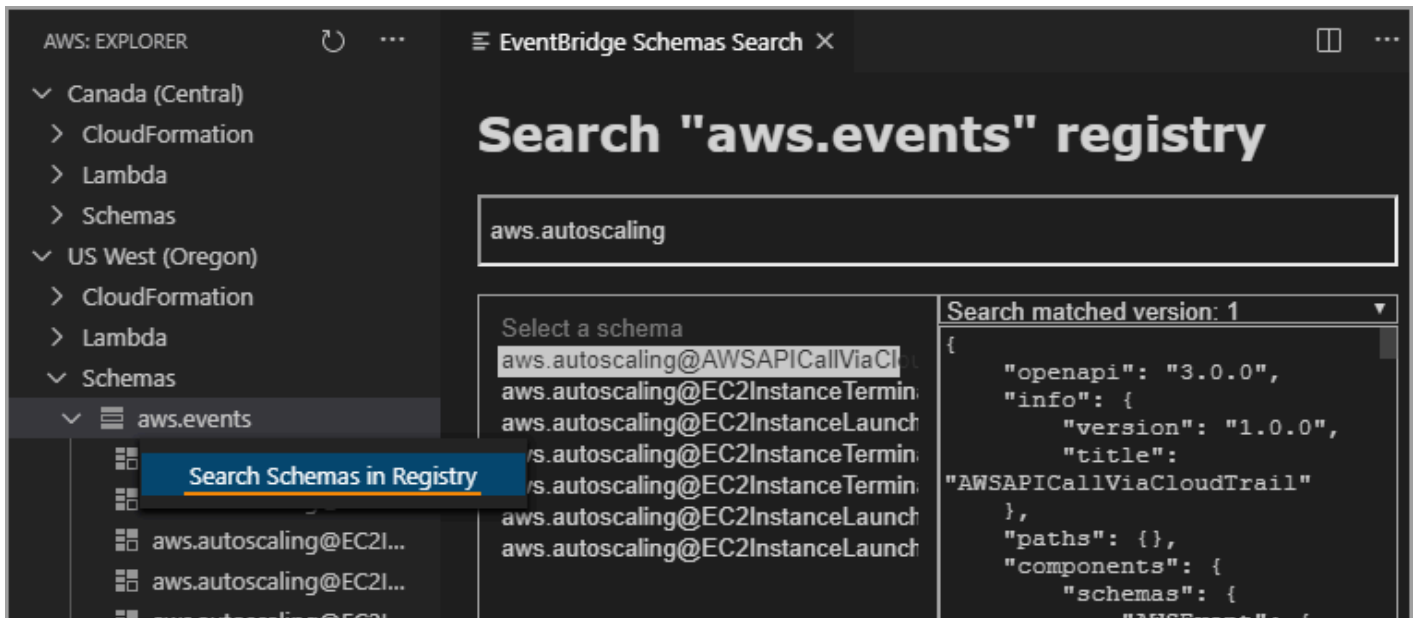


Búsqueda de un esquema disponible

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 de búsqueda de EventBridge esquemas, comience a escribir el título del esquema que desee 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 AWS CloudFormation plantillas, planes de Terraform y documentos de políticas JSON, utilizando el IAM Access Analyzer del. AWS Toolkit for Visual Studio Code

Las comprobaciones de políticas de IAM Access Analyzer incluyen la validación de políticas y las 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 prácticas recomendadas 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 de políticas.

También puedes realizar comprobaciones de políticas personalizadas para nuevos accesos, en función de tus estándares de seguridad. Se asocia un cargo a cada verificación de política personalizada para un nuevo acceso. Para obtener información detallada sobre los precios, consulte el sitio de [precios de AWS IAM Access Analyzer](#). Para obtener más información sobre las comprobaciones de políticas de IAM Access Analyzer, consulte el tema [Comprobaciones para validar las políticas](#) en la Guía del usuario. AWS Identity and Access Management

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

Temas

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

Uso de AWS IAM Access Analyzer

En las siguientes secciones se describe cómo realizar la validación de las políticas de IAM y las 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

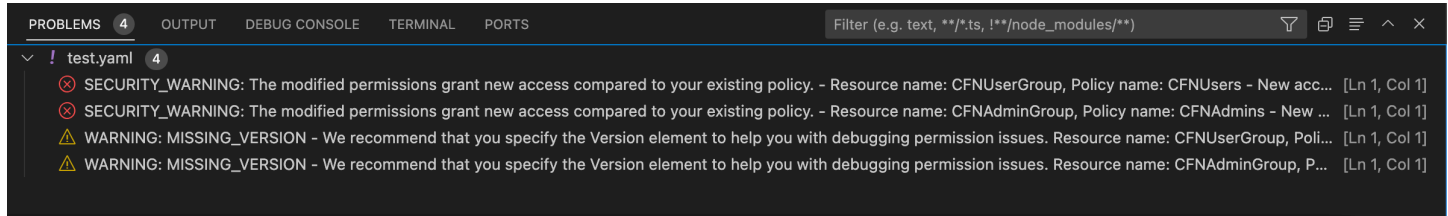
Se deben cumplir los siguientes requisitos previos para poder trabajar con las comprobaciones de políticas de IAM Access Analyzer desde el kit de herramientas.

- Instale Python 3.6 o posterior.
- Instale el validador de [políticas de IAM para Terraform AWS CloudFormation o el validador de políticas de IAM para Terraform](#) que requieren las herramientas CLI de Python y se especifican en la ventana 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 AWS CloudFormation plantillas, planes de Terraform y documentos de políticas de JSON utilizando el AWS Toolkit for Visual Studio Code. Los resultados de las comprobaciones se pueden ver en el panel de problemas de VS Code. La siguiente imagen muestra el panel de problemas de VS Code.



IAM Access Analyzer ofrece tres tipos de comprobaciones:

- Política de validación
- CheckAccessNotGranted
- CheckNoNewAccess

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

Note

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

Ejecutando la política de validación

La comprobación de la política de validación, también conocida como validación de políticas, valida la política según la gramática y las AWS 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 de IAM Access Analyzer, abra la paleta de comandos de VS Code pulsando **CTRL+Shift+P**, buscando y haciendo clic para **IAM Policy Checks** abrir el panel de comprobaciones de políticas de IAM en el editor de VS Code.
3. En el panel de comprobaciones de políticas de IAM, seleccione el tipo de documento en el menú desplegable.
4. En la sección Validar políticas, pulse el botón Ejecutar la validación de políticas para ejecutar la comprobación de validación de políticas.
5. En el panel de problemas de VS Code, revise los resultados de la verificación de su política.
6. Actualice su política y repita este procedimiento y vuelva a ejecutar la comprobación de validación de la política hasta que la comprobación de la política deje de mostrar advertencias o errores de seguridad.

En ejecución CheckAccessNotGranted

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

Note

Se asocia un cargo a cada comprobación de política personalizada. Para obtener más información sobre los precios de las verificaciones de políticas personalizadas, consulte la guía de precios de [AWS IAM Access Analyzer](#).

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 de IAM Access Analyzer, abra la paleta de comandos de VS Code pulsando **CTRL+Shift+P**, buscando y haciendo clic para **IAM Policy Checks** abrir el panel de comprobaciones de políticas de IAM en el editor de VS Code.
3. En el panel de comprobaciones de políticas de IAM, seleccione el tipo de documento en el menú desplegable.
4. En la sección Verificaciones de políticas personalizadas, seleccione CheckAccessNotGranted.

5. En el campo de entrada de texto, introduce una lista de acciones separadas por comas. Como alternativa, puede proporcionar una ruta de archivo que apunte a una lista de acciones.


 Note

La ruta del archivo normalmente la proporciona el administrador o el equipo de seguridad y puede ser una ruta de archivo del sistema o un URI de bucket de Amazon S3. Para usar un URI de bucket de Amazon S3, su rol actual debe tener acceso al bucket de Amazon S3.

6. Pulse el botón Ejecutar una verificación de política personalizada.
7. En el panel de problemas de VS Code, revise los resultados de la verificación de su política. Las verificaciones de políticas personalizadas arrojan un FAIL resultado PASS o.
8. Actualice su política y repita este procedimiento y vuelva a ejecutar la CheckAccessNotGranted comprobación hasta que se devuelvaPASS.

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.

 Note

Se asocia un cargo a cada comprobación de política personalizada. Para obtener más información sobre los precios de las verificaciones de políticas personalizadas, consulte la [AWS guía de precios de IAM Access Analyzer](#).

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 de IAM Access Analyzer, abra la paleta de comandos de VS Code pulsando **CTRL+Shift+P**, buscando y haciendo clic para **IAM Policy Checks** abrir el panel de comprobaciones de políticas de IAM en el editor de VS Code.
3. En el panel de comprobaciones de políticas de IAM, seleccione el tipo de documento en el menú desplegable.
4. En la sección Verificaciones de políticas personalizadas, seleccione CheckNoNewAccess.

5. Introduzca un documento de política de JSON de referencia. Como alternativa, puede proporcionar una ruta de archivo que haga referencia a un documento de política de JSON.

 Note

La ruta del archivo normalmente la proporciona el administrador o el equipo de seguridad y puede ser una ruta de archivo del sistema o un URI de bucket de Amazon S3. Para usar un URI de bucket de Amazon S3, su rol actual debe tener acceso al bucket de Amazon S3.

6. Seleccione el tipo de política de referencia que coincida con el tipo de su documento de referencia.
7. Pulse el botón Ejecutar una verificación de política personalizada.
8. En el panel de problemas de VS Code, revise los resultados de la verificación de su póliza. Las verificaciones de políticas personalizadas arrojan un FAIL resultado PASS o.
9. Actualice su política y repita este procedimiento y vuelva a ejecutar la CheckNoNewAccess comprobación hasta que se devuelvaPASS.

Uso de AWS IoT en 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.

Requisitos previos de AWS IoT

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 un dispositivo concreto o de una entidad lógica. 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 funciones 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 icono 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.

Políticas de AWS IoT

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](#)

Para crear 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. Expanda la subsección Políticas (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 Políticas (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.

Trabajo con funciones de AWS Lambda

El AWS Toolkit for Visual Studio Code ofrece soporte para las funciones de [AWS Lambda](#). Con el kit de herramientas para VS Code, puede crear código para funciones de Lambda que forman parte de [aplicaciones sin servidor](#). Además, puede invocar funciones de Lambda localmente o en AWS.

Lambda es un servicio informático completamente administrado que ejecuta código como respuesta a eventos generados por código personalizado o procedentes de varios servicios de AWS, como Amazon Simple Storage Service (Amazon S3), Amazon DynamoDB, Amazon Kinesis, Amazon Simple Notification Service (Amazon SNS) y Amazon Cognito.

Temas

- [Interacción con funciones de Lambda remotas](#)

Interacción con funciones de Lambda remotas

Con el kit de herramientas para VS Code, puede interactuar con las funciones de [AWS Lambda](#) de varias maneras, como se describe más adelante en este tema.

Para obtener más información acerca de Lambda, consulte la [Guía para desarrolladores de AWS Lambda](#).

Note

Si ya ha creado funciones de Lambda mediante la AWS Management Console o de alguna otra manera, puede invocarlas desde Toolkit. Para crear una nueva función (mediante VS Code) que pueda implementar en AWS Lambda, antes debe [crear una aplicación sin servidor](#).

Temas

- [Requisitos previos](#)
- [Invocar una función Lambda](#)
- [Eliminación de una función Lambda](#)
- [Importar una función Lambda](#)
- [Carga de una función de Lambda](#)

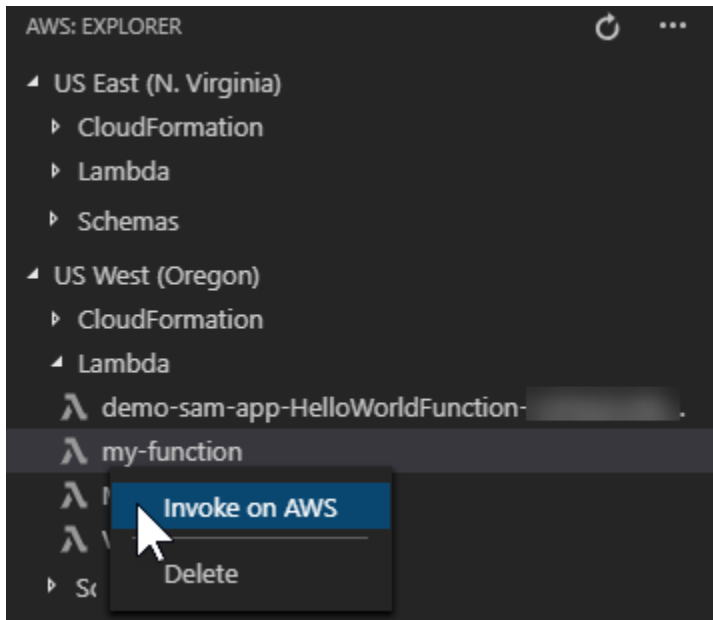
Requisitos previos

- 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 AWS Lambda. Si en AWS Explorer, en Lambda, aparece un mensaje del tipo “Error loading Lambda resources” (Error al cargar recursos de Lambda), 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.

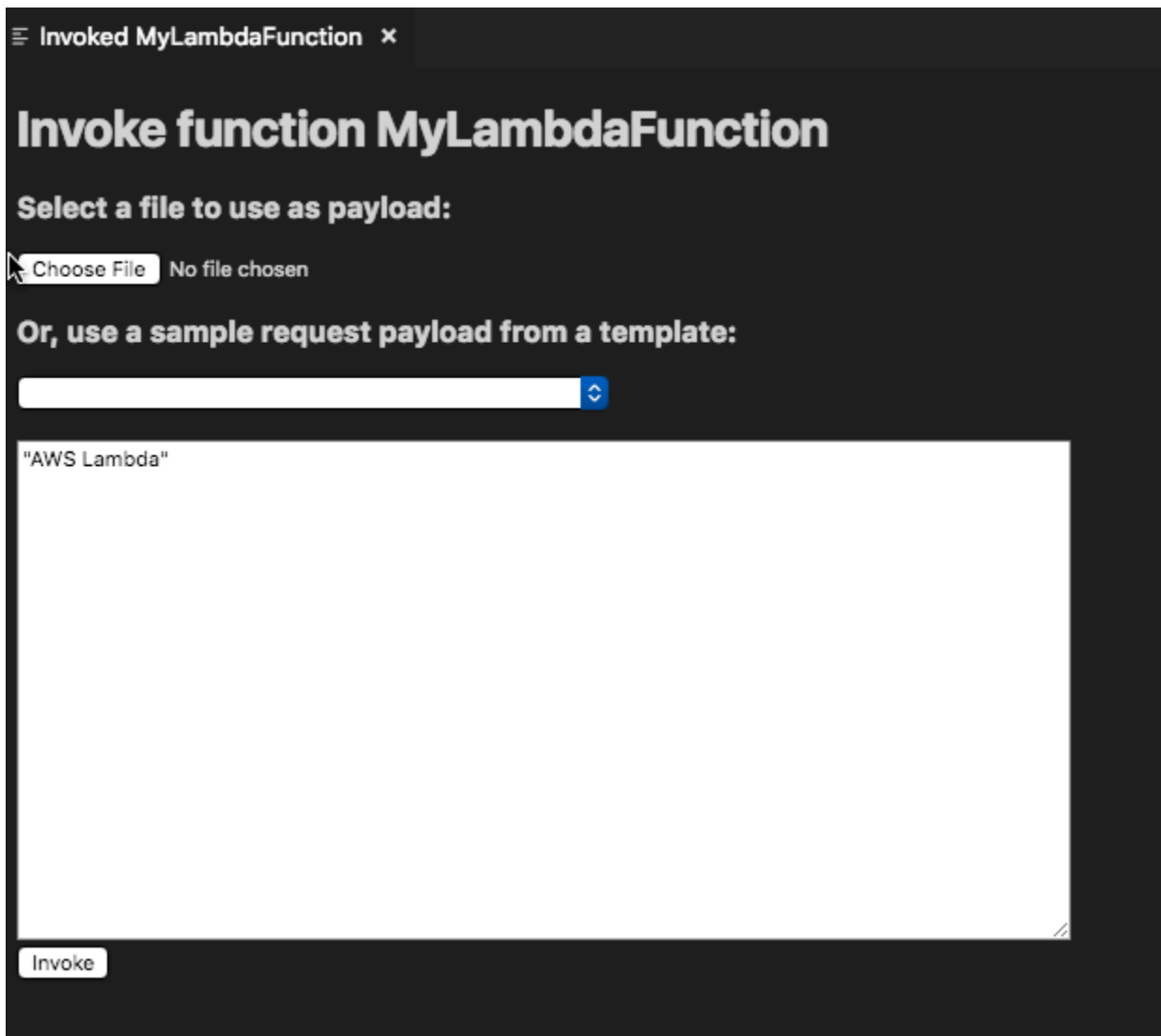
Invocar una función Lambda

Puede invocar una función de Lambda en AWS mediante el kit de herramientas para VS Code.

1. En AWS Explorer, elija el nombre de la función de Lambda que desea invocar y, a continuación, abra su menú contextual.



2. Elija Invoke on AWS (Invocar en).
3. En la ventana de invocación que se abre, escriba la entrada que necesita la función de Lambda. La función de Lambda podría, por ejemplo, requerir una cadena como entrada, como se muestra en el cuadro de texto.



La salida de la función de Lambda aparecerá igual que en cualquier otro proyecto en el que se use VS Code.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  AWS Lambda
Loading response...
Invocation result for arn:aws:lambda:us-west-2:282147211633:function:MyLambdaFunction
Logs:
START RequestId: 04c5f09a-508f-4251-bc6d-663023153d7c Version: $LATEST
Input: AWS LambdaEND RequestId: 04c5f09a-508f-4251-bc6d-663023153d7c
REPORT RequestId: 04c5f09a-508f-4251-bc6d-663023153d7c Duration: 253.98 ms Billed Duration: 300 ms Memory Size: 128 MB
Max Memory Used: 86 MB

Payload:
"Hello, AWS Lambda!"
```

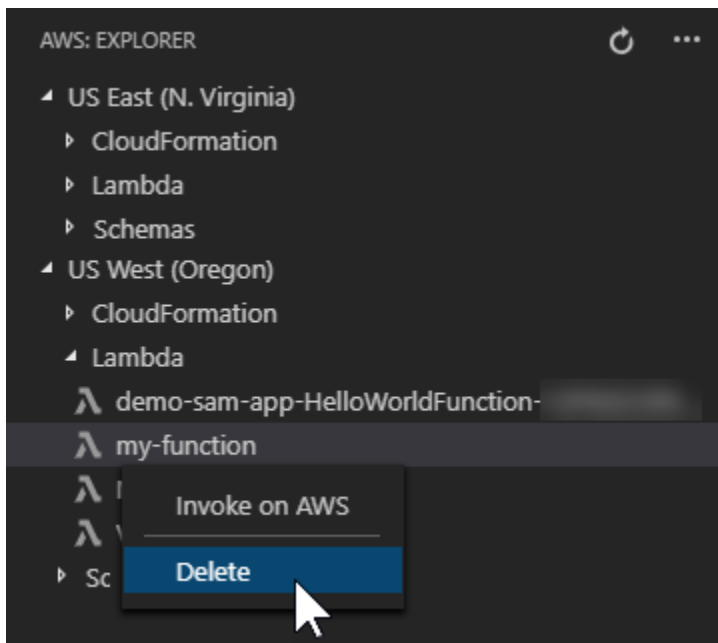
Eliminación de una función Lambda

También puede eliminar una función de Lambda utilizando el mismo menú contextual.

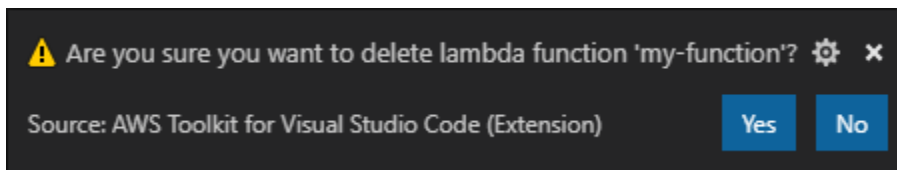
Warning

No utilice este procedimiento para eliminar funciones de Lambda asociadas a [AWS CloudFormation](#) (por ejemplo, la función de Lambda que se creó al [crear una aplicación sin servidor](#) anteriormente en esta guía). Estas funciones deben eliminarse a través de la pila de AWS CloudFormation.

1. En AWS Explorer, elija el nombre de la función de Lambda que desea eliminar y, a continuación, abra su menú contextual.



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



Una vez eliminada la función, deja de aparecer en la lista del Explorador de AWS.

Importar una función Lambda

Puede importar código desde una función Lambda remota a su espacio de trabajo de VS Code para editarlo y depurarlo.

Note

El kit de herramientas solo admite la importación de funciones de Lambda mediante los tiempos de ejecución de Node.js y Python compatibles.

1. Desde el Explorador de AWS, elija el nombre de la función de Lambda que desea importar y, a continuación, abra su menú contextual.
2. Seleccionar Importar....
3. Elija una carpeta a la que quiera importar el código de Lambda. Las carpetas situadas fuera del espacio de trabajo actual se añadirán a él.
4. Tras la descarga, el kit de herramientas añade el código a su espacio de trabajo y abre el archivo que contiene el código del controlador de Lambda. El kit de herramientas también crea una configuración de inicio, que aparece en el panel de ejecución de VS Code para que pueda ejecutar y depurar localmente la función de Lambda mediante AWS Serverless Application Model. Para obtener más información acerca del uso de AWS SAM, consulte [the section called “Ejecutar y depurar una aplicación sin servidor desde una plantilla \(local\)”](#).

Carga de una función de Lambda

Puede actualizar las funciones de Lambda existentes con código local. Al actualizar el código de esta manera, no se utiliza AWS SAM CLI para la implementación ni se crea una pila de AWS CloudFormation. Esta funcionalidad puede cargar una función de Lambda con cualquier tiempo de ejecución admitido por Lambda.

Warning

El kit de herramientas no puede comprobar si el código funciona. Asegúrese de que el código funcione antes de actualizar las funciones de Lambda de producción.

1. Desde el Explorador de AWS, elija el nombre de la función de Lambda que desea importar y, a continuación, abra su menú contextual.
2. Elija Upload Lambda... (Cargar Lambda...)
3. Elija entre las tres opciones para cargar su función de Lambda. Las opciones son:

Cargar un archivo .zip prediseñado

- Elija Archivo ZIP en el menú de selección rápida.
- Elija un archivo.zip de su sistema de archivos y confirme la carga con el cuadro de diálogo modal. Con esta acción se carga el archivo .zip tal cual y se actualiza inmediatamente la función de Lambda después de la implementación.

Cargar un directorio tal cual

- Elija Directorio en el menú de selección rápida.
- Elija un directorio del sistema de archivos.
- Seleccione No cuando se le solicite crear el directorio y, a continuación, confirme la carga con el cuadro de diálogo modal. Con esta acción se carga el directorio y se actualiza inmediatamente la función de Lambda después de la implementación.

Crear y cargar un directorio

Note

Esto requiere la CLI de AWS SAM.

- Elija Directorio en el menú de selección rápida.
- Elija un directorio del sistema de archivos.
- Seleccione Sí cuando se le solicite crear el directorio y, a continuación, confirme la carga con el cuadro de diálogo modal. Con esta acción se compila el código en el directorio mediante el comando `sam build` de AWS SAM CLI y se actualiza inmediatamente la función de Lambda después de la implementación.

Note

El kit de herramientas le avisará si no puede detectar un controlador coincidente antes de subirlo. Si desea cambiar el controlador vinculado a la función de Lambda, puede hacerlo a través la AWS Management Console o de la AWS CLI.

Amazon Redshift en el kit de herramientas para VS Code

Amazon Redshift es un servicio de data warehouse 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 ComComience 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 desee 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 icono 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 icono 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 Crear carpeta.
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 eliminen, 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 icono 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 icono 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 icono 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, muéstrelo 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 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.

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](#)
- [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 desde el AWS Toolkit for Visual Studio Code mediante AWS Serverless Application Model (AWS SAM) y pilas de AWS CloudFormation.

Requisitos previos

Para poder crear una Aplicación sin servidor de AWS o trabajar con ella, deben cumplirse los siguientes requisitos previos.

Note

Es posible que las siguientes operaciones requieran que cierre o reinicie VS Code para que se completen los cambios.

- Instalación de la interfaz de línea de comandos (CLI) de AWS SAM Para obtener instrucciones e información adicional acerca de cómo instalar la CLI de AWS SAM, consulte el tema sobre [cómo instalar la CLI de AWS SAM](#) en la Guía del usuario de AWS Serverless Application Model.
- Identifique la región de AWS predeterminada en el archivo de configuración de AWS. 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 más información sobre cómo configurar la cadena de herramientas desde el AWS Toolkit for Visual Studio Code, consulte el tema sobre cómo [configurar la cadena de herramientas](#) en esta Guía del usuario.
- Instale la [extensión de compatibilidad con el lenguaje YAML](#) desde el Marketplace de VS Code. Es necesaria para poder acceder a la característica CodeLens de los archivos de plantilla de AWS SAM. Para obtener más información sobre CodeLens, consulte la sección [CodeLens](#) en la documentación de VS Code.

Permisos de IAM para aplicaciones sin servidor

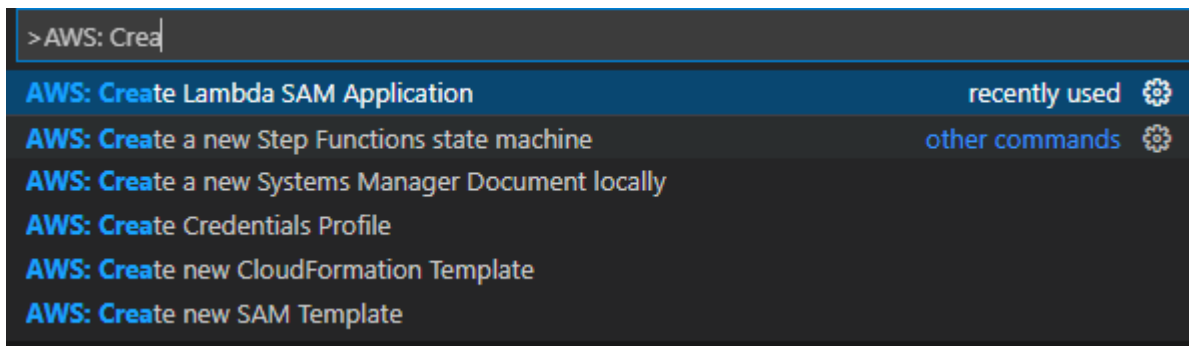
En el kit de herramientas para VS Code, debe tener un perfil de credenciales que contenga los permisos de AWS Identity and Access Management (IAM) necesarios para implementar y ejecutar aplicaciones sin servidor. Debe tener el acceso de lectura/escritura adecuado a los siguientes servicios: AWS CloudFormation, IAM, Lambda, Amazon API Gateway, Amazon 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)

En este ejemplo se muestra cómo crear una aplicación sin servidor con el kit de herramientas para 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 de 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 CLI de AWS SAM no está instalada, aparecerá 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. Seleccione el tiempo de ejecución de la aplicación de AWS SAM.

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

Elige 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 basado en ARM64 en lugar del entorno predeterminado basado en x86_64 está disponible para los siguientes tiempos de ejecución:

- nodejs12.x (ZIP e imagen)
- nodejs12.x (ZIP e imagen)
- python3.8 (ZIP e imagen)
- python3.8 (ZIP e imagen)
- python3.8 (ZIP e imagen)
- python3.8 (ZIP e imagen)
- java8.al2 con Gradle (ZIP e imagen)
- java8.al2 con Maven (solo ZIP)
- java8.al2 con Gradle (ZIP e imagen)
- java8.al2 con Maven (solo ZIP)

Important

Debe instalar la versión de AWS CLI 1.33.0 o posterior para permitir que las aplicaciones se ejecuten en entornos basados en 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.

Comience a configurar el comportamiento de depuración mediante la característica [CodeLens](#) de VS Code para identificar una función de Lambda apta. CodeLens permite interacciones basadas en el contenido con el código fuente. Para obtener información sobre cómo garantizar el acceso a la característica CodeLens, consulte la sección [Requisitos previos](#) que aparece con anterioridad en este tema.

Note

En este ejemplo, va a depurar una aplicación que usa 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
- Java: 8, 8.al2, 11
- Go: 1.x

Su elección de lenguaje también afecta a la manera en que CodeLens detecta los controladores de Lambda válidos. 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. En la carpeta de la aplicación (por ejemplo, my-sample-app), abra el archivo `template.yaml`.

Note

Si utiliza una plantilla con un nombre que no sea `template.yaml`, el indicador CodeLens no está 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 indicador CodeLens de este recurso, seleccione Agregar 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 es `"template"` para que la plantilla de AWS SAM sea el punto de entrada para 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 plantilla de AWS SAM. En este caso, es `HelloWorldFunction` de tipo `AWS::Serverless::Function`.

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.

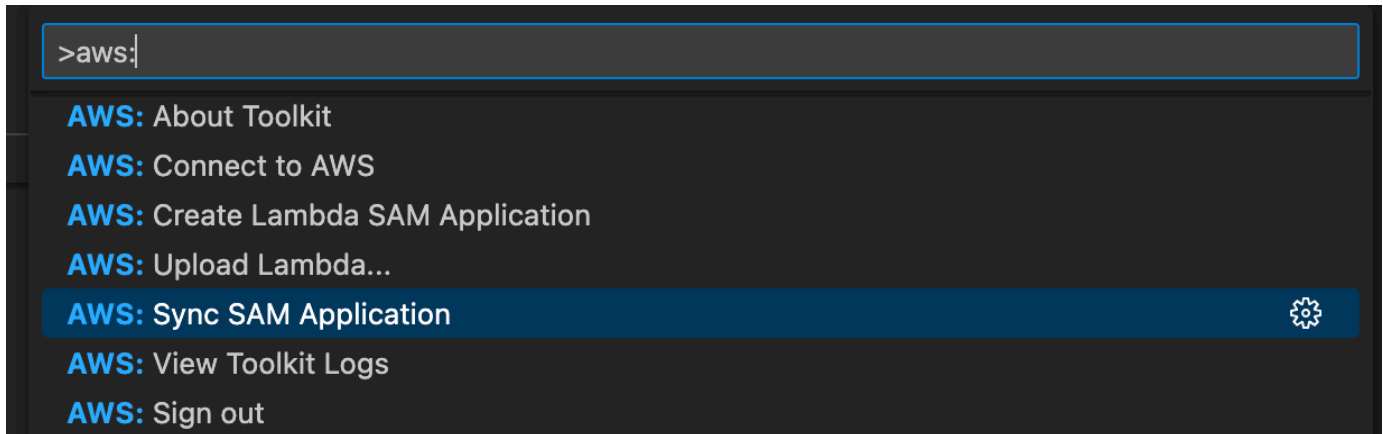
Cuando se inician las sesiones de depuración, el panel DEBUG CONSOLE (CONSOLA DE DEPURACIÓN) muestra la salida de la depuración y los valores devueltos por la función de Lambda. (Al depurar aplicaciones de AWS SAM, el kit de herramientas de AWS se selecciona como el canal de Salida en el panel Salida).

Sincronización de aplicaciones de AWS SAM

El AWS Toolkit for Visual Studio Code ejecuta el comando `sam sync` de la CLI de AWS SAM para implementar las aplicaciones sin servidor en la Nube de AWS. Para obtener más información sobre la sincronización de AWS SAM, consulte el tema [Referencia de los comandos de la CLI de AWS SAM](#) en la Guía del usuario de AWS Serverless Application Model.

El siguiente procedimiento describe cómo implementar sus aplicaciones sin servidor en la Nube de AWS con `sam sync` desde el kit de herramientas para 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 AWSy seleccione Sincronizar aplicación de SAM para empezar a configurar la sincronización.



3. Elija la región de AWS con la que desea sincronizar la 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:

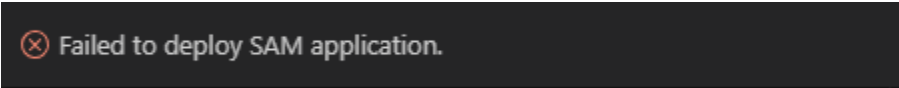
- El bucket debe estar en la región en la que esté realizando la sincronización.
- El nombre del bucket de Amazon S3 debe ser único globalmente para todos los nombres de buckets existentes en Amazon S3.

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.

A screenshot of a terminal window showing an error message: "Failed to deploy SAM application." The message is displayed in white text on a dark background.**Note**

Para obtener más información sobre la sincronización, puede acceder a los registros del AWS Toolkit for Visual Studio Code desde la paleta de comandos.

Para acceder a los registros del AWS Toolkit for Visual Studio Code desde la paleta de comandos, expanda Ver, elija Paleta de comandos, a continuación, busque **AWS: View AWS Toolkits Logs** y selecciónelo 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 [Interacción con funciones de Lambda remotas](#) de esta Guía del usuario.

Eliminación de una aplicación sin servidor de la nube de Nube de AWS

Eliminar una aplicación sin servidor implica eliminar la pila de AWS CloudFormation que ha implementado previamente en la nube de AWS. Tenga en cuenta que este procedimiento no elimina el directorio de la aplicación del anfitrión local.

1. Abra la [AWS Explorador](#).
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 AWS CloudFormation.

3. Abra el menú contextual (clic con el botón derecho) del nombre de la pila de AWS CloudFormation que corresponde a la aplicación sin servidor que desea eliminar y, a continuación, seleccione Eliminar pila de AWS CloudFormation.
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 AWS CloudFormation en el Explorador de AWS.

Ejecución y depuración de funciones de Lambda directamente desde el código

Al probar la aplicación de AWS SAM, puede optar por ejecutar y depurar solo la función de Lambda y excluir otros recursos definidos por la plantilla de AWS SAM. Este enfoque implica el uso de la característica [CodeLens](#) para identificar los controladores de funciones de Lambda en el código fuente que se pueden invocar directamente.

Los controladores de Lambda que detecta CodeLens dependen del lenguaje y el tiempo de ejecución que esté utilizando para su aplicación.

Lenguaje/tiempo de ejecución	Criterios para identificar las funciones de Lambda mediante los indicadores de 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>

Lenguaje/tiempo de ejecución	Criterios para identificar las funciones de Lambda mediante los indicadores de CodeLens
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)	<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. <p>La extensión ms-python.python (o cualquier extensión que proporcione símbolos de lenguaje para Python) está instalada y habilitada.</p>

Lenguaje/tiempo de ejecución	Criterios para identificar las funciones de Lambda mediante los indicadores de 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 motor de ejecución de Java que utilice.</p> <p>La extensión vscjava.vscode-java-debug (o cualquier extensión que proporcione un depurador de Java) está instalada y habilitada.</p>

Lenguaje/tiempo de ejecución	Criterios para identificar las funciones de Lambda mediante los indicadores de 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 golang.go 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 la aplicación (por ejemplo, `my-sample-app`), expanda la carpeta de la función (en este caso, `hello-world`) y abra el archivo `app.js`.
3. En el indicador de CodeLens que identifica un controlador de función de de Lambda válido, elija `Add Debug Configuration`.
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 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.

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 aplicaciones de AWS SAM, el kit de herramientas de AWS se selecciona como el canal de Salida en el panel 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-invoke",
  "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.

```

"lambda": {
  "runtime": "nodejs12.x",
  "event": {
    "json": {}
  }
}

```

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
type	Especifica qué extensión administra la configuración de lanzamiento. Siempre se establece en <code>aws-sam</code> para usar AWS SAMCLI 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 (<code>aws-sam</code>). Siempre se establece en <code>direct-invoke</code> para lanzar la función de Lambda.

Propiedad	Descripción
<code>invokeTarget</code>	<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>: se establece en <code>template</code>.• <code>templatePath</code> : ruta al archivo de plantilla de AWS SAM.• <code>logicalId</code> : nombre del recurso de AWS::Lambda::Function o AWS::Serverless::Function 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> <div data-bbox="592 520 1507 1709" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"> <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 data-bbox="690 926 1474 1360">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 data-bbox="690 1518 1474 1675">"environmentVariables": { "SAMPLE1": "My sample 1 value" }</pre> </div>
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 AWS 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
<code>skipNewImageCheck</code>	Especifica si el comando debe omitir la extracción de la última imagen de Docker para el tiempo de ejecución de Lambda.	<code>false</code>
<code>template</code>	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.	<code>"parameters": {}</code>

Propiedades de conexión de AWS ("**aws**")

Propiedad	Descripción	Valor predeterminado
<code>credentials</code>	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.
<code>region</code>	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-invoke",
      "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-invoke",
      "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 `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\\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 ofrece visibilidad y control de su infraestructura en AWS. Systems Manager proporciona una interfaz de usuario unificada para que pueda ver los datos operativos de varios servicios de AWS y automatizar tareas operativas en todos sus recursos de AWS.

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 Automatización 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 de 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:

```
{
  "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 al archivo y guardarlo en VS Code, pero el nombre del archivo no es visible para 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 en AWS antes de poder ejecutarlo.

Apertura de un documento de automatización de Systems Manager existente

Utilice el Explorador de AWS para buscar documentos de Automatización de Systems Manager existentes. 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. Desde el Explorador de AWS, en Systems Manager, elija el icono de descarga del documento que desee abrir y, a continuación, seleccione 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 en 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 De mi propiedad de los documentos de Systems Manager en el Explorador de AWS. Puede ser propietario de documentos de automatización que ya existan en AWS y puede ser propietario de documentos nuevos o actualizados que haya publicado anteriormente en AWS desde VS Code.

Cuando abre un documento de Automatización para editarlo en VS Code, puede hacer más con él que en la AWS Management Console. 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 AWS Explorer, vaya al documento en el que quiere establecer la versión predeterminada, abra el menú contextual (haga clic con el botón derecho) del documento y elija Set default version (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 el 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 icono de carga.
4. En el cuadro de diálogo del flujo de trabajo de publicación, elija la región de AWS en la que desee publicar el documento de Automatización.
5. Si va a publicar un documento nuevo, elija Creación rápida. O elija Actualización rápida para actualizar un documento de Automatización existente en esa región de AWS.
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. Desde el Explorador de AWS para Systems Manager, abra el menú contextual (clic con el botón derecho) del documento que quiera eliminar y elija Eliminar documento.

Ejecución de un documento de Automatización de Systems Manager

Una vez publicado el documento de Automatización en AWS, puede ejecutarlo para llevar a cabo tareas en su nombre en su cuenta de AWS. Para ejecutar el documento de automatización, utilice la AWS Management Console, las API de Systems Manager, la AWS CLI o AWS Tools for 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 quiere usar uno de los AWS SDK con las API de Systems Manager para ejecutar el documento de automatización, consulte las [referencias de AWS SDK](#).

Note

La ejecución de un documento de Automatización puede crear nuevos recursos en AWS y puede incurrir en costos de facturación. Le recomendamos encarecidamente que comprenda lo que creará su documento de Automatización en su cuenta antes de iniciarlo.

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 AWS Management Console.

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 el perfil de las credenciales de AWS tiene 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](#).

Publiqué mi documento de Automatización en AWS, pero no lo veo en la AWS Management Console.

Asegúrese de que ha publicado el documento en la misma región de AWS que está explorando en la AWS Management Console.

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. Puede identificar los recursos de AWS que ha creado desde la [consola de Administración de facturación y costos de AWS](#), explorar los cargos y elegir qué recursos quiere eliminar.

Trabajando con AWS Step Functions

El AWS Toolkit for Visual Studio Code (Código VS) brinda soporte para [AWS Step Functions](#). Con el kit de herramientas para VS Code, puede crear, actualizar y ejecutar máquinas de estado de Step Functions.

Temas

- [Trabajando con AWS Step Functions](#)

Trabajando con AWS Step Functions

Puede usar el AWS Toolkit for Visual Studio Code (código VS) para realizar diversas operaciones con [máquinas de estado](#).

Temas

- [Requisitos previos](#)
- [Trabajar con máquinas de estado en VS Code](#)
- [Plantillas de máquina de estado](#)
- [Visualización de gráfico de máquina de estado](#)
- [Fragmentos de código](#)
- [Finalización y validación de código](#)

Requisitos previos

- Asegúrese de que el sistema cumple los requisitos previos especificados en [Instalación del kit de herramientas para VS Code](#) y después instale el kit de herramientas.
- Asegúrese de que ha configurado sus credenciales antes de abrir el Explorador de AWS .

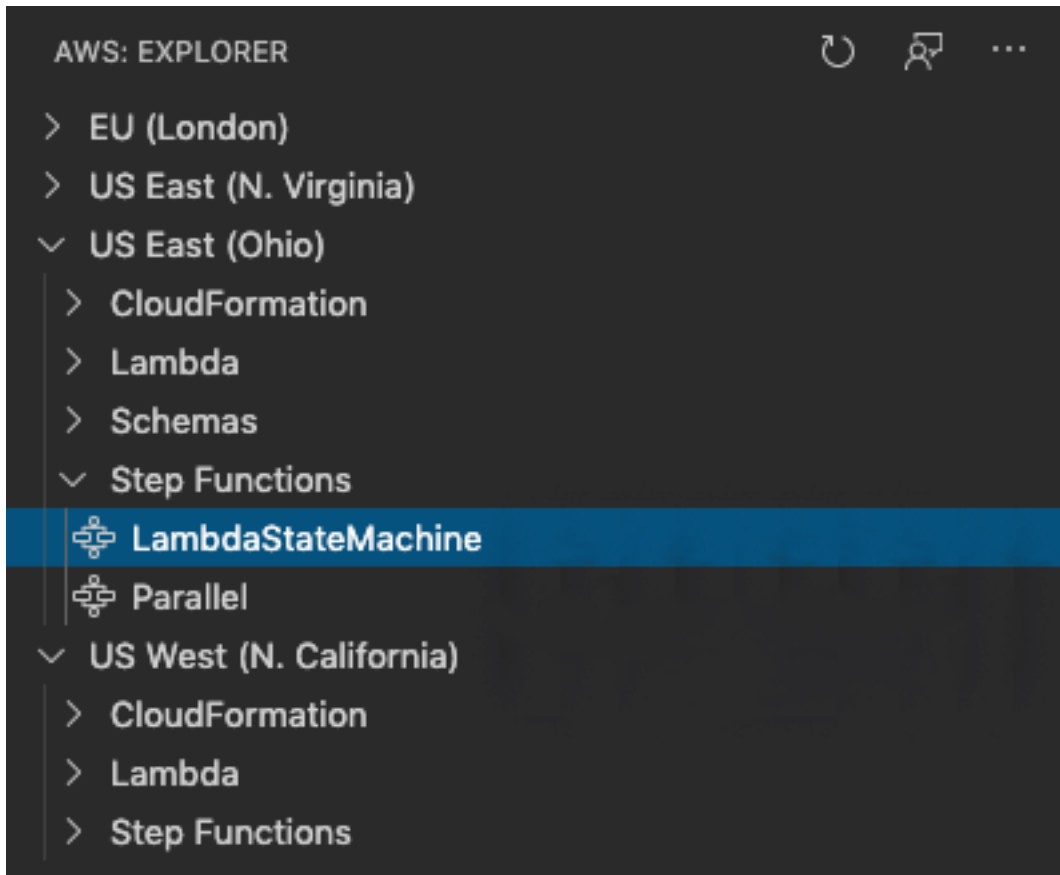
Trabajar con máquinas de estado en VS Code

Puede utilizar VS Code para interactuar con máquinas de estado remotas y desarrollar máquinas de estado localmente en formato JSON o YAML. Puede crear o actualizar máquinas de estado, mostrar las máquinas de estado existentes, ejecutarlas y descargarlas. VS Code también le permite crear nuevas máquinas de estado a partir de plantillas, ver una visualización de su máquina de estado y proporciona fragmentos de código, finalización de código y validación de código.

Mostrar las máquinas de estado existentes

Si ya ha creado máquinas de estado, puede ver una lista de ellas:

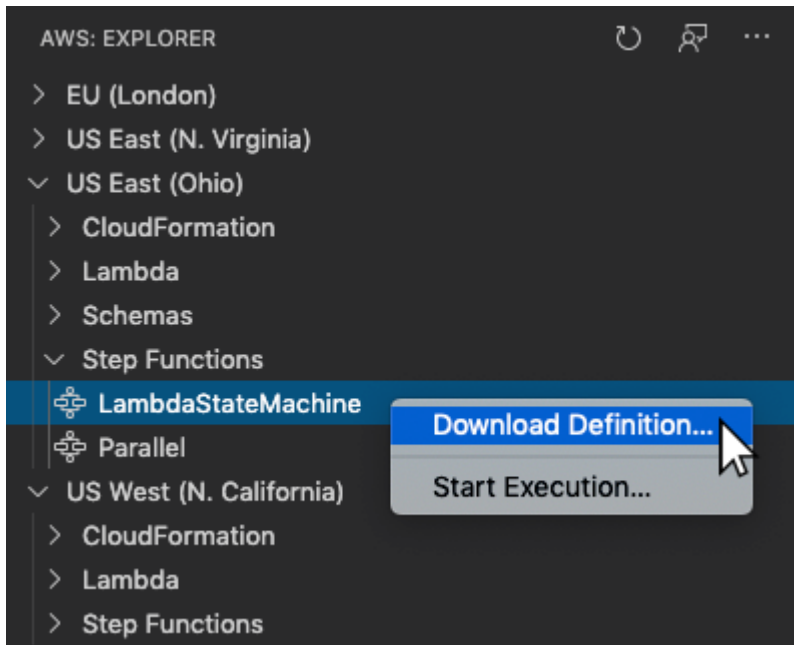
1. Abra AWS Explorer.
2. Seleccione Step Functions.
3. Compruebe que muestra todas las máquinas de estado de su cuenta.



Descargar una máquina de estado

Para descargar una máquina de estado:

1. Desde el Explorador de AWS , haga clic con el botón derecho en la máquina de estado que desea descargar.
2. Seleccione Download (Descargar) y, a continuación, seleccione la ubicación en la que desea descargar la máquina de estado.
3. Verifique que se haya descargado correctamente.



Creación de una máquina de estado

Puede crear una nueva máquina de estado usted mismo o puede utilizar una plantilla. Para obtener más información sobre cómo crear una máquina de estado a partir de una plantilla, consulte la sección [State Machine Template \(Plantillas de máquina de estado\)](#). Para crear una nueva máquina de estado:

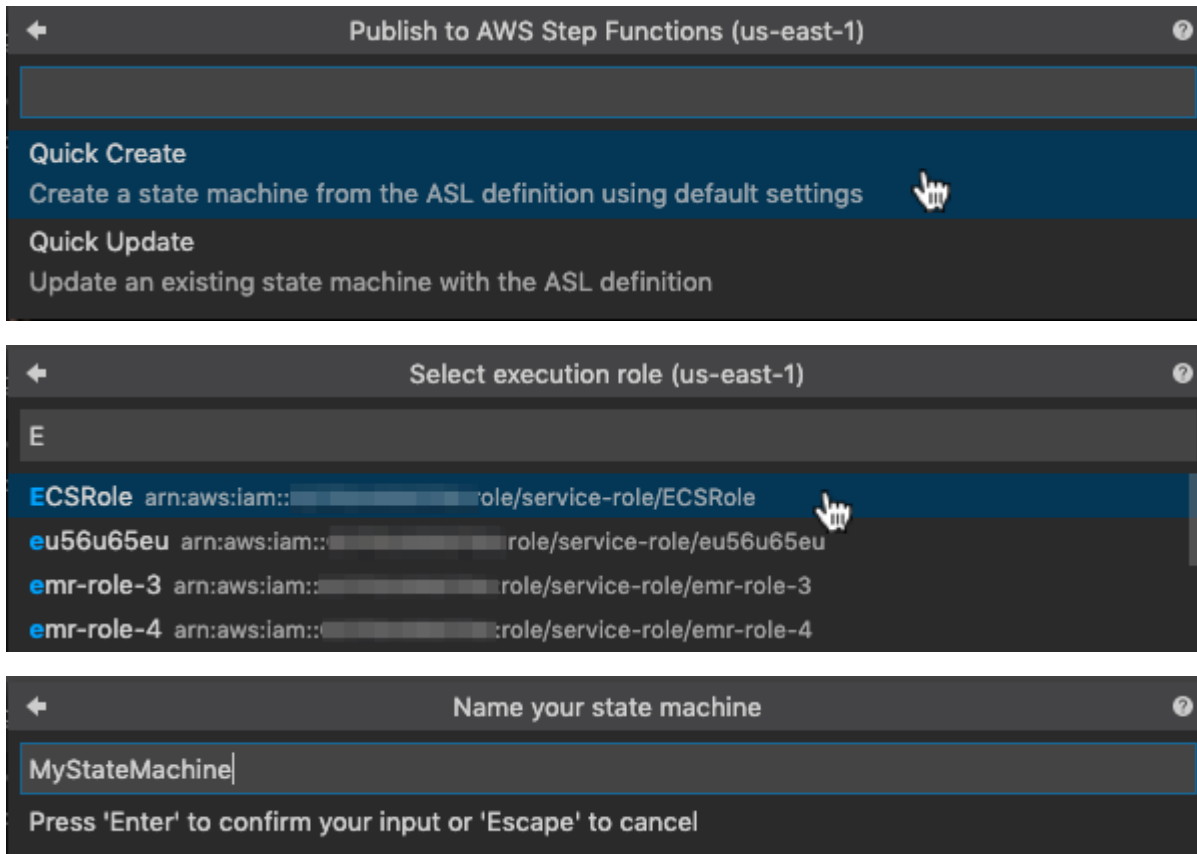
1. Cree un nuevo archivo de [Lenguaje de estados de Amazon](#) (ASL) con la definición de máquina de estado. Utiliza el menú de la parte inferior derecha para configurarlo como lenguaje de estados de Amazon.
2. Seleccione **Publish to Step Functions (Publicar en Step Functions)**.

```

Publish to Step Functions | Render graph
1  {
2    "StartAt": "FirstState",
3    "States": {
4      "FirstState": {
5        "Type": "Task",
6        "Resource": "arn:aws:lambda:us-east-1:123456789012:function:Function",
7        "Next": "ChoiceState"
8      },
9      "ChoiceState": {
10     "Type": "Choice",

```

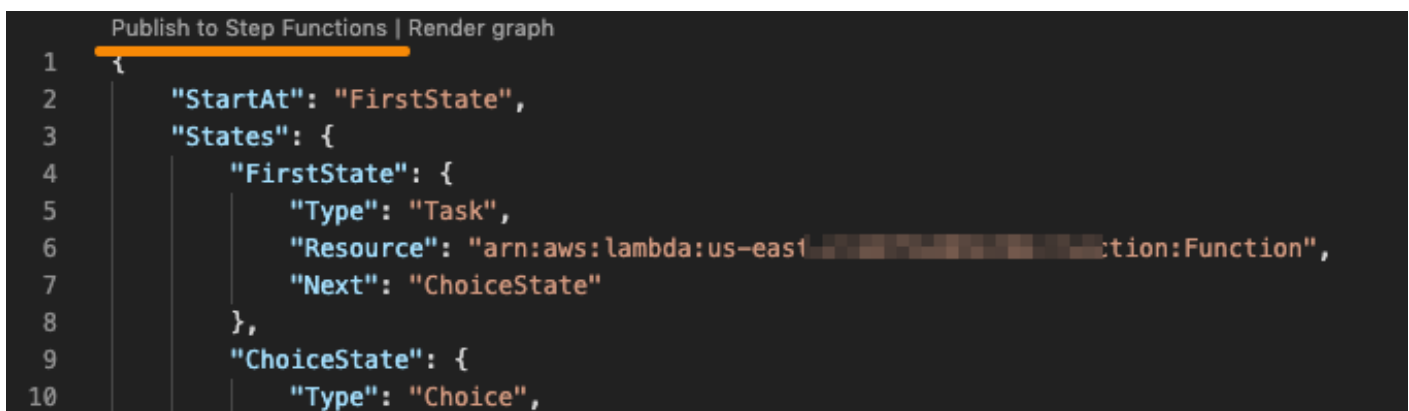
3. Seleccione **Quick Create (Creación rápida)**, elija un rol y asigne un nombre a su máquina de estado.



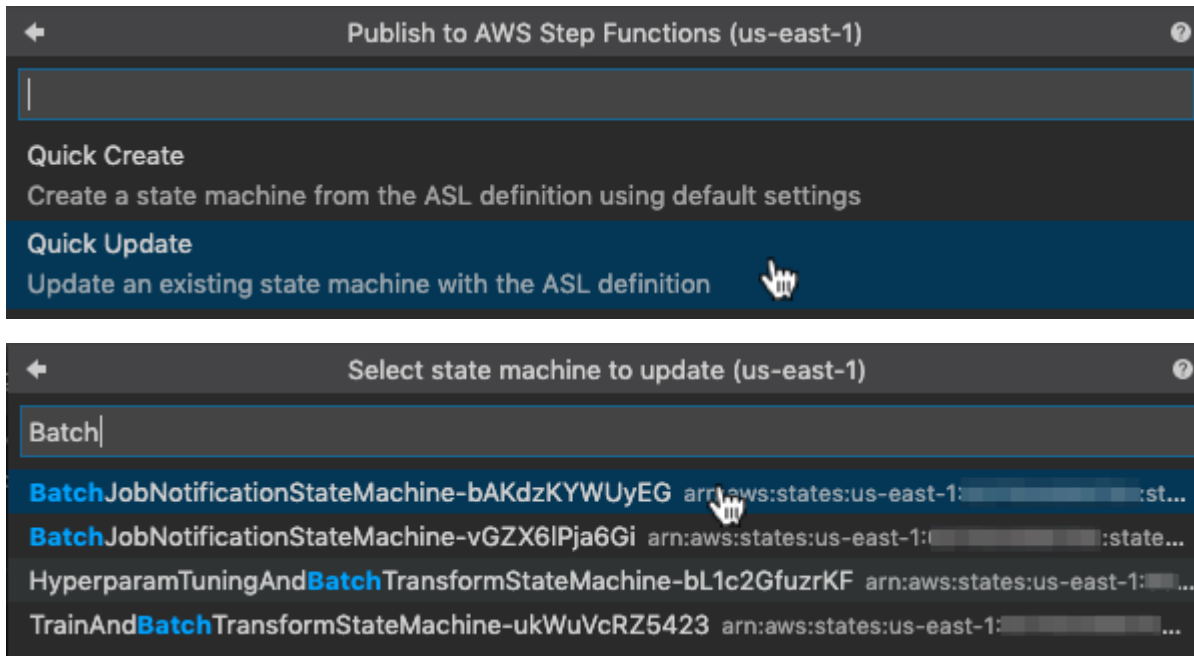
Actualización de la máquina de estado

Para actualizar una máquina de estado

1. Edite el archivo ASL con la definición de máquina de estado.
2. Seleccione Publish to Step Functions (Publicar en Step Functions).



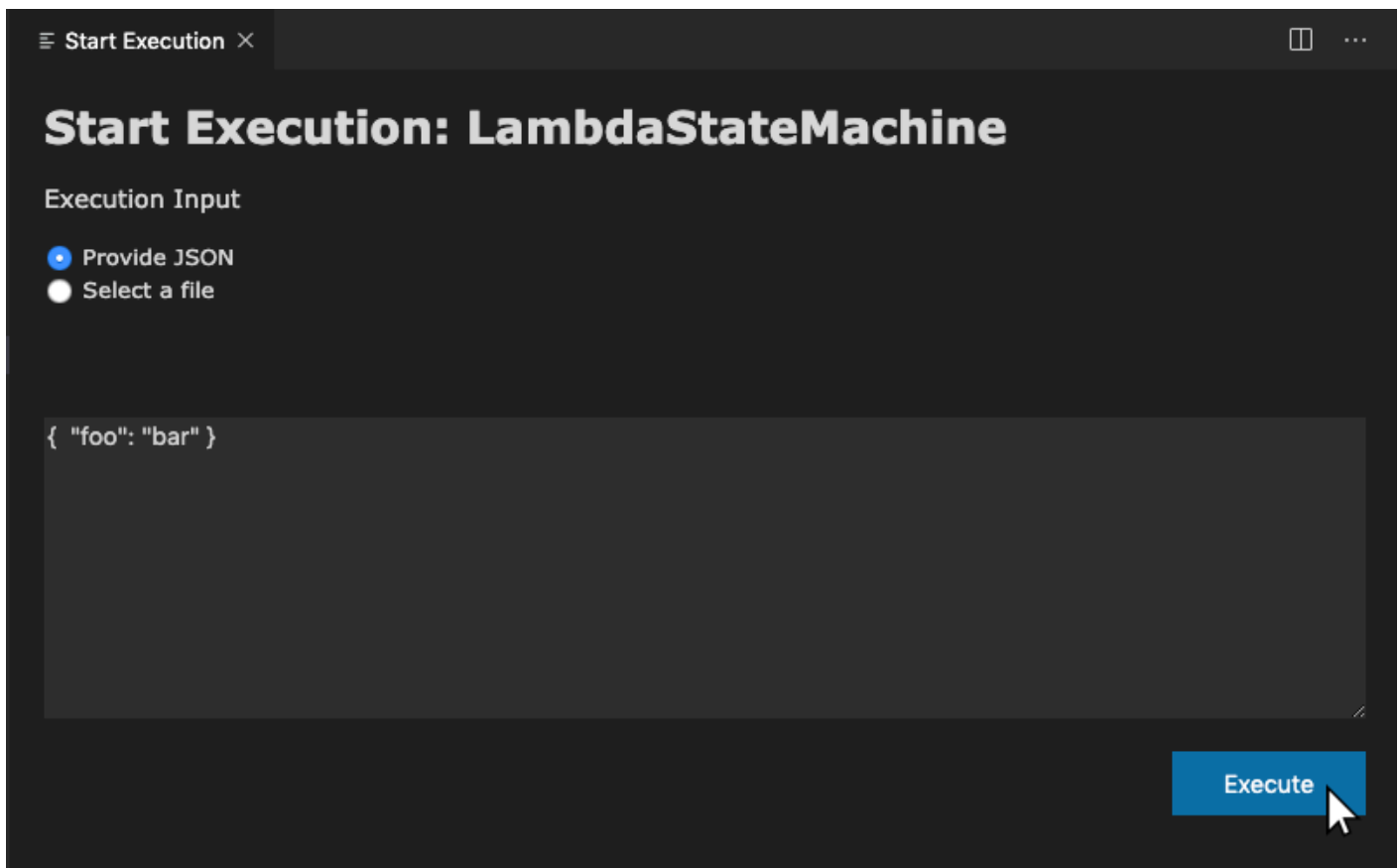
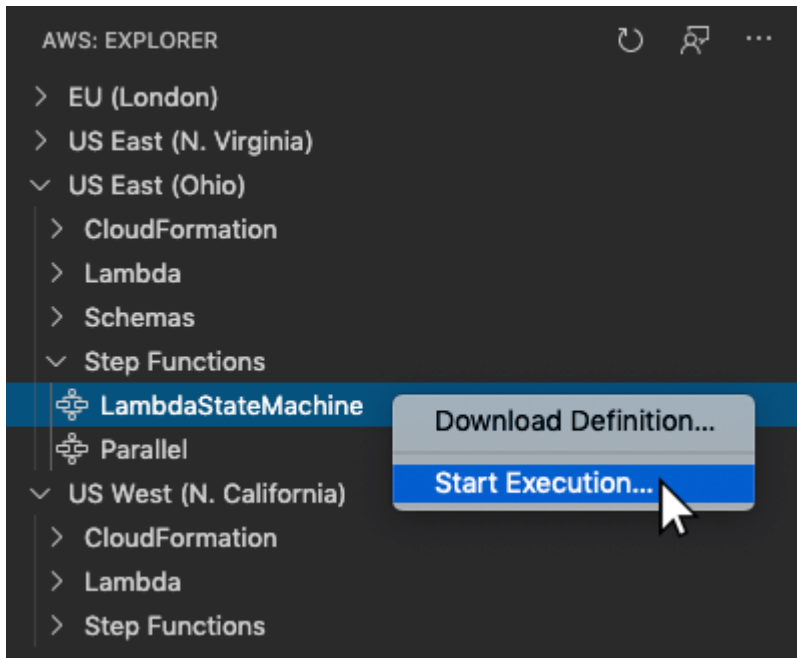
3. Seleccione Quick Update (Actualización rápida), y, a continuación, seleccione la máquina de estado que desea actualizar.



Ejecución de una máquina de estado

Para ejecutar una máquina de estado:

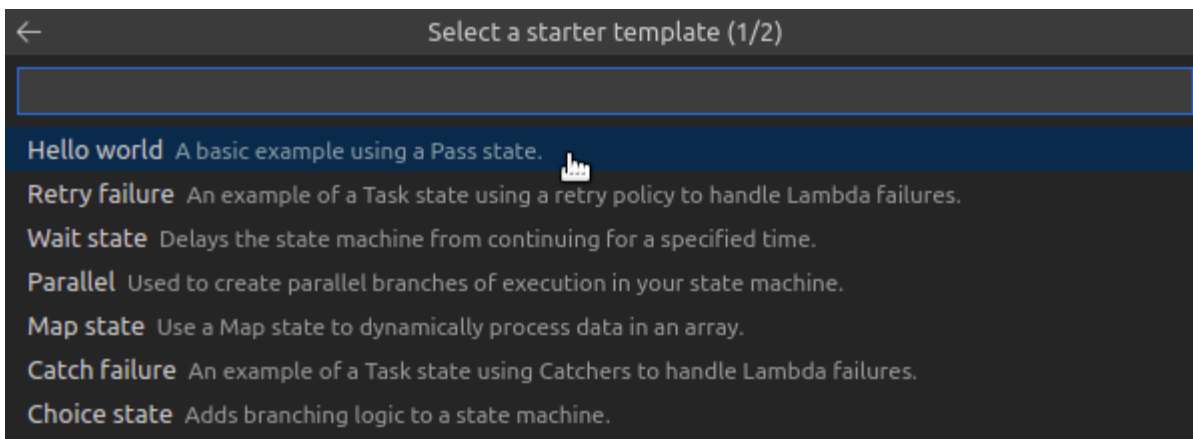
1. Desde el Explorador de AWS , haga clic con el botón derecho en la máquina de estado que desea ejecutar.
2. Proporcione la entrada de la máquina de estado. Puede probar la entrada desde un archivo y la entrada en un cuadro de texto.
3. Inicie la máquina de estado y verifique que se ejecuta correctamente.



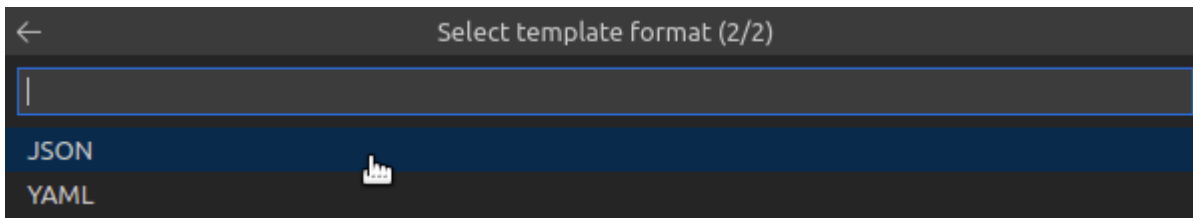
Plantillas de máquina de estado

Cuando crea una máquina de estado, tiene la opción de crearla a partir de una plantilla. La plantilla contiene una definición de máquina de estado de ejemplo con varios estados de uso común y proporciona un punto de partida. Para utilizar plantillas de máquina de estado:

1. Abra la paleta de comandos en VS Code.
2. Seleccione Kit de herramientas de AWS : crear una nueva máquina de estado Step Functions.
3. Elija la plantilla que desea utilizar.



4. Elija si quiere usar el formato de plantilla JSON o YAML.



Visualización de gráfico de máquina de estado

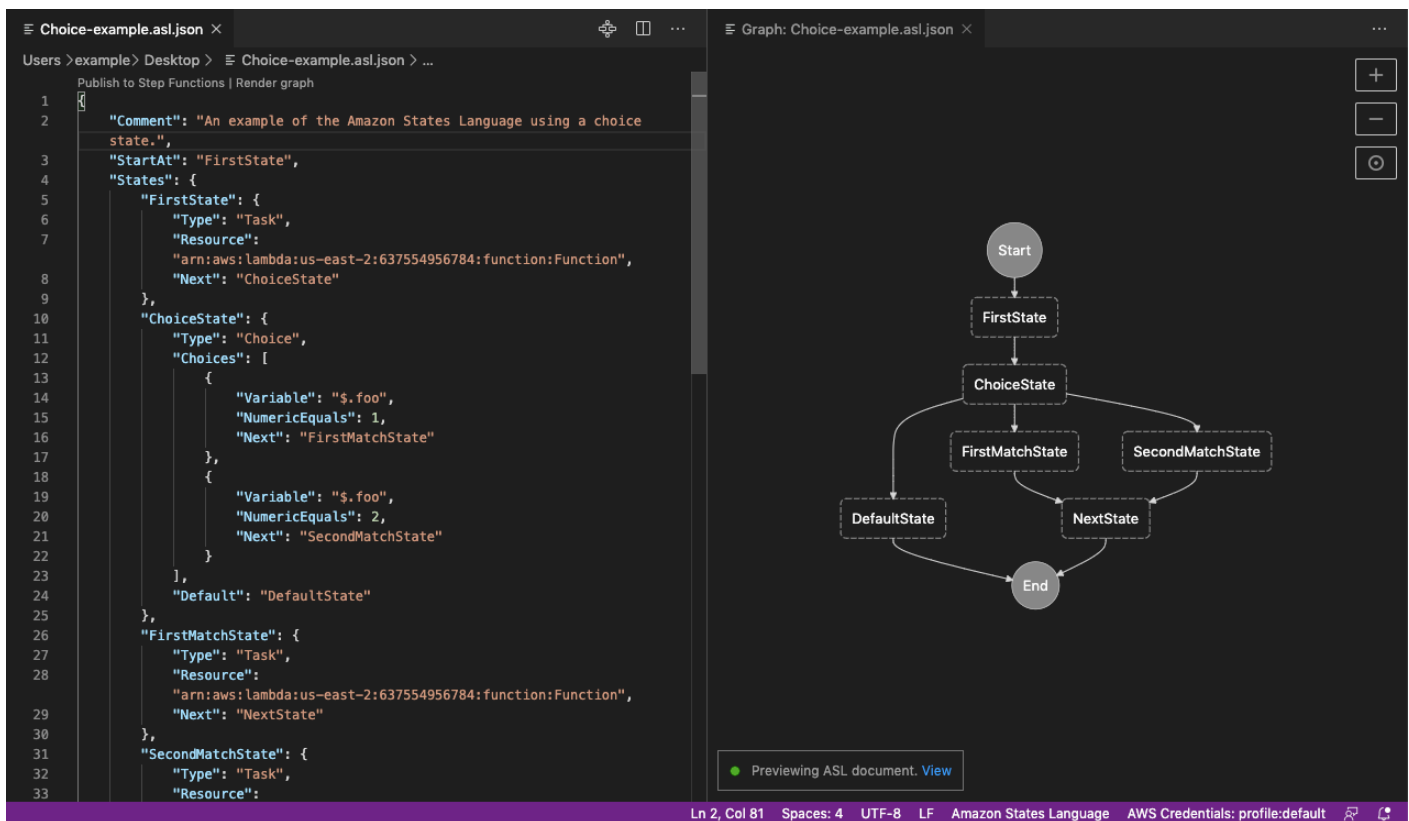
Las visualizaciones de gráficos le permiten ver cómo se ve su máquina de estado en formato gráfico. Cuando cree una visualización gráfica, se abrirá otra pestaña y mostrará una visualización del JSON o YAML de la máquina de estado. A continuación, puede comparar la definición de máquina de estado que está escribiendo simultáneamente con su visualización. A medida que cambie la definición de la máquina de estado, la visualización se actualizará.

Note

Para crear una visualización de una definición de máquina de estado, la definición debe estar abierta en el editor activo. Si cierra o cambia el nombre del archivo de definición, la visualización se cerrará.

Para crear una visualización de gráfico de máquina de estado:

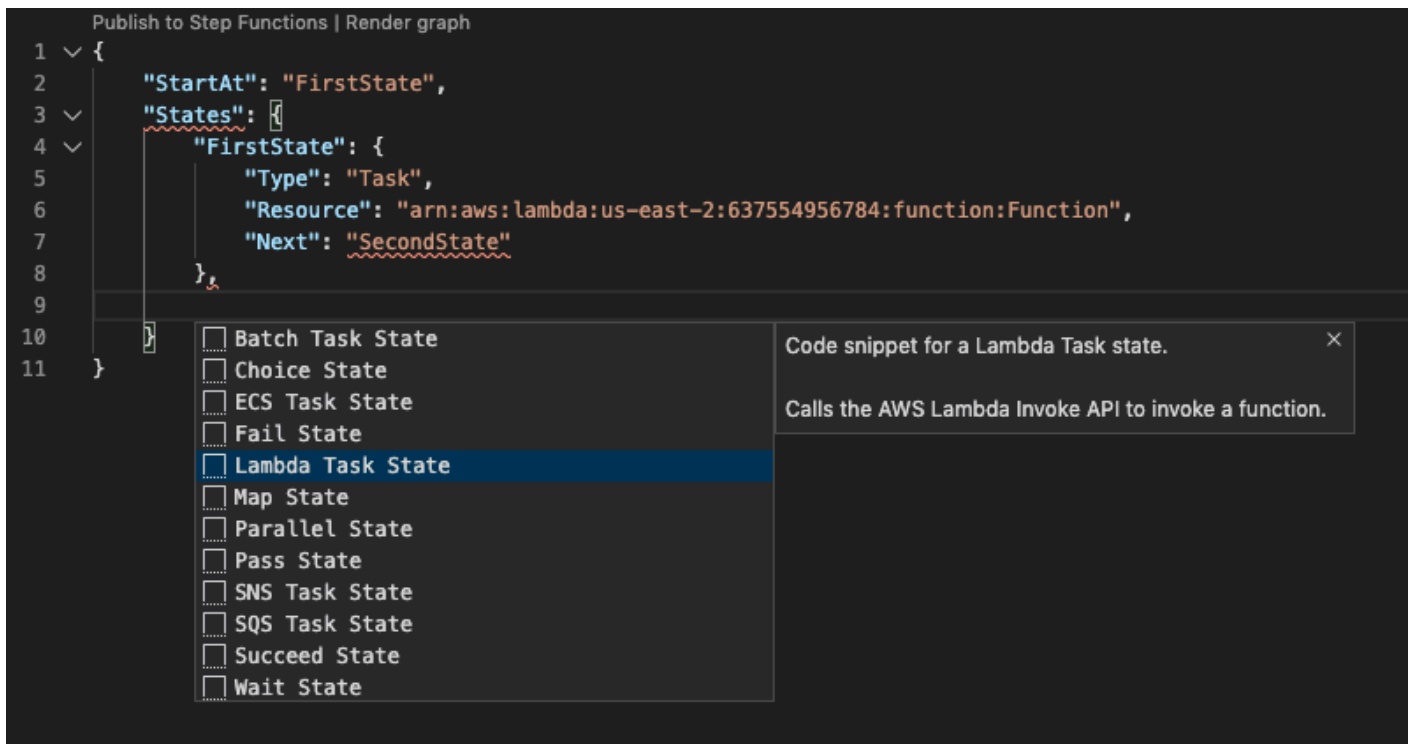
1. Defina su máquina de estado.
2. Abra la paleta de comandos en VS Code.
3. Para crear una visualización, utilice el botón de visualización en la esquina superior derecha o elija Gráfico de representación de AWS .



Fragmentos de código

Los fragmentos de código le permiten insertar secciones cortas de código. Para usar fragmentos de código:

1. Abra un archivo y guárdelo con la extensión `.asl.json` para el formato JSON o `.asl.yaml` para el formato YAML.
2. Cree una nueva máquina de estado con la propiedad `States` (Estados).
3. Coloque el cursor dentro de `States` (Estados).
4. Utilice la combinación de teclas `Control + Space` y seleccione su fragmento de código preferido.
5. Utilice `Tab` para recorrer la variable y los parámetros en el fragmento de código.
6. Pruebe los fragmentos `Retry` y `Catch` colocando el cursos en el estado relacionado.



Finalización y validación de código

Para ver cómo funciona la finalización del código:

1. Cree varios estados.
2. Coloque el cursor después de una propiedad siguiente o predeterminada. `StartAt`
3. Utilice la combinación de teclas `Control + Space` para enumerar las finalizaciones disponibles. Se puede acceder a propiedades adicionales utilizando `Control + Space` de nuevo y se basarán en el `Type` de `State`.
4. A medida que trabaje, se llevará a cabo la validación de código para:

- Propiedades que faltan
- Valores incorrectos
- Estado no terminal
- Estados inexistentes a los que se señala

```

"FirstMatchState": {
  "Type": "Task",
  "Resource": "arn:aws:lambda:us-east-2:637554956784:function:Function",
  "Next": ""
},
"SecondMatchState": {
  "Type": "ChoiceState",
  "Resource": "arn:aws:lambda:us-east-2:637554956784:function:Function",
  "Next": "SecondMatchState"
},
"DefaultState": {
  "Type": "Fail",
  "Error": "DefaultStateError",
  "Cause": "No Matches!"
},
"NextState": {
  "Type": "Task",
  "Resource": "arn:aws:lambda:us-east-2:637554956784:function:Function",
  "End": true
}
    
```

```

"FirstMatchState": {
  "Type": "Task",
  "Resource": "arn:aws:lambda:us-east-2:637554956784:function:Function",
  "Catch": [
    {
      "Error": "DefaultStateError",
      "Next": "SecondMatchState"
    }
  ]
},
    
```

An array of objects, called Catchers, that define a fallback state. This state is executed if the state encounters runtime errors and its retry policy is exhausted or isn't defined.

Trabajando con Threat Composer

Puede utilizarla 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 el proceso de modelado de amenazas.

Para obtener información detallada sobre la herramienta Threat Composer, consulte el repositorio [Threat Composer GitHub](#).

En los temas siguientes se describe cómo trabajar con Threat Composer en el AWS Toolkit for Visual Studio Code.

Temas

- [Cómo trabajar con Threat Composer desde el kit de herramientas](#)

Cómo trabajar con Threat Composer desde el kit de herramientas

Con Threat Composer, puede crear, ver y editar los modelos de amenazas de Threat Composer directamente en VS Code. Para obtener información detallada sobre la herramienta Threat Composer, consulte el [GitHub repositorio de Threat Composer](#).

En las siguientes secciones se describe cómo acceder a las herramientas de Threat Composer del 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 abre automáticamente y muestra una visualización del archivo del modelo de amenazas en la ventana del editor de VS Code.

Creación de un nuevo modelo de amenazas de Threat Composer

1. En el menú principal de VS Code, expanda Archivo y, a continuación, seleccione Nuevo archivo.
2. En el cuadro de diálogo Nuevo archivo, seleccione Threat Composer File... .

3. Cuando se le pida, introduzca una `yfile` name, a continuación, pulse la **enter** tecla para abrir Threat Composer y crear una visualización del archivo de modelo de amenazas vacío en una nueva ventana del editor de VS Code.

Crear un nuevo modelo de amenazas de Threat Composer a partir de la paleta de comandos

1. Desde VS Code, abra la paleta de comandos pulsando **Cmd + Shift + P** o **Ctrl + Shift + P** (Windows).
2. En el campo de búsqueda, introduzca **Threat Composer** y seleccione Crear nuevo archivo de Threat Composer cuando aparezca en los resultados.
3. Cuando se le pida, introduzca una `yfile` name, a continuación, pulse la **enter** tecla para abrir Threat Composer y crear una visualización de su archivo de modelo de amenazas vacío en una nueva ventana del editor de VS Code.

Usar recursos de

Además de acceder a servicios de AWS que se enumeran de forma predeterminada en AWS Explorer, también puede ir a Resources (Recursos) y elegir entre cientos de recursos para agregarlos a la interfaz. En AWS, un recurso es una entidad con la que se puede trabajar. Algunos de los recursos que se pueden agregar son Amazon AppFlow, Amazon Kinesis Data Streams, roles de AWS IAM, Amazon VPC y 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:

- Ver una lista de los recursos existentes que están disponibles en la región actual de AWS de 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.
- Ver la documentación de AWS 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 del AWS Toolkit for Visual Studio Code, la opción para crear, editar y eliminar los recursos es una característica experimental. 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.

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 Experimentos con el kit de herramientas de AWS, seleccione `jsonResourceModification` para permitirle crear, actualizar y eliminar recursos.

Para obtener más información, consulte [Uso de características experimentales](#).

Permisos de IAM para acceder a los recursos

Necesita permisos de AWS Identity and Access Management específicos para acceder a los recursos asociados con los servicios de AWS. Por ejemplo, una entidad de IAM, como un usuario o un rol, requiere permisos de `Lambda::Function` para acceder a los recursos de AWS Toolkit.

Además de los permisos para los recursos de servicio, una entidad de IAM requiere permisos para que el kit de herramientas para VS Code pueda llamar a las operaciones de la API de control en la nube de AWS en su nombre. Las operaciones de la API de Cloud Control permiten al usuario o rol de IAM acceder y actualizar los recursos remotos.

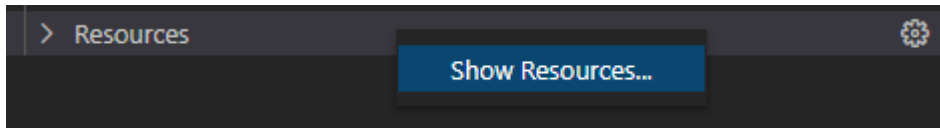
La forma más sencilla de conceder permisos consiste en adjuntar la política administrada de AWS, `PowerUserAccess`, a la entidad de IAM que llama a estas operaciones de la API mediante la interfaz de Toolkit. Esta [política administrada](#) otorga una serie de permisos para realizar tareas de desarrollo 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 desee 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.

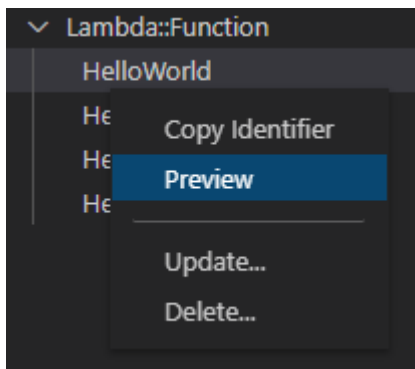
Cuando aparezca la plantilla de recursos, puede modificarla pulsando el icono de actualización situado a la derecha de la pestaña del editor. Para actualizar un recurso, debe tener activada la [???](#) necesaria.

- 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 [???](#) en esta versión de AWS Toolkit for Visual Studio Code).

Warning

Si se elimina un recurso, cualquier pila de AWS CloudFormation que utilice ese recurso no se actualizará. Para corregir este error de actualización, es necesario volver a crear el recurso o eliminar la referencia al mismo en la plantilla AWS CloudFormation 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.

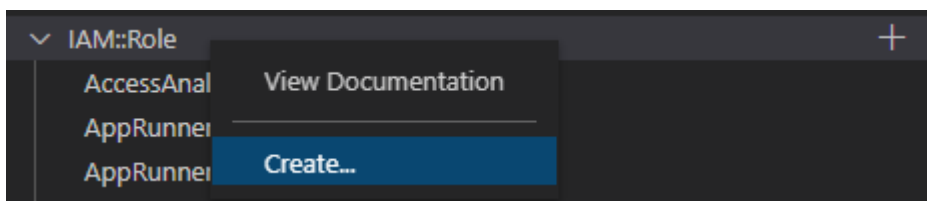
La creación de un nuevo recurso implica agregar un tipo de recurso a la lista de Recursos y, a continuación, editar una plantilla con formato JSON que defina el recurso, sus propiedades y sus atributos.

Por ejemplo, un recurso que pertenece al tipo de recurso AWS Toolkit:SageMaker::UserProfile se define con una plantilla que crea un perfil de usuario para Amazon SageMaker 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.

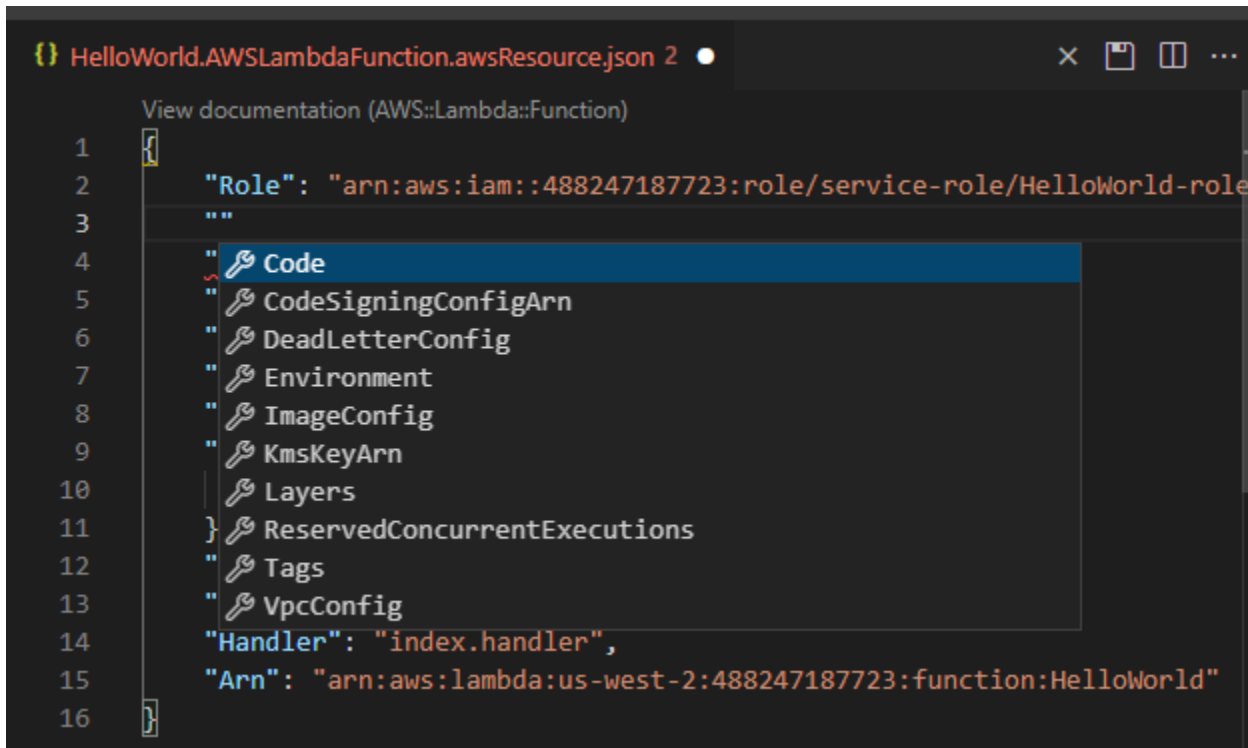
1. Agregue el tipo de recurso del recurso que desea crear haciendo clic con el botón derecho en Recursos y, luego, elija Mostrar recursos.
2. Después de añadir el tipo de recurso en Recursos, seleccione el icono con el signo más (“+”) para abrir el archivo de plantilla en un editor nuevo.

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.



```
1 {
2   "Role": "arn:aws:iam::488247187723:role/service-role/HelloWorld-role",
3   ""
4   "Code"
5   "CodeSigningConfigArn"
6   "DeadLetterConfig"
7   "Environment"
8   "ImageConfig"
9   "KmsKeyArn"
10  "Layers"
11  "ReservedConcurrentExecutions"
12  "Tags"
13  "VpcConfig"
14  "Handler": "index.handler",
15  "Arn": "arn:aws:lambda:us-west-2:488247187723:function:HelloWorld"
16 }
```

4. Cuando termine de declarar el recurso, elija Guardar para validar la plantilla y guardar el recurso en la nube de AWS 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.

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. Usted es responsable de mantener el control sobre el contenido alojado en esta infraestructura. Este contenido incluye la configuración de seguridad y las tareas de administración para el que utiliza Servicios de AWS. Para obtener más información sobre la privacidad de los datos, consulte las [Preguntas frecuentes sobre la privacidad de datos](#). Para obtener información sobre la protección de datos en Europa, consulte la publicación de blog [AWSShared Responsibility Model and GDPR](#) en el Blog de seguridad de AWS.

Con fines de protección de datos, recomendamos proteger las credenciales de 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 con sus obligaciones laborales. También recomendamos proteger sus datos de las siguientes formas:

- Utilice autenticación multifactor (MFA) en cada cuenta.
- Utilice SSL/TLS para comunicarse con los recursos de AWS. Nosotros exigimos TLS 1.2 y recomendamos TLS 1.3.
- Configure la API y el registro de actividad del usuario con AWS CloudTrail.
- Utilice las soluciones de cifrado de AWS, junto con todos los controles de seguridad predeterminados dentro de los servicios de Servicios de AWS.
- Utilice 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 140-2 al acceder a AWS a través de una interfaz de línea de comandos o una API, utilice un punto de conexión de FIPS. Para obtener más información sobre los puntos de conexión de FIPS disponibles, consulte [Estándar de procesamiento de la información federal \(FIPS\) 140-2](#).

Se recomienda encarecidamente no ingresar 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 formato libre utilizados para nombres se pueden 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 revisión de la Guía del usuario de AWS Toolkit for Visual Studio Code

Última actualización importante de la documentación: 14 de octubre de 2021

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
Guía de usuario creada: Crear una plantilla de CloudFormation	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
Actualización menor de la IU	Se cambió 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
Guía del usuario creada para Amazon Elastic Container Service Exec	Esta es una descripción general de Amazon ECS Exec.	13 de diciembre de 2021
Guía de usuario creada para el servicio AWS IoT del kit de herramientas para VS Code	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
Compatibilidad con características experimentales	Se ha añadido compatibilidad para activar características	14 de octubre de 2021

	sticas experimentales en los servicios de AWS.	
Compatibilidad con recursos de AWS	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
Información general sobre el servicio Amazon ECR para el AWS Toolkit for Visual Studio Code	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
Compatibilidad con entornos ARM64	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
Aplicación sin servidor de AWS	Se ha añadido compatibilidad para ejecutar aplicaciones de AWS SAM en la plataforma ARM64	30 de septiembre de 2021
Actualización del formato de la sección Node.js	Se ha actualizado el formato de Node.js/ TypeScript según los comentarios de los clientes.	12 de agosto de 2021
Compatibilidad con App Runner	Se ha agregado soporte para AWS App Runner para AWS Toolkit for Visual Studio Code.	11 de agosto de 2021
Depuración de las funciones de Go	Se ha agregado compatibilidad para depurar las funciones locales de Go.	10 de mayo de 2021

Depuración de las funciones de Java	Se ha agregado compatibilidad para depurar las funciones locales de Java.	22 de abril de 2021
Compatibilidad de YAML con AWS Step Functions	Se ha agregado compatibilidad de YAML con AWS Step Functions.	4 de marzo de 2021
Depuración de recursos de Amazon API Gateway	Se ha añadido compatibilidad con la depuración de recursos locales de Amazon API Gateway.	1 de diciembre de 2020
Amazon API Gateway	Se ha agregado compatibilidad con Amazon API Gateway.	1 de diciembre de 2020
Aplicación sin servidor de AWS	Se ha añadido compatibilidad de las imágenes de contenedor de Lambda con las aplicaciones sin servidor.	1 de diciembre de 2020
Compatibilidad con AWS Systems Manager	Se ha añadido compatibilidad con los documentos de Automatización de Systems Manager.	30 de septiembre de 2020
Registros de CloudWatch	Se ha agregado compatibilidad de para Registros de CloudWatch.	24 de agosto de 2020
Amazon S3	Se ha agregado compatibilidad con Amazon S3.	30 de julio de 2020
Compatibilidad con AWS Step Functions	Se agregó compatibilidad con AWS Step Functions.	31 de marzo de 2020
Contenido de seguridad	Se ha añadido contenido de seguridad.	6 de febrero de 2020

Trabajo con esquemas de Amazon EventBridge	Se ha agregado compatibilidad con esquemas de Amazon EventBridge	1 de diciembre de 2019
AWS CDK	Versión de prueba del servicio AWS CDK.	25 de noviembre de 2019
Uso de un proceso de credenciales externo	Se ha añadido información sobre el uso de un proceso de credenciales externo para obtener credenciales de AWS.	25 de septiembre de 2019
Uso de IntelliSense para archivos de definición de tareas	Se ha agregado compatibilidad con IntelliSense para trabajar con archivos de definición de tareas de Amazon ECS.	24 de septiembre de 2019
Guía del usuario del AWS Toolkit for Visual Studio Code	Versión de disponibilidad general.	11 de julio de 2019
Guía del usuario del AWS Toolkit for Visual Studio Code	Se actualizó la estructura del documento para mayor claridad y facilidad de uso.	3 de julio de 2019
Instalación de la AWS Toolkit for Visual Studio Code	Se agregó información sobre la instalación de los SDK de lenguaje para admitir varias cadenas de herramientas.	12 de junio de 2019
Configuración de la cadena de herramientas	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.