

Guía del usuario

AWS CodeBuild



Versión de API 2016-10-06

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS CodeBuild: 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

¿Qué es AWS CodeBuild?	1
.....	1
¿Cómo se ejecuta CodeBuild	1
Precios para CodeBuild	3
¿Cómo puedo empezar CodeBuild?	3
Conceptos	3
Cómo funciona CodeBuild	3
Pasos siguientes	5
Introducción	6
Primeros pasos con la consola	6
Pasos	6
Paso 1: Crear el código fuente	7
Paso 2: Crear el archivo de especificación de compilación	10
Paso 3: Crear dos buckets de S3	12
Paso 4: Cargar el código fuente y el archivo buildspec	13
Paso 5: Crear el proyecto de compilación	15
Paso 6: Ejecutar la compilación	17
Paso 7: Ver información resumida sobre la compilación	17
Paso 8: Ver información detallada sobre la compilación	19
Paso 9: Obtener el artefacto de salida de la compilación	20
Paso 10: Eliminar los buckets de S3	21
Encapsulación	22
Primeros pasos con AWS CLI	22
Pasos	22
Paso 1: Crear el código fuente	23
Paso 2: Crear el archivo de especificación de compilación	26
Paso 3: Crear dos buckets de S3	28
Paso 4: Cargar el código fuente y el archivo buildspec	29
Paso 5: Crear el proyecto de compilación	31
Paso 6: Ejecutar la compilación	35
Paso 7: Ver información resumida sobre la compilación	37
Paso 8: Ver información detallada sobre la compilación	40
Paso 9: Obtener el artefacto de salida de la compilación	42
Paso 10: Eliminar los buckets de S3	43

Encapsulación	44
Muestras	45
Ejemplos basados en casos de uso	45
Muestras de servicios cruzados	46
Ejemplo de insignias de compilación	88
Crear un informe de pruebas con el ejemplo de AWS CLI	93
Muestras de Docker para CodeBuild	100
Alojar la salida de la compilación en un bucket de S3	115
Ejemplo de varios orígenes de entrada y varios artefactos de salida	118
Versiones del entorno de ejecución en el ejemplo del archivo buildspec	122
Ejemplo de versión de origen	131
Muestras de repositorios de fuentes de terceros para CodeBuild	135
Ejemplo para utilizar el control de versiones semánticas para asignar nombre a artefactos de compilación	154
Ejemplos de Windows	156
Ejecución de los ejemplos	157
Estructura de directorios	158
Archivos	159
Planificación de una compilación	178
Referencia de la especificación de compilación	180
Nombre de archivo y ubicación de almacenamiento de buildspec	181
Sintaxis de buildspec	182
Ejemplo de un archivo buildspec	204
Versiones de buildspec	207
Referencia de especificaciones de compilación por lotes	208
Referencia de entornos de compilación	216
Imágenes de Docker proporcionadas por CodeBuild	217
Modos y tipos de computación del entorno de compilación	239
Intérpretes de comandos y comandos de los entornos de compilación	247
Variables de entorno en los entornos de compilación	249
Tareas en segundo plano en entornos de compilación	254
Compilar localmente	255
Requisitos previos	255
Configure la imagen de compilación	255
Ejecutar el agente de CodeBuild	256
Recepción de notificaciones de versiones nuevas del agente de CodeBuild	257

Compatibilidad con VPC	259
Casos de uso	259
Permitir el acceso de Amazon VPC a sus proyectos CodeBuild	260
Prácticas recomendadas para las VPC	261
Solución de problemas con la configuración de la VPC	262
Limitaciones de las VPC	263
Uso de puntos de enlace de la VPC	263
Antes de crear los puntos de enlace de la VPC	263
Creación de puntos de conexión de VPC para CodeBuild	264
Crear una política de punto de conexión de VPC para CodeBuild	265
Plantilla de VPC de AWS CloudFormation	265
Uso de un servidor proxy	271
Componentes necesarios para ejecutar CodeBuild en un servidor proxy	272
Ejecución de CodeBuild en un servidor proxy explícito	275
Ejecución de CodeBuild en un servidor proxy transparente	280
Ejecución de un administrador de paquetes y otras herramientas en un servidor proxy	281
Trabajar con proyectos de compilación y compilaciones	284
Trabajar con proyectos de compilación	284
Creación de un proyecto de compilación	285
Creación de una regla de notificación	328
Ver una lista de nombres de proyectos de compilación	332
Ver los detalles de un proyecto de compilación	334
Almacenamiento en caché de compilaciones	337
Desencadenadores de compilación	342
GitLab conexiones	350
Webhooks	356
Cambiar la configuración de un proyecto de compilación	402
Eliminar un proyecto de compilación	428
Uso de proyectos compartidos	430
Etiquetado de un proyecto	435
Compilaciones por lotes	441
GitHub Corredor de acción	445
Proyectos de compilación pública	468
Trabajo con compilaciones	469
Ejecutar una compilación	470
Ver detalles de las compilaciones	482

Ver una lista de identificadores de compilación	485
Ver una lista de identificadores de compilación de un proyecto de compilación	488
Detener una compilación	492
Detener una compilación por lotes	494
Volver a intentar una compilación	496
Session Manager	497
Eliminación de compilaciones	502
Trabajar con AWS Lambda computación	504
¿Qué herramientas y entornos de tiempo de ejecución se incluyen en las imágenes de Docker del entorno de tiempo de ejecución seleccionado que se ejecuta en AWS Lambda?	504
¿Qué pasa si la imagen seleccionada no incluye las herramientas que necesito?	504
¿En qué regiones se admite la AWS Lambda computación CodeBuild?	505
Limitaciones de AWS Lambda cómputo	505
AWS Lambda calcular muestras	506
Implemente una función Lambda con AWS SAM CodeBuild Lambda Java	506
Cree una aplicación React de una sola página con CodeBuild Lambda Node.js	511
Actualizar la configuración de una función Lambda con CodeBuild Lambda Python	514
Trabajar con capacidad reservada	519
¿Cómo comienzo a utilizar las flotas de capacidad reservada?	520
Prácticas recomendadas	520
¿Puedo compartir una flota de capacidad reservada en varios CodeBuild proyectos?	521
¿Qué regiones admiten flotas con capacidad reservada?	521
Propiedades de flota con capacidad reservada	521
Muestras de capacidad reservada	524
Ejemplo de almacenamiento en caché con capacidad reservada	524
Limitaciones de las flotas de capacidad reservada	526
Uso de informes de pruebas	527
Crear un informe de pruebas	528
Uso de grupos de informes	530
Crear un grupo de informes	531
Actualizar un grupo de informes	536
Especificar archivos de prueba	540
Especificar comandos de prueba	540
Nomenclatura de grupos de informes	541
Etiquetar un grupo de informes	542
Usar grupos de informes compartidos	548

Uso de informes	554
Uso de permisos de informes de pruebas	555
Crear un rol para los informes de pruebas	556
Permisos para las operaciones de informes de pruebas	558
Ejemplos de permisos de informes de pruebas	558
Consultar los informes de pruebas	559
Consultar los informes de pruebas de una compilación	559
Consultar los informes de pruebas de un grupo de informes	560
Consultar los informes de pruebas de una cuenta de AWS	560
Generación de informes de pruebas con marcos de trabajo de pruebas	560
Generación de informes con Jasmine	561
Generación de informes con Jest	563
Generación de informes con pytest	564
Generación de informes con RSpec	565
Informes de cobertura de código	566
.....	566
Crear un informe de cobertura de código	567
Detección automática de informes	568
Configure la detección automática de informes mediante la consola	569
Configurar la detección automática de informes mediante variables de entorno del proyecto	570
Registro y monitorización	571
Registrar llamadas a la API de AWS CodeBuild con AWS CloudTrail	571
Información de AWS CodeBuild en CloudTrail	571
Descripción de las entradas de los archivos de registro de AWS CodeBuild	572
Monitorización de AWS CodeBuild	575
Métricas de CloudWatch	576
Métricas de CloudWatch de la utilización de recursos	578
Dimensiones de CloudWatch	580
Alarmas de CloudWatch	580
Métricas de CodeBuild	581
Métricas de utilización de recursos de CodeBuild	583
Alarmas de CodeBuild	587
Seguridad	589
Protección de datos	589
Cifrado de datos	591

Administración de claves	592
Privacidad de tráfico	592
Administración de identidades y accesos	593
Información general sobre la administración del acceso	593
Uso de políticas basadas en identidades	597
AWS CodeBuild referencia de permisos	627
Uso de etiquetas para controlar el acceso a los recursos de AWS CodeBuild	634
Visualización de recursos en la consola	638
Validación de conformidad	638
Resiliencia	639
Seguridad de la infraestructura	640
Acceso al proveedor de fuentes	640
GitHub y un token de acceso a GitHub Enterprise Server	641
GitHub Aplicación OAuth	645
Contraseña o token de acceso de la aplicación Bitbucket	645
Aplicación OAuth de Bitbucket	650
Prevenición de la sustitución confusa entre servicios	650
Temas avanzados	652
Configuración avanzada	652
Agregar permisos de acceso de CodeBuild a un grupo o un usuario de IAM	653
Crear un rol de servicio de CodeBuild	660
Creación de una clave administrada por el cliente	668
Instalación y configuración de AWS CLI	670
Referencia de la línea de comandos	671
Referencia de los SDK y las herramientas de AWS	672
SDK y herramientas de AWS compatibles con AWS CodeBuild	673
Especificación del punto de conexión	673
Especificación del punto de conexión de AWS CodeBuild (AWS CLI)	674
Especificación del punto de conexión de AWS CodeBuild (SDK de AWS)	675
Úselo CodePipeline con CodeBuild	677
Requisitos previos	678
Crear una canalización (consola)	680
Crear canalización (AWS CLI)	685
Añadir acción de compilación	690
Añadir una acción de prueba	694
Uso de CodeBuild con Jenkins	698

Configuración de Jenkins	698
Instalación del complemento	698
Uso del complemento	698
Uso de CodeBuild con Codecov	700
Integrar Codecov en un proyecto de compilación	701
Aplicaciones sin servidor	704
Recursos relacionados	53
Solución de problemas	706
Apache Maven crea artefactos de referencia en un repositorio incorrecto	707
De forma predeterminada, los comandos de compilación se ejecutan como usuario raíz	709
Las compilaciones pueden generar un error si los nombres de archivo contienen caracteres ajenos al inglés (Estados Unidos)	709
Las compilaciones pueden fallar al obtener parámetros del almacén de parámetros de Amazon EC2	710
No se puede obtener acceso al filtro de ramificaciones en la consola de CodeBuild	711
No se puede ver si la compilación se ha realizado correctamente o no	712
No se ha notificado el estado de la compilación al proveedor de fuentes	712
No se puede encontrar ni seleccionar la imagen base de la plataforma Windows Server Core 2019	712
Comandos anteriores de los archivos buildspec no reconocidos por comandos más recientes	713
Error: "Access denied" (Acceso denegado) al intentar descargar la caché	713
Aparece el error "BUILD_CONTAINER_UNABLE_TO_PULL_IMAGE" al usar una imagen de compilación personalizada	714
Error: «Se encontró que el contenedor de compilación estaba inactivo antes de completar la compilación. El contenedor de compilación murió porque no tenía memoria suficiente o la imagen de Docker no es compatible. ErrorCode: 500"	715
Error: "Cannot connect to the Docker daemon" (No se puede conectar al demonio de Docker) al ejecutar una compilación	716
Error: "no CodeBuild está autorizado a ejecutar: sts:AssumeRole" al crear o actualizar un proyecto de compilación	717
Error: «Error al llamar GetBucketAcl: el propietario del bucket ha cambiado o el rol de servicio ya no tiene permiso para llamar a s3:GetBucketAcl»	718
Error: "Failed to upload artifacts: Invalid arn (Error al cargar artefactos: arn no válido)" al ejecutar una compilación	718

Error: "Error de clon de Git: no es posible acceder a 'your-repository-URL ': problema con el certificado de SSL: certificado autofirmado"	718
Error: "The bucket you are attempting to access must be addressed using the specified endpoint (El bucket al que intenta obtener acceso debe direccionarse utilizando el punto de conexión especificado)" al ejecutar una compilación	719
Error: "Esta imagen de compilación requiere seleccionar al menos una versión de tiempo de ejecución".	720
Error: aparecer el error "QUEUED: INSUFFICIENT_SUBNET" cuando se produce un error en una compilación de una cola de compilaciones	721
Error: «No se pudo descargar la caché RequestError: se produjo un error en el envío de la solicitud debido a: x509: no se pudieron cargar las raíces del sistema y no se proporcionaron las raíces»	721
Error: «No se puede descargar el certificado de S3. AccessDenied»	722
Error: "Unable to locate credentials" (No se encuentran credenciales)	722
RequestError error de tiempo de espera cuando se ejecuta CodeBuild en un servidor proxy ...	724
El shell de Bourne (sh) debe existir en las imágenes de compilación	725
Advertencia: "Skipping install of runtimes. runtime version selection is not supported by this build image" (Si se omite la instalación de los entornos de ejecución, no se podrá seleccionar la versión del entorno de ejecución en esta imagen de compilación) al ejecutar una compilación	726
Error: «No se puede verificar JobWorker la identidad»	726
No se ha podido iniciar la compilación	726
Acceder a GitHub los metadatos en compilaciones almacenadas en caché local	727
AccessDenied: El propietario del bucket del grupo de informes no coincide con el propietario del bucket de S3...	727
Cuotas	728
Cuotas de servicio	728
Otros límites	733
Proyectos de compilación	733
Compilaciones	734
Flotas de computación	734
Informes	736
Etiquetas	736
Avisos de terceros AWS CodeBuild para Windows	738
1) Imagen de Docker base - windowsservercore	738
2) Imagen de Docker base de Windows - Choco	740

3) Imagen de Docker base de Windows - git --versión 2.16.2	740
4) Imagen de Docker basada en Windows: versión 15.0.26320.2 microsoft-build-tools	740
5) Imagen de Docker base de Windows - nuget.commandline --version 4.5.1	745
7) Imagen de Docker base de Windows - netfx-4.6.2-devpack	745
8) Imagen de Docker base de Windows - visualfsharpools, v 4.0	747
9) Imagen de Docker basada en Windows: -4.6 netfx-pcl-reference-assemblies	747
10) Imagen de Docker base de Windows - visualcppbuildtools, v 14.0.25420.1	751
11) Imagen de Docker basada en Windows: 3-ondemand-package.cab microsoft-windows-netfx	755
12) Imagen de Docker base de Windows - dotnet-sdk	756
Historial de documentos	758
Actualizaciones anteriores	777
Glosario de AWS	791
.....	dccxcii

¿Qué es AWS CodeBuild?

AWS CodeBuild es un servicio de construcción totalmente gestionado en la nube. CodeBuild compila el código fuente, ejecuta pruebas unitarias y produce artefactos listos para su despliegue. CodeBuild elimina la necesidad de aprovisionar, administrar y escalar sus propios servidores de compilación. Este servicio proporciona entornos de compilación preconfigurados para lenguajes de programación y herramientas de compilación populares como Apache Maven, Gradle, etc. También puede personalizar los entornos de compilación CodeBuild para usar sus propias herramientas de compilación. CodeBuild se escala automáticamente para cumplir con los picos de demanda de construcción.

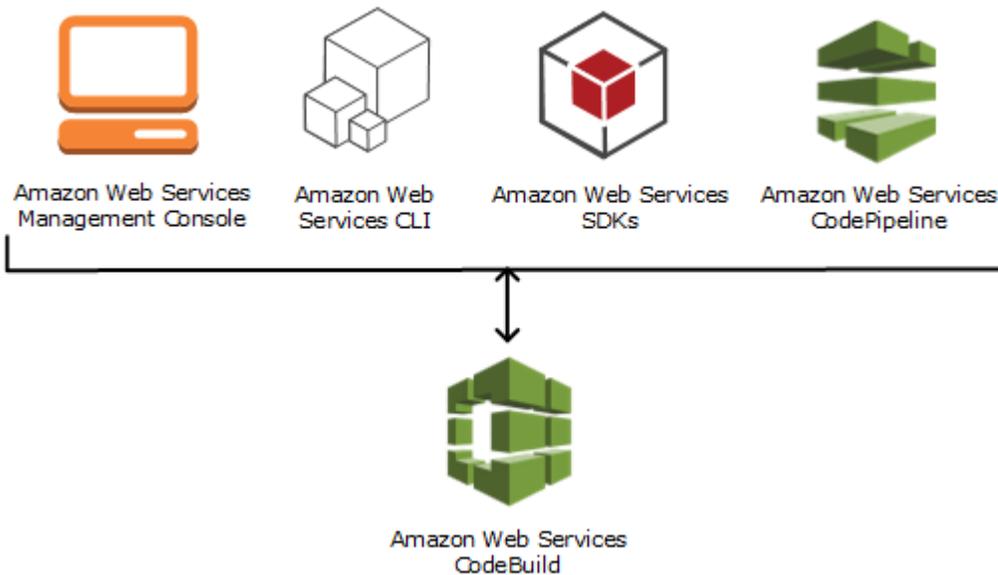
CodeBuild ofrece las siguientes ventajas:

- **Totalmente administrado:** CodeBuild elimina la necesidad de configurar, aplicar parches, actualizar y administrar sus propios servidores de compilación.
- **Bajo demanda:** se CodeBuild escala según demanda para satisfacer sus necesidades de construcción. Solo se paga por el número de minutos de compilación consumidos.
- **Listo para usar:** CodeBuild proporciona entornos de compilación preconfigurados para los lenguajes de programación más populares. Lo único que tiene que hacer es apuntar a su script de compilación para iniciar su primera compilación.

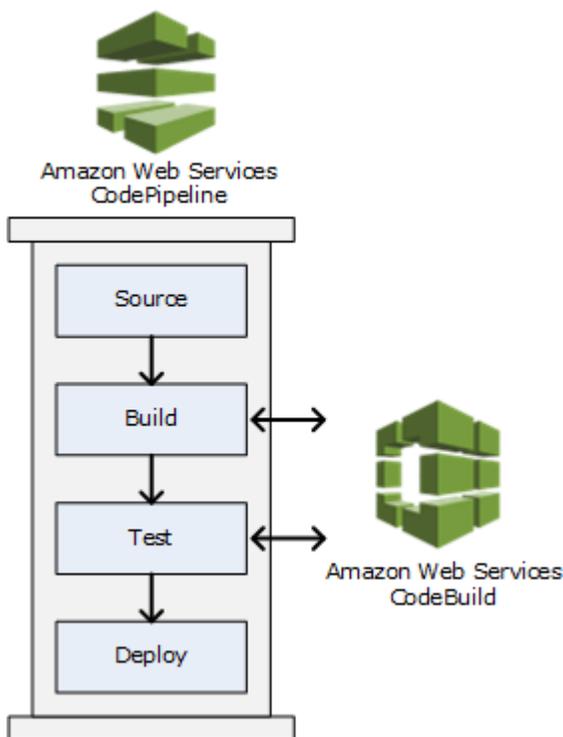
Para obtener más información, consulte [AWS CodeBuild](#).

¿Cómo se ejecuta CodeBuild

Puede utilizar la consola de AWS CodePipeline o AWS CodeBuild para ejecutar CodeBuild. También puede automatizar la ejecución de CodeBuild mediante AWS Command Line Interface (AWS CLI) o los AWS SDK.



Como se muestra en el siguiente diagrama, puedes añadirla CodeBuild como acción de compilación o prueba a la fase de creación o prueba de una canalización. AWS CodePipeline es un servicio de entrega continua que puedes usar para modelar, visualizar y automatizar los pasos necesarios para publicar tu código. Esto incluye la compilación de su código. Una canalización es una construcción de flujo de trabajo que describe la forma en la que se someten los cambios en el código a un proceso de lanzamiento.



Para usarlo CodePipeline para crear una canalización y, después, añadir una acción de CodeBuild compilación o prueba, consulte [Úselo CodePipeline con CodeBuild](#). Para obtener más información al respecto CodePipeline, consulte la [Guía AWS CodePipeline del usuario](#).

La CodeBuild consola también proporciona una forma de buscar rápidamente sus recursos, como repositorios, proyectos de creación, aplicaciones de implementación y canalizaciones. Seleccione Ir al recurso o pulse la tecla / y escriba el nombre del recurso. Se muestran todas las coincidencias en la lista. En las búsquedas, no se distingue entre mayúsculas y minúsculas. Solo puede ver los recursos para los que tiene permiso. Para obtener más información, consulte [Visualización de recursos en la consola](#).

Precios para CodeBuild

Para obtener más información, consulte [CodeBuild los precios](#).

¿Cómo puedo empezar CodeBuild?

Le recomendamos que siga los pasos que se describen a continuación:

1. Obtenga más información CodeBuild leyendo la información en [Conceptos](#).
2. Experimente con CodeBuild un escenario de ejemplo siguiendo las instrucciones de [Primeros pasos con la consola](#).
3. CodeBuild Úselo en sus propios escenarios siguiendo las instrucciones de [Planificación de una compilación](#).

Conceptos de AWS CodeBuild

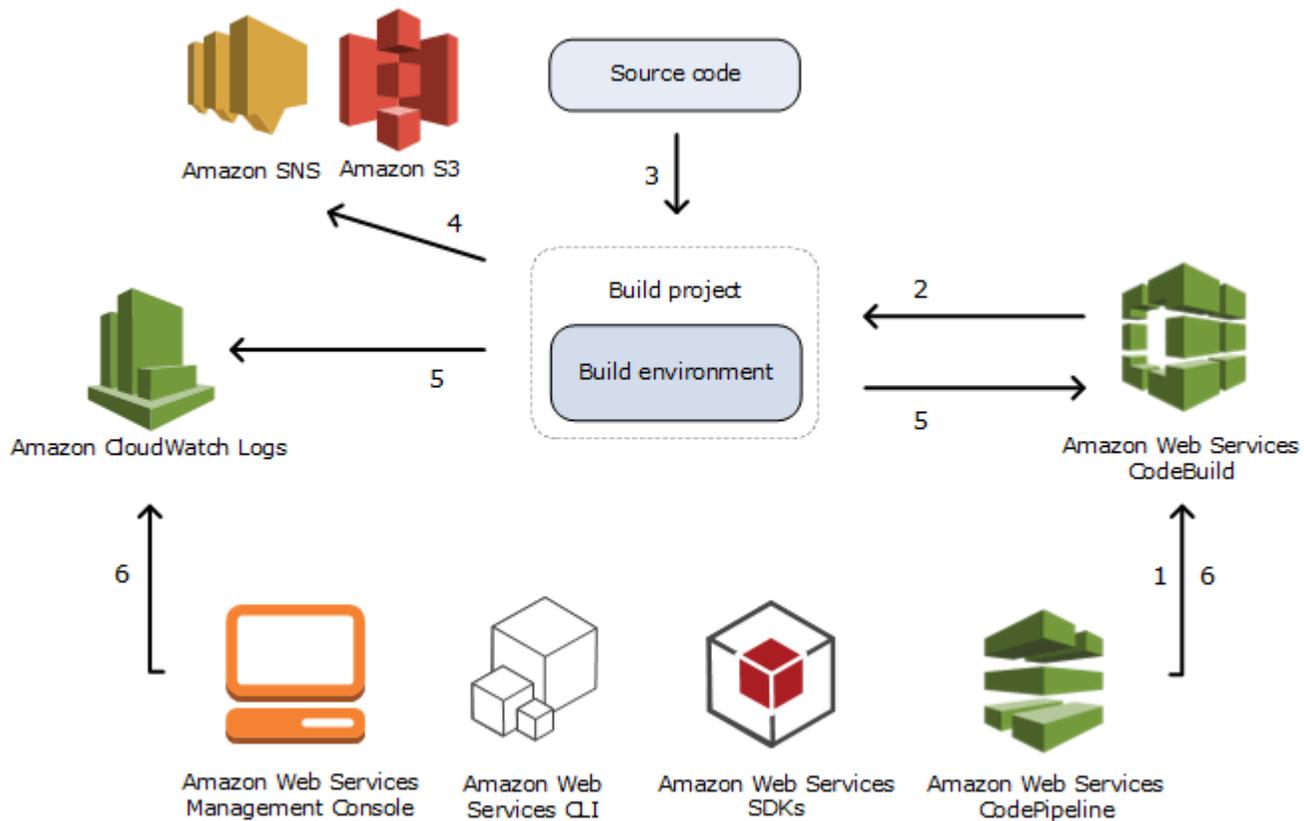
Los siguientes conceptos son importantes para comprender cómo funciona CodeBuild.

Temas

- [Cómo funciona CodeBuild](#)
- [Pasos siguientes](#)

Cómo funciona CodeBuild

En el diagrama siguiente se muestra lo que ocurre al ejecutar una compilación con CodeBuild:



- Como entrada, debe proporcionar a CodeBuild un proyecto de compilación. Un proyecto de compilación incluye información sobre cómo ejecutar una compilación, incluido dónde obtener el código fuente, qué entorno de compilación se debe usar, qué comandos de compilación se deben ejecutar y dónde se debe almacenar el resultado de la compilación. Un entorno de compilación representa una combinación de sistema operativo, tiempo de ejecución del lenguaje de programación y herramientas que CodeBuild utiliza para ejecutar una compilación. Para obtener más información, consulte:
 - [Creación de un proyecto de compilación](#)
 - [Referencia de entornos de compilación](#)
- CodeBuild utiliza el proyecto de compilación para crear el entorno de compilación.
- CodeBuild descarga el código fuente en el entorno de compilación y, a continuación, utiliza la especificación de compilación, tal y como se define en el proyecto de compilación, o incluida directamente en el código fuente. Una especificación de compilación es una colección de comandos de compilación y opciones de configuración relacionadas, en formato YAML, que CodeBuild utiliza para ejecutar una compilación. Para obtener más información, consulte [Referencia de la especificación de compilación](#).

4. Si existe alguna salida de la compilación, el entorno de compilación la carga en un bucket de S3. El entorno de compilación también puede realizar las tareas que indique en la especificación de compilación (por ejemplo, enviar notificaciones de compilación a un tema de Amazon SNS). Para ver un ejemplo, consulte [Ejemplo de notificaciones de compilación](#).
5. Mientras se ejecuta la compilación, el entorno de compilación envía información a CodeBuild y Registros de Amazon CloudWatch.
6. Mientras la compilación se está ejecutando, puede usar la consola de AWS CodeBuild, AWS CLI o los SDK de AWS para obtener información resumida sobre la compilación de e información detallada sobre la compilación de Registros de Amazon CloudWatch. Si utiliza AWS CodePipeline para ejecutar compilaciones, puede obtener información limitada sobre la compilación de CodePipeline.

Pasos siguientes

Ahora que ya sabe más sobre AWS CodeBuild, le recomendamos que siga estos pasos:

1. Experimente con CodeBuild en un escenario de ejemplo siguiendo las instrucciones de [Primeros pasos con la consola](#).
2. Utilice CodeBuild en sus propios escenarios siguiendo las instrucciones de [Planificación de una compilación](#).

Introducción a CodeBuild

En los siguientes tutoriales, utilizará AWS CodeBuild para crear una colección de archivos de entrada de código fuente de muestra en una versión implementable del código fuente.

Ambos tutoriales tienen la misma entrada y los mismos resultados, pero uno usa la consola de AWS CodeBuild y el otro, la AWS CLI.

Important

No es conveniente que utilice la cuenta raíz de AWS para completar este tutorial.

Empezar a trabajar con AWS CodeBuild utilizando la consola

En este tutorial, utilizará AWS CodeBuild para compilar una colección de archivos de entrada de código fuente de muestra (artefactos de entrada de compilación o entrada de compilación) en una versión desplegable del código fuente (artefacto de salida de compilación o salida de compilación). En concreto, se indica CodeBuild que se utilice Apache Maven, una herramienta de compilación común, para crear un conjunto de archivos de clases de Java en un archivo de Java Archive (JAR). No necesita estar familiarizado con Apache Maven o Java para completar este tutorial.

Puede trabajar con ellos CodeBuild a través de la CodeBuild consola AWS CodePipelineAWS CLI, el o los AWS SDK. En este tutorial se muestra cómo utilizar la CodeBuild consola. Para obtener más información sobre del uso de CodePipeline, consulte [Úselo CodePipeline con CodeBuild](#).

Important

Los pasos de este tutorial requieren que cree recursos (por ejemplo, un bucket de S3) que podrían generar cargos en su cuenta de AWS. Estos incluyen posibles cargos por CodeBuild y por AWS los recursos y acciones relacionados con Amazon S3 y CloudWatch Logs. AWS KMS Para obtener más información, consulte [AWS CodeBuildprecios](#), [precios de Amazon S3](#), [AWS Key Management Serviceprecios](#) y [CloudWatch precios de Amazon](#).

Pasos

- [Paso 1: Crear el código fuente](#)

- [Paso 2: Crear el archivo de especificación de compilación](#)
- [Paso 3: Crear dos buckets de S3](#)
- [Paso 4: Cargar el código fuente y el archivo buildspec](#)
- [Paso 5: Crear el proyecto de compilación](#)
- [Paso 6: Ejecutar la compilación](#)
- [Paso 7: Ver información resumida sobre la compilación](#)
- [Paso 8: Ver información detallada sobre la compilación](#)
- [Paso 9: Obtener el artefacto de salida de la compilación](#)
- [Paso 10: Eliminar los buckets de S3](#)
- [Encapsulación](#)

Paso 1: Crear el código fuente

(Parte de: [Empezar a trabajar con AWS CodeBuild utilizando la consola](#))

En este paso, crea el código fuente que CodeBuild desea incluir en el segmento de salida. Este código fuente se compone de dos archivos de clases Java y un archivo Apache Maven Object Model (POM).

1. En un directorio vacío del equipo o la instancia local, cree esta estructura de directorios.

```
(root directory name)
  |-- src
    |-- main
    |   |-- java
    |-- test
    |   |-- java
```

2. Con el editor de texto que desee, cree este archivo, asígnele el nombre `MessageUtil.java` y guárdelo en el directorio `src/main/java`.

```
public class MessageUtil {
    private String message;

    public MessageUtil(String message) {
        this.message = message;
    }
}
```

```
public String printMessage() {
    System.out.println(message);
    return message;
}

public String salutationMessage() {
    message = "Hi!" + message;
    System.out.println(message);
    return message;
}
}
```

Este archivo de clases crea como resultado la cadena de caracteres que se la ha pasado. El constructor `MessageUtil` establece la cadena de caracteres. El método `printMessage` crea la salida. El método `salutationMessage` muestra `Hi!` seguido de la cadena de caracteres.

3. Cree este archivo, asígnele el nombre `TestMessageUtil.java` y guárdelo en el directorio `/src/test/java`.

```
import org.junit.Test;
import org.junit.Ignore;
import static org.junit.Assert.assertEquals;

public class TestMessageUtil {

    String message = "Robert";
    MessageUtil messageUtil = new MessageUtil(message);

    @Test
    public void testPrintMessage() {
        System.out.println("Inside testPrintMessage()");
        assertEquals(message,messageUtil.printMessage());
    }

    @Test
    public void testSalutationMessage() {
        System.out.println("Inside testSalutationMessage()");
        message = "Hi!" + "Robert";
        assertEquals(message,messageUtil.salutationMessage());
    }
}
```

Este archivo de clases establece la variable `message` de la clase `MessageUtil` en `Robert`. A continuación, comprueba si la variable `message` se ha establecido correctamente comprobando si las cadenas `Robert` y `Hi!Robert` aparecen en la salida.

4. Cree este archivo, asígnele el nombre `pom.xml` y guárdelo en el directorio raíz (nivel superior).

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/
maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>org.example</groupId>
  <artifactId>messageUtil</artifactId>
  <version>1.0</version>
  <packaging>jar</packaging>
  <name>Message Utility Java Sample App</name>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.11</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
  <build>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-compiler-plugin</artifactId>
        <version>3.8.0</version>
      </plugin>
    </plugins>
  </build>
</project>
```

Apache Maven utiliza las instrucciones de este archivo para convertir los archivos `MessageUtil.java` y `TestMessageUtil.java` en un archivo denominado `messageUtil-1.0.jar` y, a continuación, ejecuta las pruebas especificadas.

En este punto, la estructura de directorios debería ser similar a la siguiente.

```
(root directory name)
|-- pom.xml
`-- src
    |-- main
    |   |-- java
    |       |-- MessageUtil.java
    |-- test
    |   |-- java
    |       |-- TestMessageUtil.java
```

Siguiente paso

[Paso 2: Crear el archivo de especificación de compilación](#)

Paso 2: Crear el archivo de especificación de compilación

(Paso anterior: [Paso 1: Crear el código fuente](#))

En este paso, creará un archivo de especificación de compilación. Una especificación de compilación es un conjunto de comandos de compilación y configuraciones relacionadas, en formato YAML, que se CodeBuild utiliza para ejecutar una compilación. Sin una especificación de compilación, CodeBuild no se puede convertir correctamente la entrada de compilación en el resultado de la compilación ni localizar el artefacto de salida de la compilación en el entorno de compilación para cargarlo en el depósito de salida.

Cree este archivo, asígnele el nombre `buildspec.yml` y guárdelo en el directorio raíz (nivel superior).

```
version: 0.2

phases:
  install:
    runtime-versions:
      java: corretto11
  pre_build:
    commands:
      - echo Nothing to do in the pre_build phase...
  build:
    commands:
      - echo Build started on `date`
      - mvn install
  post_build:
```

```
  commands:
    - echo Build completed on `date`
artifacts:
  files:
    - target/messageUtil-1.0.jar
```

Important

Como una declaración de especificación de compilación debe ser una declaración YAML válida, los espacios de la declaración son importantes. Si el número de espacios de la declaración de especificación de compilación no coincide con esta declaración, la compilación podría producir un error inmediatamente. Puede utilizar un validador de YAML para comprobar si la declaración de especificación de compilación es una declaración YAML válida.

Note

En lugar de incluir un archivo de especificación de compilación en el código fuente, puede declarar los comandos de la compilación por separado cuando cree un proyecto de compilación. Esto resulta útil si desea compilar el código fuente con diferentes comandos de compilación sin actualizar el código fuente del repositorio cada vez. Para obtener más información, consulte [Sintaxis de buildspec](#).

En esta declaración de especificación de compilación:

- `version` representa la versión del estándar de especificación de compilación que se va a usar. Esta declaración de especificación de compilación usa la última versión, `0.2`.
- `phases` representa las fases de compilación en las que puede indicar a CodeBuild que ejecute comandos. Estas fases de compilación se muestran aquí como `install`, `pre_build`, `build` y `post_build`. No puede cambiar los nombres de estas fases de compilación ni puede crear nombres de fases de compilación adicionales.

En este ejemplo, durante la `build` fase, CodeBuild ejecuta el `mvn install` comando. Este comando indica a Apache Maven que compile, pruebe y empaquete los archivos de clases Java compilados en un artefacto de salida de la compilación. En aras de una mayor exhaustividad, se incluyen algunos comandos `echo` en cada fase de compilación de este ejemplo. Cuando vea

información de compilación detallada más adelante en este tutorial, la salida de estos comandos de echo puede ayudarlo a comprender mejor cómo CodeBuild ejecuta los comandos y en qué orden lo hace. (Aunque en este ejemplo se incluyen todas las fases de compilación, no es necesario que incluya una fase de compilación si no piensa ejecutar ningún comando durante esa fase). Para cada fase de compilación, CodeBuild ejecuta cada comando especificado, uno a la vez, en el orden indicado, de principio a fin.

- `artifacts` representa el conjunto de artefactos de salida de la compilación que se CodeBuild carga en el depósito de salida. `files` representa los archivos que se van a incluir en el resultado de la compilación. CodeBuild carga el único `messageUtil-1.0.jar` archivo que se encuentra en el directorio `target` relativo del entorno de compilación. El nombre del archivo `messageUtil-1.0.jar` y el nombre del directorio `target` se basan en la forma en que Apache Maven crea y almacena los artefactos de salida de la compilación para este ejemplo únicamente. En sus propias compilaciones, estos nombres de archivos y directorios son diferentes.

Para obtener más información, consulte [Referencia de la especificación de compilación](#).

En este punto, la estructura de directorios debería ser similar a la siguiente.

```
(root directory name)
|-- pom.xml
|-- buildspec.yml
`-- src
    |-- main
    |   |-- java
    |       |-- MessageUtil.java
    |-- test
    |   |-- java
    |       |-- TestMessageUtil.java
```

Siguiente paso

[Paso 3: Crear dos buckets de S3](#)

Paso 3: Crear dos buckets de S3

(Paso anterior: [Paso 2: Crear el archivo de especificación de compilación](#))

Aunque puede usar un solo bucket para este tutorial, dos buckets hacen que sea más fácil ver de dónde proviene la entrada de compilación y hacia dónde va la salida de compilación.

- Uno de estos buckets (bucket de entrada) almacena la entrada de compilación. En este tutorial, el nombre de este depósito de entrada es `codebuild-region-ID-account-ID-input-bucket`, donde *region-ID* es la región de AWS del bucket y *account-ID* es su ID de cuenta de AWS.
- El otro bucket (bucket de salida) almacena la salida de la compilación. En este tutorial, el nombre de este bucket de salida es `codebuild-region-ID-account-ID-output-bucket`.

Si elige nombres diferentes para estos buckets, asegúrese de usarlos en este tutorial.

Estos dos buckets están en la misma región de AWS que sus compilaciones. Por ejemplo, si indicas CodeBuild que se ejecute una compilación en la región EE.UU. Este (Ohio), estos depósitos también deben estar en la región EE.UU. Este (Ohio).

Para obtener más información, consulte [Creación de un bucket](#) en la Guía del usuario de Amazon Simple Storage Service.

Note

Aunque CodeBuild también es compatible con las entradas de compilación almacenadas en CodeCommit los repositorios de Bitbucket y en los repositorios, en este tutorial no se muestra cómo utilizarlas. GitHub Para obtener más información, consulte [Planificación de una compilación](#).

Siguiente paso

[Paso 4: Cargar el código fuente y el archivo buildspec](#)

Paso 4: Cargar el código fuente y el archivo buildspec

(Paso anterior: [Paso 3: Crear dos buckets de S3](#))

En este paso, añadirá el código fuente y el archivo de especificación de compilación al bucket de entrada.

Con la utilidad zip del sistema operativo, cree un archivo llamado `MessageUtil.zip` que incluya `MessageUtil.java`, `TestMessageUtil.java`, `pom.xml` y `buildspec.yml`.

La estructura de directorios del archivo `MessageUtil.zip` debe ser similar a la siguiente.

```
MessageUtil.zip
|-- pom.xml
|-- buildspec.yml
`-- src
    |-- main
    |   |-- java
    |   |-- MessageUtil.java
    |-- test
    |   |-- java
    |   |-- TestMessageUtil.java
```

 Important

No incluya el directorio (*root directory name*), solo los directorios y archivos del directorio (*root directory name*).

Cargue el archivo MessageUtil.zip identificado en el bucket de entrada denominado codebuild-*region-ID-account-ID*-input-bucket.

 Important

CodeCommit En GitHub el caso de los repositorios y de Bitbucket, por convención, debes almacenar un archivo de especificaciones de compilación denominado buildspec.yml en la raíz (nivel superior) de cada repositorio o incluir la declaración de especificaciones de compilación como parte de la definición del proyecto de compilación. No cree un archivo ZIP que contenga el código fuente del repositorio y el archivo de especificación de compilación. Solo para entradas de compilación almacenadas en buckets de S3, debe crear un archivo ZIP que contenga el código fuente y convencionalmente, un archivo de especificación de compilación denominado buildspec.yml en la raíz (nivel superior) o incluya la declaración de especificaciones de compilación como parte de la definición del proyecto de compilación. Si desea utilizar otro nombre para el archivo de especificación de compilación, o si quiere hacer referencia a una especificación de compilación en una ubicación distinta de la raíz, puede especificar una invalidación de la especificación de compilación como parte de la definición del proyecto de compilación. Para obtener más información, consulte [Nombre de archivo y ubicación de almacenamiento de buildspec](#).

Siguiente paso

[Paso 5: Crear el proyecto de compilación](#)

Paso 5: Crear el proyecto de compilación

(Paso anterior: [Paso 4: Cargar el código fuente y el archivo buildspec](#))

En este paso, creará un proyecto de compilación que usará AWS CodeBuild para ejecutar la compilación. Un proyecto de compilación incluye información sobre cómo ejecutar una compilación, incluido dónde obtener el código fuente, qué entorno de compilación se debe usar, qué comandos de compilación se deben ejecutar y dónde se debe almacenar el resultado de la compilación. Un entorno de compilación representa una combinación de sistema operativo, lenguaje de programación, tiempo de ejecución y herramientas que se CodeBuild utilizan para ejecutar una compilación. El entorno de compilación se expresa como una imagen de Docker. Para obtener más información, consulte [Descripción general de Docker](#) en la página de documentos de Docker.

Para este entorno de compilación, se indica CodeBuild que se utilice una imagen de Docker que contenga una versión del kit de desarrollo de Java (JDK) y Apache Maven.

Para crear el proyecto de compilación

1. Inicie sesión en la AWS Management Console y abra la consola de AWS CodeBuild en <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Usa el selector de AWS regiones para elegir una AWS región que sea compatible. CodeBuild Para obtener más información, consulte [Puntos de conexión y cuotas de AWS CodeBuild](#) en la Referencia general de Amazon Web Services.
3. Si se muestra una página de CodeBuild información, elija Crear proyecto de construcción. De lo contrario, en el panel de navegación, expanda Compilar, elija Proyectos de compilación y, a continuación, elija Crear proyecto de compilación.
4. En la página Create build project (Crear proyecto de compilación), en Project configuration (Configuración del proyecto), en Project name (Nombre de proyecto), escriba un nombre para este proyecto de compilación (en este ejemplo, `codebuild-demo-project`). Los nombres de los proyectos de compilación debe ser únicos en cada cuenta de AWS. Si elige otro nombre, asegúrese de utilizarlo durante todo el tutorial.

Note

En la página Create build project (Crear proyecto de compilación), podría aparecer un mensaje de error similar al siguiente: You are not authorized to perform this operation (No tiene autorización para realizar esta operación). Lo más probable es que haya iniciado sesión en AWS Management Console como un usuario que no tiene permiso para crear proyectos de compilación. Para solucionar este problema, cierre la sesión de la AWS Management Console y, a continuación, inicie sesión con credenciales que pertenezcan a una de las siguientes entidades de IAM:

- Un usuario administrador en su cuenta de AWS. Para obtener más información, consulte [Creación del primer usuario raíz y grupo de Cuenta de AWS](#) en la Guía del usuario.
- Un usuario en su cuenta de AWS con las políticas administradas `AWSCodeBuildAdminAccess`, `AmazonS3ReadOnlyAccess` y `IAMFullAccess` vinculadas a ese usuario o un grupo de IAM al que pertenece el usuario. Si no dispone de un usuario o grupo de AWS en su cuenta de con estos permisos, y no puede añadir estos permisos al usuario o grupo, póngase en contacto con el administrador de su cuenta de AWS para obtener ayuda. Para obtener más información, consulte [AWS políticas gestionadas \(predefinidas\) para AWS CodeBuild](#).

Ambas opciones incluyen permisos de administrador que le permiten crear un proyecto de compilación y completar este tutorial. Le recomendamos que utilice siempre los permisos mínimos necesarios para realizar la tarea. Para obtener más información, consulte [AWS CodeBuild referencia de permisos](#).

5. En Fuente, como Proveedor de código fuente, elija Amazon S3.
6. En Bucket, elija `codebuild-region-ID-account-ID-input-bucket`.
7. En S3 object key (Clave de objeto de S3), escriba **MessageUtil.zip**.
8. En Environment (Entorno), para Environment image (Imagen de entorno), deje Managed image (Imagen administrada) seleccionado.
9. En Operating system (Sistema operativo), elija Amazon Linux 2.
10. En Runtime(s) (Tiempo de ejecución), elija Standard (Estándar).
11. En Imagen, elija `aws/codebuild/amazonlinux2-x86_64-standard:4.0`.

12. En Service role (Rol de servicio), deje la opción New service role (Nuevo rol de servicio) seleccionada y no haga ningún cambio en Role name (Nombre de rol).
13. En Buildspec, deje Use a buildspec file (Usar un archivo buildspec) seleccionado.
14. En Artefactos, como Tipo, seleccione Amazon S3.
15. En Nombre de bucket, elija codebuild-**regon-ID-account-ID**-output-bucket.
16. Deje Name (Nombre) y Path (Ruta) en blanco.
17. Elija Crear el proyecto de compilación.

Siguiente paso

[Paso 6: Ejecutar la compilación](#)

Paso 6: Ejecutar la compilación

(Paso anterior: [Paso 5: Crear el proyecto de compilación](#))

En este paso, indicará a AWS CodeBuild que ejecute la compilación con la configuración del proyecto de compilación.

Para ejecutar la compilación

1. Abra la consola de AWS CodeBuild en <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. En el panel de navegación, elija Proyectos de compilación.
3. En la lista de proyectos de construcción, elija y codebuild-demo-project, a continuación, elija Iniciar compilación. La compilación comienza inmediatamente.

Siguiente paso

[Paso 7: Ver información resumida sobre la compilación](#)

Paso 7: Ver información resumida sobre la compilación

(Paso anterior: [Paso 6: Ejecutar la compilación](#))

En este paso, verá información resumida sobre el estado de la compilación.

Para ver la información resumida de la compilación

1. Si <build-ID>no se muestra la página codebuild-demo-project:, en la barra de navegación, seleccione Crear historial. A continuación, en la lista de proyectos de construcción, para Proyecto, elija el enlace Construir y ejecutar para codebuild-demo-project. Debe haber un solo enlace coincidente. (Si ya ha completado este tutorial antes, elija el vínculo con el valor más reciente en la columna Completed (Completado)).
2. En la página Estado de compilación, en Detalles de fase, se deben mostrar las fases de compilación siguientes como Con éxito en la columna Estado:
 - SUBMITTED
 - QUEUED
 - PROVISIONING
 - DOWNLOAD_SOURCE
 - INSTALL
 - PRE_BUILD
 - BUILD
 - POST_BUILD
 - UPLOAD_ARTIFACTS
 - FINALIZING
 - COMPLETED

En Build Status (Estado de la compilación), debería mostrarse Succeeded (Realizado correctamente).

Si en su lugar aparece In Progress (En curso) elija el botón de actualizar.

3. Junto a cada fase de compilación, el valor Duration (Duración) indica cuánto tiempo ha tardado la fase de compilación. El valor End time (Hora de finalización) indica que esa fase de compilación ha terminado.

Siguiente paso

[Paso 8: Ver información detallada sobre la compilación](#)

Paso 8: Ver información detallada sobre la compilación

(Paso anterior: [Paso 7: Ver información resumida sobre la compilación](#))

En este paso, verás información detallada sobre tus CloudWatch registros de compilación.

Note

Para proteger la información confidencial, los CodeBuild registros ocultan lo siguiente:

- ID de clave de acceso de AWS. Para obtener más información, consulte [Administración de claves de acceso para usuarios de IAM](#) en la Guía del usuario de AWS Identity and Access Management.
- Cadenas especificadas mediante el almacén de parámetros. Para obtener más información, consulte [Almacén de parámetros de Systems Manager](#) y [Tutorial de la consola del almacén de parámetros de Systems Manager](#) en la Guía del usuario de Amazon EC2 Systems Manager.
- Cadenas especificadas mediante AWS Secrets Manager. Para obtener más información, consulte [Administración de claves](#).

Para ver información detallada sobre la compilación

1. Con la página de detalles de la compilación mostrada del paso anterior, se muestran las últimas 10,000 líneas del log de compilación en Build logs. Para ver el registro de compilación completo en CloudWatch Logs, selecciona el enlace Ver todo el registro.
2. En el flujo de registro de CloudWatch registros, puedes buscar los eventos del registro. De forma predeterminada, solo se muestra el último conjunto de eventos de log. Para ver eventos de log anteriores, desplácese hasta el principio de la lista.
3. En este tutorial, la mayoría de los eventos de registro contienen información detallada sobre la descarga e instalación de archivos de dependencia de compilación de CodeBuild en su entorno de compilación, lo que probablemente no le interese. Puede usar el cuadro Filter events para reducir la información que se muestra. Por ejemplo, si escribe "[INFO]" en el cuadro Filter events (Filtrar eventos), solo se muestran los eventos que contienen [INFO]. Para obtener más información, consulte [Sintaxis de filtros y patrones](#) en la Guía del CloudWatch usuario de Amazon.

Siguiente paso

[Paso 9: Obtener el artefacto de salida de la compilación](#)

Paso 9: Obtener el artefacto de salida de la compilación

(Paso anterior: [Paso 8: Ver información detallada sobre la compilación](#))

En este paso, obtendrá el `messageUtil-1.0.jar` archivo que se CodeBuild creó y se cargó en el depósito de salida.

Puede utilizar la CodeBuild consola o la consola Amazon S3 para completar este paso.

Para obtener el artefacto de salida de la compilación (consola de AWS CodeBuild)

1. Con la CodeBuild consola aún abierta y la página de detalles de la compilación del paso anterior, selecciona la pestaña Detalles de la construcción y desplázate hacia abajo hasta la sección Artefactos.

Note

Si la página de detalles de la compilación no se muestra, en la barra de navegación, elija Historial de compilación y, a continuación, elija el enlace Ejecución de la compilación.

2. El enlace a la carpeta Amazon S3 se encuentra en la Ubicación de carga de artefactos. Este enlace abre la carpeta de Amazon S3 donde se encuentra el archivo de artefactos de salida de compilación de `messageUtil-1.0.jar`.

Para obtener el artefacto de salida de la compilación (consola de Amazon S3)

1. Abra la consola de Amazon S3 en <https://console.aws.amazon.com/s3>.
2. Abra `codebuild-region-ID-account-ID-output-bucket`.
3. Abra la carpeta `codebuild-demo-project`.
4. Abra la carpeta `target`, donde se encuentra el archivo de artefactos de salida de compilación `messageUtil-1.0.jar`.

Siguiente paso

[Paso 10: Eliminar los buckets de S3](#)

Paso 10: Eliminar los buckets de S3

(Paso anterior: [Paso 9: Obtener el artefacto de salida de la compilación](#))

Para evitar cargos continuos en su cuenta de AWS, puede eliminar los buckets de entrada utilizados en este tutorial. Para obtener más instrucciones, consulte [Eliminar o vaciar un bucket](#) en la Guía del usuario de Amazon Simple Storage Service.

Si va a utilizar el usuario de IAM o un usuario administrador de IAM para eliminar estos buckets, el usuario debe tener más permisos de acceso. Añada la siguiente instrucción entre los marcadores (**### BEGIN ADDING STATEMENT HERE ###** y **### END ADDING STATEMENTS HERE ###**) a una política de acceso existente para el usuario.

Las elipsis (...) de esta declaración se usan para abreviar. No elimine las instrucciones de la política de acceso existente. No escriba estos puntos suspensivos en la política.

```
{
  "Version": "2012-10-17",
  "Id": "...",
  "Statement": [
    ### BEGIN ADDING STATEMENT HERE ###
    {
      "Effect": "Allow",
      "Action": [
        "s3:DeleteBucket",
        "s3:DeleteObject"
      ],
      "Resource": "*"
    }
    ### END ADDING STATEMENT HERE ###
  ]
}
```

Siguiente paso

[Encapsulación](#)

Encapsulación

En este tutorial, se ha usado AWS CodeBuild para construir un conjunto de archivos de clase Java en un archivo JAR. A continuación, ha consultado los resultados de la compilación.

Ahora puedes probar a utilizarla CodeBuild en tus propios escenarios. Siga las instrucciones en [Planificación de una compilación](#). Si piensa que aún no está preparado, tal vez desee intentar compilar algunos de los ejemplos. Para obtener más información, consulte [Muestras](#).

Introducción a AWS CodeBuild utilizando la AWS CLI

En este tutorial, usará AWS CodeBuild para compilar una selección de archivos de entrada de código fuente de muestra (llamados artefactos de entrada de la compilación o entrada de la compilación) en una versión implementable del código fuente (llamada artefacto de salida de compilación o salida de compilación). En concreto, se indica CodeBuild que se utilice Apache Maven, una herramienta de compilación común, para crear un conjunto de archivos de clases de Java en un archivo de Java Archive (JAR). No necesita estar familiarizado con Apache Maven o Java para completar este tutorial.

Puede trabajar con ellos CodeBuild a través de la CodeBuild consola AWS CodePipelineAWS CLI, el o los AWS SDK. En este tutorial se muestra cómo usarlo CodeBuild con. AWS CLI Para obtener información sobre el uso CodePipeline, consulte [Úselo CodePipeline con CodeBuild](#).

Important

Los pasos de este tutorial requieren que cree recursos (por ejemplo, un bucket de S3) que podrían generar cargos en su cuenta de AWS. Estos incluyen posibles cargos por CodeBuild y por AWS los recursos y acciones relacionados con Amazon S3 y CloudWatch Logs. AWS KMS Para obtener más información, consulte [CodeBuildprecios](#), [precios de Amazon S3](#), [AWS Key Management Serviceprecios](#) y [CloudWatch precios de Amazon](#).

Pasos

- [Paso 1: Crear el código fuente](#)
- [Paso 2: Crear el archivo de especificación de compilación](#)
- [Paso 3: Crear dos buckets de S3](#)

- [Paso 4: Cargar el código fuente y el archivo buildspec](#)
- [Paso 5: Crear el proyecto de compilación](#)
- [Paso 6: Ejecutar la compilación](#)
- [Paso 7: Ver información resumida sobre la compilación](#)
- [Paso 8: Ver información detallada sobre la compilación](#)
- [Paso 9: Obtener el artefacto de salida de la compilación](#)
- [Paso 10: Eliminar los buckets de S3](#)
- [Encapsulación](#)

Paso 1: Crear el código fuente

(Parte de: [Introducción a AWS CodeBuild utilizando la AWS CLI](#))

En este paso, crea el código fuente que CodeBuild desea incluir en el segmento de salida. Este código fuente se compone de dos archivos de clases Java y un archivo Apache Maven Object Model (POM).

1. En un directorio vacío del equipo o la instancia local, cree esta estructura de directorios.

```
(root directory name)
  |-- src
    |-- main
      |-- java
    |-- test
      |-- java
```

2. Con el editor de texto que desee, cree este archivo, asígnele el nombre `MessageUtil.java` y guárdelo en el directorio `src/main/java`.

```
public class MessageUtil {
    private String message;

    public MessageUtil(String message) {
        this.message = message;
    }

    public String printMessage() {
        System.out.println(message);
        return message;
    }
}
```

```
}

public String salutationMessage() {
    message = "Hi!" + message;
    System.out.println(message);
    return message;
}
}
```

Este archivo de clases crea como resultado la cadena de caracteres que se la ha pasado. El constructor `MessageUtil` establece la cadena de caracteres. El método `printMessage` crea la salida. El método `salutationMessage` muestra `Hi!` seguido de la cadena de caracteres.

3. Cree este archivo, asígnele el nombre `TestMessageUtil.java` y guárdelo en el directorio `/src/test/java`.

```
import org.junit.Test;
import org.junit.Ignore;
import static org.junit.Assert.assertEquals;

public class TestMessageUtil {

    String message = "Robert";
    MessageUtil messageUtil = new MessageUtil(message);

    @Test
    public void testPrintMessage() {
        System.out.println("Inside testPrintMessage()");
        assertEquals(message,messageUtil.printMessage());
    }

    @Test
    public void testSalutationMessage() {
        System.out.println("Inside testSalutationMessage()");
        message = "Hi!" + "Robert";
        assertEquals(message,messageUtil.salutationMessage());
    }
}
```

Este archivo de clases establece la variable `message` de la clase `MessageUtil` en `Robert`. A continuación, comprueba si la variable `message` se ha establecido correctamente comprobando si las cadenas `Robert` y `Hi!Robert` aparecen en la salida.

4. Cree este archivo, asígnele el nombre `pom.xml` y guárdelo en el directorio raíz (nivel superior).

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/
maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>org.example</groupId>
  <artifactId>messageUtil</artifactId>
  <version>1.0</version>
  <packaging>jar</packaging>
  <name>Message Utility Java Sample App</name>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.11</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
  <build>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-compiler-plugin</artifactId>
        <version>3.8.0</version>
      </plugin>
    </plugins>
  </build>
</project>
```

Apache Maven utiliza las instrucciones de este archivo para convertir los archivos `MessageUtil.java` y `TestMessageUtil.java` en un archivo denominado `messageUtil-1.0.jar` y, a continuación, ejecuta las pruebas especificadas.

En este punto, la estructura de directorios debería ser similar a la siguiente.

```
(root directory name)
|-- pom.xml
`-- src
    |-- main
    |   |-- java
```

```
|           |-- MessageUtil.java
|-- test
|           |-- java
|           |-- TestMessageUtil.java
```

Siguiente paso

[Paso 2: Crear el archivo de especificación de compilación](#)

Paso 2: Crear el archivo de especificación de compilación

(Paso anterior: [Paso 1: Crear el código fuente](#))

En este paso, creará un archivo de especificación de compilación. Una especificación de compilación es un conjunto de comandos de compilación y configuraciones relacionadas, en formato YAML, que se CodeBuild utiliza para ejecutar una compilación. Sin una especificación de compilación, CodeBuild no se puede convertir correctamente la entrada de compilación en el resultado de la compilación ni localizar el artefacto de salida de la compilación en el entorno de compilación para cargarlo en el depósito de salida.

Cree este archivo, asígnele el nombre `buildspec.yml` y guárdelo en el directorio raíz (nivel superior).

```
version: 0.2

phases:
  install:
    runtime-versions:
      java: corretto11
  pre_build:
    commands:
      - echo Nothing to do in the pre_build phase...
  build:
    commands:
      - echo Build started on `date`
      - mvn install
  post_build:
    commands:
      - echo Build completed on `date`
artifacts:
  files:
```

```
- target/messageUtil-1.0.jar
```

Important

Como una declaración de especificación de compilación debe ser una declaración YAML válida, los espacios de la declaración son importantes. Si el número de espacios de la declaración de especificación de compilación no coincide con esta declaración, la compilación podría producir un error inmediatamente. Puede utilizar un validador de YAML para comprobar si la declaración de especificación de compilación es una declaración YAML válida.

Note

En lugar de incluir un archivo de especificación de compilación en el código fuente, puede declarar los comandos de la compilación por separado cuando cree un proyecto de compilación. Esto resulta útil si desea compilar el código fuente con diferentes comandos de compilación sin actualizar el código fuente del repositorio cada vez. Para obtener más información, consulte [Sintaxis de buildspec](#).

En esta declaración de especificación de compilación:

- `version` representa la versión del estándar de especificación de compilación que se va a usar. Esta declaración de especificación de compilación usa la última versión, `0.2`.
- `phases` representa las fases de compilación en las que puede indicar a CodeBuild que ejecute comandos. Estas fases de compilación se muestran aquí como `install`, `pre_build`, `build` y `post_build`. No puede cambiar los nombres de estas fases de compilación ni puede crear nombres de fases de compilación adicionales.

En este ejemplo, durante la `build` fase, CodeBuild ejecuta el `mvn install` comando. Este comando indica a Apache Maven que compile, pruebe y empaquete los archivos de clases Java compilados en un artefacto de salida de la compilación. En aras de una mayor exhaustividad, se incluyen algunos comandos `echo` en cada fase de compilación de este ejemplo. Cuando vea información de compilación detallada más adelante en este tutorial, la salida de estos comandos de `echo` puede ayudarlo a comprender mejor cómo CodeBuild ejecuta los comandos y en qué orden lo hace. (Aunque en este ejemplo se incluyen todas las fases de compilación, no es

necesario que incluya una fase de compilación si no piensa ejecutar ningún comando durante esa fase). Para cada fase de compilación, CodeBuild ejecuta cada comando especificado, uno a la vez, en el orden indicado, de principio a fin.

- `artifacts` representa el conjunto de artefactos de salida de la compilación que se CodeBuild carga en el depósito de salida. `files` representa los archivos que se van a incluir en el resultado de la compilación. CodeBuild carga el único `messageUtil-1.0.jar` archivo que se encuentra en el directorio `target` relativo del entorno de compilación. El nombre del archivo `messageUtil-1.0.jar` y el nombre del directorio `target` se basan en la forma en que Apache Maven crea y almacena los artefactos de salida de la compilación para este ejemplo únicamente. En sus propias compilaciones, estos nombres de archivos y directorios son diferentes.

Para obtener más información, consulte [Referencia de la especificación de compilación](#).

En este punto, la estructura de directorios debería ser similar a la siguiente.

```
(root directory name)
|-- pom.xml
|-- buildspec.yml
`-- src
    |-- main
    |   |-- java
    |       |-- MessageUtil.java
    |-- test
    |   |-- java
    |       |-- TestMessageUtil.java
```

Siguiente paso

[Paso 3: Crear dos buckets de S3](#)

Paso 3: Crear dos buckets de S3

(Paso anterior: [Paso 2: Crear el archivo de especificación de compilación](#))

Aunque puede usar un solo bucket para este tutorial, dos buckets hacen que sea más fácil ver de dónde proviene la entrada de compilación y hacia dónde va la salida de compilación.

- Uno de estos buckets (bucket de entrada) almacena la entrada de compilación. En este tutorial, el nombre de este depósito de entrada es `codebuild-region-ID-account-ID-input-bucket`, donde *region-ID* es la región de AWS del bucket y *account-ID* es su ID de cuenta de AWS.

- El otro bucket (bucket de salida) almacena la salida de la compilación. En este tutorial, el nombre de este bucket de salida es `codebuild-region-ID-account-ID-output-bucket`.

Si elige nombres diferentes para estos buckets, asegúrese de usarlos en este tutorial.

Estos dos buckets están en la misma región de AWS que sus compilaciones. Por ejemplo, si indicas CodeBuild que se ejecute una compilación en la región EE.UU. Este (Ohio), estos depósitos también deben estar en la región EE.UU. Este (Ohio).

Para obtener más información, consulte [Creación de un bucket](#) en la Guía del usuario de Amazon Simple Storage Service.

Note

Aunque CodeBuild también es compatible con las entradas de compilación almacenadas en CodeCommit los repositorios de Bitbucket y en los repositorios, en este tutorial no se muestra cómo utilizarlas. GitHub Para obtener más información, consulte [Planificación de una compilación](#).

Siguiente paso

[Paso 4: Cargar el código fuente y el archivo buildspec](#)

Paso 4: Cargar el código fuente y el archivo buildspec

(Paso anterior: [Paso 3: Crear dos buckets de S3](#))

En este paso, añadirá el código fuente y el archivo de especificación de compilación al bucket de entrada.

Con la utilidad zip del sistema operativo, cree un archivo llamado `MessageUtil.zip` que incluya `MessageUtil.java`, `TestMessageUtil.java`, `pom.xml` y `buildspec.yml`.

La estructura de directorios del archivo `MessageUtil.zip` debe ser similar a la siguiente.

```
MessageUtil.zip
|-- pom.xml
|-- buildspec.yml
```

```
`-- src
  |-- main
  |   |-- java
  |   |-- MessageUtil.java
  |-- test
  |   |-- java
  |   |-- TestMessageUtil.java
```

Important

No incluya el directorio (*root directory name*), solo los directorios y archivos del directorio (*root directory name*).

Cargue el archivo `MessageUtil.zip` identificado en el bucket de entrada denominado `codebuild-region-ID-account-ID-input-bucket`.

Important

CodeCommit en GitHub el caso de los repositorios y de Bitbucket, por convención, debes almacenar un archivo de especificaciones de compilación denominado `buildspec.yml` en la raíz (nivel superior) de cada repositorio o incluir la declaración de especificaciones de compilación como parte de la definición del proyecto de compilación. No cree un archivo ZIP que contenga el código fuente del repositorio y el archivo de especificación de compilación. Solo para entradas de compilación almacenadas en buckets de S3, debe crear un archivo ZIP que contenga el código fuente y convencionalmente, un archivo de especificación de compilación denominado `buildspec.yml` en la raíz (nivel superior) o incluya la declaración de especificaciones de compilación como parte de la definición del proyecto de compilación. Si desea utilizar otro nombre para el archivo de especificación de compilación, o si quiere hacer referencia a una especificación de compilación en una ubicación distinta de la raíz, puede especificar una invalidación de la especificación de compilación como parte de la definición del proyecto de compilación. Para obtener más información, consulte [Nombre de archivo y ubicación de almacenamiento de buildspec](#).

Siguiente paso

[Paso 5: Crear el proyecto de compilación](#)

Paso 5: Crear el proyecto de compilación

(Paso anterior: [Paso 4: Cargar el código fuente y el archivo buildspec](#))

En este paso, creará un proyecto de compilación que usará AWS CodeBuild para ejecutar la compilación. Un proyecto de compilación incluye información sobre cómo ejecutar una compilación, incluido dónde obtener el código fuente, qué entorno de compilación se debe usar, qué comandos de compilación se deben ejecutar y dónde se debe almacenar el resultado de la compilación. Un entorno de compilación representa una combinación de sistema operativo, lenguaje de programación, tiempo de ejecución y herramientas que se CodeBuild utilizan para ejecutar una compilación. El entorno de compilación se expresa como una imagen de Docker. Para obtener más información, consulte [Descripción general de Docker](#) en la página de documentos de Docker.

Para este entorno de compilación, se indica CodeBuild que se utilice una imagen de Docker que contenga una versión del kit de desarrollo de Java (JDK) y Apache Maven.

Para crear el proyecto de compilación

1. Utilice la AWS CLI para ejecutar el comando `create-project`:

```
aws codebuild create-project --generate-cli-skeleton
```

En el resultado se muestran datos con formato JSON. Copie los datos en un archivo denominado `create-project.json` en la ubicación del equipo o instancia local en la que haya instalado la AWS CLI. Si ha elegido usar un nombre de archivo diferente, asegúrese de usarlo en este tutorial.

Modifique los datos copiados de modo que sigan este formato y, a continuación, guarde los resultados:

```
{
  "name": "codebuild-demo-project",
  "source": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-input-bucket/MessageUtil.zip"
  },
  "artifacts": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-output-bucket"
  },
}
```

```
"environment": {
  "type": "LINUX_CONTAINER",
  "image": "aws/codebuild/standard:5.0",
  "computeType": "BUILD_GENERAL1_SMALL"
},
"serviceRole": "serviceIAMRole"
}
```

Sustituya *ServiceIamRole* por el nombre de recurso de Amazon (ARN) de un rol CodeBuild de servicio (por ejemplo,). `arn:aws:iam::account-ID:role/role-name` Para crear uno, consulte [Crear un rol de servicio de CodeBuild](#).

En estos datos:

- `name` representa un identificador obligatorio para este proyecto de compilación (en este ejemplo, `codebuild-demo-project`). Los nombres de los proyectos de compilación deben ser únicos en todos los proyectos de compilación de su cuenta.
- En `source`, `type` es un valor obligatorio que representa el tipo de repositorio del código fuente (en este ejemplo, `S3` para un bucket de Amazon S3).
- En `source`, `location` representa la ruta del código fuente (en este ejemplo, el nombre del bucket de entrada seguido del nombre del archivo ZIP).
- En `artifacts`, `type` es un valor obligatorio que representa el tipo de repositorio del artefacto de salida de la compilación (en este ejemplo, `S3` para un bucket de Amazon S3).
- En `artifacts`, `location` representa el nombre del bucket de salida que ha creado o identificado anteriormente (en este ejemplo, `codebuild-region-ID-account-ID-output-bucket`).
- En `environment`, `type` es un valor obligatorio que representa el tipo de entorno de compilación (en este ejemplo, `LINUX_CONTAINER`).
- Para `environment`, `image` es un valor obligatorio que representa la combinación de nombre y etiqueta de imagen de Docker que utiliza este proyecto de compilación, según lo especificado por el tipo de repositorio de imágenes de Docker (en este ejemplo, `aws/codebuild/standard:5.0` para una imagen de Docker en el repositorio de imágenes de Docker). CodeBuild `aws/codebuild/standard` es el nombre de la imagen de Docker. `5.0` es la etiqueta de la imagen de Docker.

Para encontrar más imágenes de Docker que pueda utilizar en sus escenarios, consulte [Referencia de entornos de compilación](#).

- Para `environment`, `computeType` es un valor obligatorio que representa los recursos informáticos utilizados CodeBuild (en este ejemplo, `BUILD_GENERAL1_SMALL`).

Note

Otros valores disponibles en los datos originales con formato JSON, como `description`, `buildspec`, `auth` (incluido `type` y `resource`), `path`, `namespaceType`, `name` (para `artifacts`), `packaging`, `environmentVariables` (incluido `name` y `value`), `timeoutInMinutes`, `encryptionKey` y `tags` (incluido `key` y `value`) son opcionales. No se utilizan en este tutorial, por lo que no se muestran aquí. Para obtener más información, consulte [Crear un proyecto de compilación \(AWS CLI\)](#).

2. Cambie al directorio que contiene el archivo que acaba de guardar y, a continuación, vuelva a ejecutar el comando `create-project`.

```
aws codebuild create-project --cli-input-json file://create-project.json
```

Si el comando se ejecuta correctamente, aparecen datos similares a los siguientes en el resultado.

```
{
  "project": {
    "name": "codebuild-demo-project",
    "serviceRole": "serviceIAMRole",
    "tags": [],
    "artifacts": {
      "packaging": "NONE",
      "type": "S3",
      "location": "codebuild-region-ID-account-ID-output-bucket",
      "name": "message-util.zip"
    },
    "lastModified": 1472661575.244,
    "timeoutInMinutes": 60,
    "created": 1472661575.244,
    "environment": {
      "computeType": "BUILD_GENERAL1_SMALL",
      "image": "aws/codebuild/standard:5.0",
      "type": "LINUX_CONTAINER",
      "environmentVariables": []
    }
  }
}
```

```
    },
    "source": {
      "type": "S3",
      "location": "codebuild-region-ID-account-ID-input-bucket/MessageUtil.zip"
    },
    "encryptionKey": "arn:aws:kms:region-ID:account-ID:alias/aws/s3",
    "arn": "arn:aws:codebuild:region-ID:account-ID:project/codebuild-demo-project"
  }
}
```

- `project` representa información acerca de este proyecto de compilación.
- `tags` representa todas las etiquetas que se han declarado.
- `packaging` representa la forma en que el artefacto de salida de la compilación se almacena en el bucket de salida. `NONE` indica que se crea una carpeta en el bucket de salida. El artefacto de salida de la compilación se almacena en esa carpeta.
- `lastModified` representa la fecha, en formato de fecha Unix, en la que se cambió por última vez la información sobre el proyecto de compilación.
- `timeoutInMinutes` representa el número de minutos tras los que CodeBuild se detiene la compilación si no se ha completado. (El valor predeterminado es 60 minutos.)
- `created` representa la fecha, en formato de fecha Unix, en la que se creó el proyecto de compilación.
- `environmentVariables` representa cualquier variable de entorno que se haya declarado y que esté disponible CodeBuild para su uso durante la compilación.
- `encryptionKey` representa el ARN de la clave administrada por el cliente que CodeBuild se utilizó para cifrar el artefacto de salida de la compilación.
- `arn` representa el ARN del proyecto de compilación.

Note

Tras ejecutar el `create-project` comando, puede aparecer un mensaje de error similar al siguiente: Usuario: El ***user-ARN*** no está autorizado a realizar: `codebuild:CreateProject` Lo más probable es que esto se deba a que lo configuró AWS CLI con las credenciales de un usuario que no tiene los permisos suficientes CodeBuild para crear proyectos de compilación. Para solucionar este problema, configure la AWS CLI con credenciales que pertenezcan a alguna de las siguientes entidades de IAM:

- Un usuario administrador en su cuenta de AWS. Para obtener más información, consulte [Creación del primer usuario raíz y grupo de Cuenta de AWS](#) en la Guía del usuario.
- Un usuario en su cuenta de AWS con las políticas administradas `AWSCodeBuildAdminAccess`, `AmazonS3ReadOnlyAccess` y `IAMFullAccess` vinculadas a ese usuario o un grupo de IAM al que pertenece el usuario. Si no dispone de un usuario o grupo de AWS en su cuenta de con estos permisos, y no puede añadir estos permisos al usuario o grupo, póngase en contacto con el administrador de su cuenta de AWS para obtener ayuda. Para obtener más información, consulte [AWS políticas gestionadas \(predefinidas\) para AWS CodeBuild](#).

Siguiente paso

[Paso 6: Ejecutar la compilación](#)

Paso 6: Ejecutar la compilación

(Paso anterior: [Paso 5: Crear el proyecto de compilación](#))

En este paso, indicará a AWS CodeBuild que ejecute la compilación con la configuración del proyecto de compilación.

Para ejecutar la compilación

1. Utilice la AWS CLI para ejecutar el comando `start-build`:

```
aws codebuild start-build --project-name project-name
```

Sustituya *nombre-proyecto* por el nombre de su proyecto de compilación del paso anterior (por ejemplo, `codebuild-demo-project`).

2. Si el comando se ejecuta correctamente, aparecerán datos similares a los siguientes en el resultado:

```
{
  "build": {
    "buildComplete": false,
    "initiator": "user-name",
    "artifacts": {
```

```
    "location": "arn:aws:s3:::codebuild-region-ID-account-ID-output-bucket/  
message-util.zip"  
  },  
  "projectName": "codebuild-demo-project",  
  "timeoutInMinutes": 60,  
  "buildStatus": "IN_PROGRESS",  
  "environment": {  
    "computeType": "BUILD_GENERAL1_SMALL",  
    "image": "aws/codebuild/standard:5.0",  
    "type": "LINUX_CONTAINER",  
    "environmentVariables": []  
  },  
  "source": {  
    "type": "S3",  
    "location": "codebuild-region-ID-account-ID-input-bucket/MessageUtil.zip"  
  },  
  "currentPhase": "SUBMITTED",  
  "startTime": 1472848787.882,  
  "id": "codebuild-demo-project:0cfbb6ec-3db9-4e8c-992b-1ab28EXAMPLE",  
  "arn": "arn:aws:codebuild:region-ID:account-ID:build/codebuild-demo-  
project:0cfbb6ec-3db9-4e8c-992b-1ab28EXAMPLE"  
  }  
}
```

- `build` representa información acerca de esta compilación.
 - `buildComplete` indica si la compilación se ha completado (`true`). De lo contrario, `false`.
 - `initiator` representa la entidad que inició la compilación.
 - `artifacts` representa información sobre la salida de la compilación, incluida su ubicación.
 - `projectName` representa el nombre del proyecto de compilación.
 - `buildStatus` representa el estado actual de la compilación cuando se ejecutó el comando `start-build`.
 - `currentPhase` representa la fase actual de la compilación cuando se ejecutó el comando `start-build`.
 - `startTime` representa la hora, en formato de hora Unix, en la que se inició el proceso de compilación.
 - `id` representa el ID de la compilación.
 - `arn` representa el ARN de la compilación.

Anote el valor de `id`. Lo necesitará en el siguiente paso.

Siguiente paso

[Paso 7: Ver información resumida sobre la compilación](#)

Paso 7: Ver información resumida sobre la compilación

(Paso anterior: [Paso 6: Ejecutar la compilación](#))

En este paso, verá información resumida sobre el estado de la compilación.

Para ver la información resumida de la compilación

Utilice la AWS CLI para ejecutar el comando `batch-get-builds`.

```
aws codebuild batch-get-builds --ids id
```

Sustituya *id* por el valor de `id` que apareció en el resultado del paso anterior.

Si el comando se ejecuta correctamente, aparecen datos similares a los siguientes en el resultado.

```
{
  "buildsNotFound": [],
  "builds": [
    {
      "buildComplete": true,
      "phases": [
        {
          "phaseStatus": "SUCCEEDED",
          "endTime": 1472848788.525,
          "phaseType": "SUBMITTED",
          "durationInSeconds": 0,
          "startTime": 1472848787.882
        },
        ... The full list of build phases has been omitted for brevity ...
        {
          "phaseType": "COMPLETED",
          "startTime": 1472848878.079
        }
      ]
    }
  ]
}
```

```

    ],
    "logs": {
      "groupName": "/aws/codebuild/codebuild-demo-project",
      "deepLink": "https://console.aws.amazon.com/cloudwatch/home?region=region-ID#logEvent:group=/aws/codebuild/codebuild-demo-project;stream=38ca1c4a-e9ca-4dbc-bef1-d52bfEXAMPLE",
      "streamName": "38ca1c4a-e9ca-4dbc-bef1-d52bfEXAMPLE"
    },
    "artifacts": {
      "md5sum": "MD5-hash",
      "location": "arn:aws:s3:::codebuild-region-ID-account-ID-output-bucket/message-util.zip",
      "sha256sum": "SHA-256-hash"
    },
    "projectName": "codebuild-demo-project",
    "timeoutInMinutes": 60,
    "initiator": "user-name",
    "buildStatus": "SUCCEEDED",
    "environment": {
      "computeType": "BUILD_GENERAL1_SMALL",
      "image": "aws/codebuild/standard:5.0",
      "type": "LINUX_CONTAINER",
      "environmentVariables": []
    },
    "source": {
      "type": "S3",
      "location": "codebuild-region-ID-account-ID-input-bucket/MessageUtil.zip"
    },
    "currentPhase": "COMPLETED",
    "startTime": 1472848787.882,
    "endTime": 1472848878.079,
    "id": "codebuild-demo-project:38ca1c4a-e9ca-4dbc-bef1-d52bfEXAMPLE",
    "arn": "arn:aws:codebuild:region-ID:account-ID:build/codebuild-demo-project:38ca1c4a-e9ca-4dbc-bef1-d52bfEXAMPLE"
  }
]
}

```

- `buildsNotFound` representa los identificadores de compilación de todas las compilaciones sin información disponible. En este ejemplo, debería estar vacío.
- `builds` representa información de cada compilación con información disponible. En este ejemplo, en el resultado aparece información sobre una única compilación.

- `phases` representa el conjunto de fases de construcción que CodeBuild se ejecutan durante el proceso de creación. La información sobre cada fase de compilación se muestra por separado como `startTime`, `endTime` y `durationInSeconds` (cuándo se inició y terminó la fase de compilación, en formato de hora Unix, y cuánto duró, en segundos), así como `phaseType` como (SUBMITTED, PROVISIONING, DOWNLOAD_SOURCE, INSTALL, PRE_BUILD, BUILD, POST_BUILD, UPLOAD_ARTIFACTS, FINALIZING o COMPLETED) y `phaseStatus` (como SUCCEEDED, FAILED, FAULT, TIMED_OUT, IN_PROGRESS o STOPPED). La primera vez que ejecute el comando `batch-get-builds`, es posible que no haya muchas fases (o que no haya ninguna). Después de ejecuciones posteriores del comando `batch-get-builds` con el mismo ID de compilación, deberían aparecer más fases de compilación en el resultado.
- `logs` representa la información de Amazon CloudWatch Logs sobre los registros de la compilación.
- `md5sum` y `sha256sum` representan los valores hash MD5 y SHA-256 del artefacto de salida de la compilación. Aparecerán en el resultado solamente si el valor `packaging` del proyecto de compilación se establece en ZIP. (No ha establecido este valor en este tutorial). Puede utilizar estos valores hash junto con una herramienta de suma de comprobación para confirmar la integridad y autenticidad de los archivos.

Note

También puede utilizar la consola de Amazon S3 para ver estos valores hash. Seleccione la casilla situada junto al artefacto de salida de la compilación, elija `Actions` (Acciones) y luego elija `Properties` (Propiedades). En el panel `Propiedades`, expanda `Metadatos` y consulte los valores de `x-amz-meta-codebuild-content-md5` y `-content-sha256`. `x-amz-meta-codebuild` (En la consola de Amazon S3, el valor de `ETag` del artefacto de salida de la compilación no debe interpretarse como valor hash MD5 o SHA-256).

Si utiliza los SDK de AWS para obtener estos valores hash, los valores tendrán el nombre `codebuild-content-md5` y `codebuild-content-sha256`.

- `endTime` representa la hora, en formato de hora Unix, en la que terminó el proceso de compilación.

Note

Los metadatos de Amazon S3 tienen un nombre de CodeBuild encabezado `x-amz-meta-codebuild-buildarn` que contiene el nombre `buildArn` de la CodeBuild compilación que publica los artefactos en Amazon S3. Se añade `buildArn` para permitir el seguimiento de las notificaciones en la fuente y para hacer referencia a la compilación de donde procede el artefacto.

Siguiente paso

[Paso 8: Ver información detallada sobre la compilación](#)

Paso 8: Ver información detallada sobre la compilación

(Paso anterior: [Paso 7: Ver información resumida sobre la compilación](#))

En este paso, verá información detallada sobre sus CloudWatch registros de compilación.

Note

Para proteger la información confidencial, los CodeBuild registros ocultan lo siguiente:

- ID de clave de acceso de AWS. Para obtener más información, consulte [Administración de claves de acceso para usuarios de IAM](#) en la Guía del usuario de AWS Identity and Access Management.
- Cadenas especificadas mediante el almacén de parámetros. Para obtener más información, consulte [Almacén de parámetros de Systems Manager](#) y [Tutorial de la consola del almacén de parámetros de Systems Manager](#) en la Guía del usuario de Amazon EC2 Systems Manager.
- Cadenas especificadas mediante AWS Secrets Manager. Para obtener más información, consulte [Administración de claves](#).

Para ver información detallada sobre la compilación

1. Utilice un navegador web para ir a la ubicación `deepLink` que aparecía en el resultado del paso anterior (por ejemplo, `https://console.aws.amazon.com/cloudwatch/`

```
home?region=region-ID#logEvent:group=/aws/codebuild/codebuild-demo-project;stream=38ca1c4a-e9ca-4dbc-bef1-d52bfEXAMPLE).
```

2. En el flujo de CloudWatch registros, puede examinar los eventos del registro. De forma predeterminada, solo se muestra el último conjunto de eventos de log. Para ver eventos de log anteriores, desplácese hasta el principio de la lista.
3. En este tutorial, la mayoría de los eventos de registro contienen información detallada sobre la descarga e instalación de archivos de dependencia de compilación de CodeBuild en su entorno de compilación, lo que probablemente no le interese. Puede usar el cuadro Filter events para reducir la información que se muestra. Por ejemplo, si escribe "[INFO]" en el cuadro Filter events (Filtrar eventos), solo se muestran los eventos que contienen [INFO]. Para obtener más información, consulte [Sintaxis de filtros y patrones](#) en la Guía del CloudWatch usuario de Amazon.

Estas partes de una secuencia de CloudWatch registros pertenecen a este tutorial.

```
...
[Container] 2016/04/15 17:49:42 Entering phase PRE_BUILD
[Container] 2016/04/15 17:49:42 Running command echo Entering pre_build phase...
[Container] 2016/04/15 17:49:42 Entering pre_build phase...
[Container] 2016/04/15 17:49:42 Phase complete: PRE_BUILD Success: true
[Container] 2016/04/15 17:49:42 Entering phase BUILD
[Container] 2016/04/15 17:49:42 Running command echo Entering build phase...
[Container] 2016/04/15 17:49:42 Entering build phase...
[Container] 2016/04/15 17:49:42 Running command mvn install
[Container] 2016/04/15 17:49:44 [INFO] Scanning for projects...
[Container] 2016/04/15 17:49:44 [INFO]
[Container] 2016/04/15 17:49:44 [INFO]
-----
[Container] 2016/04/15 17:49:44 [INFO] Building Message Utility Java Sample App 1.0
[Container] 2016/04/15 17:49:44 [INFO]
-----
...
[Container] 2016/04/15 17:49:55
-----
[Container] 2016/04/15 17:49:55 T E S T S
[Container] 2016/04/15 17:49:55
-----
[Container] 2016/04/15 17:49:55 Running TestMessageUtil
[Container] 2016/04/15 17:49:55 Inside testSalutationMessage()
[Container] 2016/04/15 17:49:55 Hi!Robert
```

```
[Container] 2016/04/15 17:49:55 Inside testPrintMessage()
[Container] 2016/04/15 17:49:55 Robert
[Container] 2016/04/15 17:49:55 Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time
elapsed: 0.018 sec
[Container] 2016/04/15 17:49:55
[Container] 2016/04/15 17:49:55 Results :
[Container] 2016/04/15 17:49:55
[Container] 2016/04/15 17:49:55 Tests run: 2, Failures: 0, Errors: 0, Skipped: 0
...
[Container] 2016/04/15 17:49:56 [INFO]
-----
[Container] 2016/04/15 17:49:56 [INFO] BUILD SUCCESS
[Container] 2016/04/15 17:49:56 [INFO]
-----
[Container] 2016/04/15 17:49:56 [INFO] Total time: 11.845 s
[Container] 2016/04/15 17:49:56 [INFO] Finished at: 2016-04-15T17:49:56+00:00
[Container] 2016/04/15 17:49:56 [INFO] Final Memory: 18M/216M
[Container] 2016/04/15 17:49:56 [INFO]
-----
[Container] 2016/04/15 17:49:56 Phase complete: BUILD Success: true
[Container] 2016/04/15 17:49:56 Entering phase POST_BUILD
[Container] 2016/04/15 17:49:56 Running command echo Entering post_build phase...
[Container] 2016/04/15 17:49:56 Entering post_build phase...
[Container] 2016/04/15 17:49:56 Phase complete: POST_BUILD Success: true
[Container] 2016/04/15 17:49:57 Preparing to copy artifacts
[Container] 2016/04/15 17:49:57 Assembling file list
[Container] 2016/04/15 17:49:57 Expanding target/messageUtil-1.0.jar
[Container] 2016/04/15 17:49:57 Found target/messageUtil-1.0.jar
[Container] 2016/04/15 17:49:57 Creating zip artifact
```

En este ejemplo, completó CodeBuild satisfactoriamente las fases previa, posterior y posterior a la compilación. Ha ejecutado las pruebas unitarias y ha compilado correctamente el archivo `messageUtil-1.0.jar`.

Siguiente paso

[Paso 9: Obtener el artefacto de salida de la compilación](#)

Paso 9: Obtener el artefacto de salida de la compilación

(Paso anterior: [Paso 8: Ver información detallada sobre la compilación](#))

En este paso, obtendrá el `messageUtil-1.0.jar` archivo que se CodeBuild creó y se cargó en el depósito de salida.

Puede utilizar la CodeBuild consola o la consola Amazon S3 para completar este paso.

Para obtener el artefacto de salida de la compilación (consola de AWS CodeBuild)

1. Con la CodeBuild consola aún abierta y la página de detalles de la compilación del paso anterior, selecciona la pestaña Detalles de la construcción y desplázate hacia abajo hasta la sección Artefactos.

 Note

Si la página de detalles de la compilación no se muestra, en la barra de navegación, elija Historial de compilación y, a continuación, elija el enlace Ejecución de la compilación.

2. El enlace a la carpeta Amazon S3 se encuentra en la Ubicación de carga de artefactos. Este enlace abre la carpeta de Amazon S3 donde se encuentra el archivo de artefactos de salida de compilación de `messageUtil-1.0.jar`.

Para obtener el artefacto de salida de la compilación (consola de Amazon S3)

1. Abra la consola de Amazon S3 en <https://console.aws.amazon.com/s3>.
2. Abra `codebuild-region-ID-account-ID-output-bucket`.
3. Abra la carpeta `codebuild-demo-project`.
4. Abra la carpeta `target`, donde se encuentra el archivo de artefactos de salida de compilación `messageUtil-1.0.jar`.

Siguiente paso

[Paso 10: Eliminar los buckets de S3](#)

Paso 10: Eliminar los buckets de S3

(Paso anterior: [Paso 9: Obtener el artefacto de salida de la compilación](#))

Para evitar cargos continuos en su cuenta de AWS, puede eliminar los buckets de entrada utilizados en este tutorial. Para obtener más instrucciones, consulte [Eliminar o vaciar un bucket](#) en la Guía del usuario de Amazon Simple Storage Service.

Si va a utilizar el usuario de IAM o un usuario administrador de IAM para eliminar estos buckets, el usuario debe tener más permisos de acceso. Añada la siguiente instrucción entre los marcadores (**### BEGIN ADDING STATEMENT HERE ###** y **### END ADDING STATEMENTS HERE ###**) a una política de acceso existente para el usuario.

Las elipsis (...) de esta declaración se usan para abreviar. No elimine las instrucciones de la política de acceso existente. No escriba estos puntos suspensivos en la política.

```
{
  "Version": "2012-10-17",
  "Id": "...",
  "Statement": [
    ### BEGIN ADDING STATEMENT HERE ###
    {
      "Effect": "Allow",
      "Action": [
        "s3:DeleteBucket",
        "s3:DeleteObject"
      ],
      "Resource": "*"
    }
    ### END ADDING STATEMENT HERE ###
  ]
}
```

Siguiente paso

[Encapsulación](#)

Encapsulación

En este tutorial, se ha usado AWS CodeBuild para construir un conjunto de archivos de clase Java en un archivo JAR. A continuación, ha consultado los resultados de la compilación.

Ahora puedes probar a utilizarla CodeBuild en tus propios escenarios. Siga las instrucciones en [Planificación de una compilación](#). Si piensa que aún no está preparado, tal vez desee intentar compilar algunos de los ejemplos. Para obtener más información, consulte [Muestras](#).

CodeBuild muestras

Estos grupos de muestras se pueden usar para experimentar con AWS CodeBuild:

Temas

- [Utilice muestras basadas en casos para CodeBuild](#)
- [Ejemplos de Microsoft Windows para CodeBuild](#)

Utilice muestras basadas en casos para CodeBuild

Puedes usar estas muestras basadas en casos de uso para experimentar con: AWS CodeBuild

[Muestras de servicios cruzados](#)

Una lista de muestras multiservicio con las que experimentar. AWS CodeBuild

[Ejemplo de insignias de compilación](#)

Muestra cómo configurar CodeBuild con insignias de construcción.

[Crear un informe de pruebas con el ejemplo de AWS CLI](#)

Utiliza el AWS CLI para crear, ejecutar y ver los resultados de un informe de prueba.

[Muestras de Docker para CodeBuild](#)

Muestra cómo usar imágenes de Docker personalizadas, publicar imágenes de Docker en un repositorio de Amazon ECR y usar imágenes de Docker en un registro privado.

[Alojar la salida de la compilación en un bucket de S3](#)

Muestra cómo se crea un sitio web estático en un bucket de S3 con artefactos de compilación sin cifrar.

[Ejemplo de varios orígenes de entrada y varios artefactos de salida](#)

Muestra cómo utilizar varios orígenes de entrada y varios artefactos de salida en un proyecto de compilación.

[Versiones del entorno de ejecución en el ejemplo del archivo buildspec](#)

Muestra cómo especificar los tiempos de ejecución y sus versiones en el archivo buildspec.

[Ejemplo de versión de origen](#)

Muestra cómo usar una versión específica del código fuente en un proyecto de compilación. CodeBuild

[Ejemplos de repositorios fuente de terceros para CodeBuild](#)

Muestra cómo crear solicitudes BitBucket de GitHub Enterprise Server y GitHub generar solicitudes con webhooks. CodeBuild

[Ejemplo para utilizar el control de versiones semánticas para asignar nombre a artefactos de compilación](#)

Muestra cómo utilizar el control de versiones semánticas para crear un nombre de artefacto en el momento de la compilación.

Ejemplos de servicios cruzados para CodeBuild

Puede usar estos ejemplos de servicios cruzados para experimentar con: AWS CodeBuild

[Ejemplo de Amazon ECR](#)

Utiliza una imagen de Docker de un repositorio de Amazon ECR para usar Apache Maven para generar un único archivo JAR.

[Ejemplo de Amazon EFS](#)

Muestra cómo configurar un archivo buildspec para que un CodeBuild proyecto se monte y se compile en un sistema de archivos Amazon EFS.

[AWS CodePipeline muestras](#)

Muestra cómo crear una compilación con compilaciones por lotes, así como con múltiples fuentes de entrada y múltiples artefactos de salida. AWS CodePipeline

[AWS Config muestra](#)

Muestra cómo configurarlo AWS Config. Muestra CodeBuild los recursos a los que se hace un seguimiento y describe cómo buscar CodeBuild los proyectos en ellos AWS Config.

[Ejemplo de notificaciones de compilación](#)

Utiliza Apache Maven para producir un único archivo JAR. Envía una notificación de compilación a los suscriptores de un tema de Amazon SNS

Ejemplo de Amazon ECR para CodeBuild

En este ejemplo, se utiliza una imagen de Docker en un repositorio de imágenes Amazon Elastic Container Registry (Amazon ECR) para compilar un proyecto de Go de ejemplo.

Important

Al ejecutar este ejemplo, es posible que se produzcan cargos en su AWS cuenta. Estos incluyen los posibles cargos por AWS CodeBuild AWS los recursos y las acciones relacionados con Amazon S3 AWS KMS, CloudWatch Logs y Amazon ECR. Para obtener más información, consulte [CodeBuild los precios, precios](#) de [Amazon S3, precios](#), [AWS Key Management Service precios](#) de [Amazon y CloudWatch precios](#) de [Amazon Elastic Container Registry](#).

Ejecución del ejemplo

Para ejecutar este ejemplo

1. Para crear e insertar la imagen de Docker en su repositorio de imágenes de Amazon ECR, siga los pasos que se indican en la sección "Ejecución del ejemplo" de [Ejemplo de publicación de una imagen de Docker en un repositorio de imágenes de Amazon ECR](#).
2. Crear un proyecto de Go:
 - a. Cree los archivos tal y como se describe en las [Archivos de un proyecto de Go](#) secciones [Estructura de un proyecto de Go](#) y de este tema y, a continuación, cárguelos en un depósito de entrada de S3 o en un AWS CodeCommit repositorio de Bitbucket. GitHub

Important

No cargue (*root directory name*), solo los archivos incluidos en (*root directory name*).

Si utiliza un bucket de entrada de S3, no olvide crear un archivo ZIP que contenga los archivos y cárguelo en el bucket de entrada. No añada (*root directory name*) al archivo ZIP, solo los archivos incluidos en (*root directory name*).

- b. Crea un proyecto de compilación, ejecuta la compilación y consulta la información de compilación relacionada.

Si utilizas el AWS CLI para crear el proyecto de compilación, la entrada del `create-project` comando con formato JSON podría tener un aspecto similar al siguiente. (Sustituya los marcadores de posición por sus propios valores).

```
{
  "name": "sample-go-project",
  "source": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-input-bucket/GoSample.zip"
  },
  "artifacts": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-output-bucket",
    "packaging": "ZIP",
    "name": "GoOutputArtifact.zip"
  },
  "environment": {
    "type": "LINUX_CONTAINER",
    "image": "aws/codebuild/standard:5.0",
    "computeType": "BUILD_GENERAL1_SMALL"
  },
  "serviceRole": "arn:aws:iam::account-ID:role/role-name",
  "encryptionKey": "arn:aws:kms:region-ID:account-ID:key/key-ID"
}
```

- c. Para obtener el artefacto de salida de la compilación, abra el bucket de salida de S3.
 - d. Descargue el archivo `GoOutputArtifact.zip` en su equipo o instancia local y después extraiga el contenido del archivo. En el contenido extraído, obtenga el archivo `hello`.
3. Si se cumple una de las siguientes condiciones, debe añadir permisos a su repositorio de imágenes en Amazon ECR para que AWS CodeBuild pueda incorporar su imagen de Docker al entorno de compilación.
- Su proyecto utiliza CodeBuild credenciales para extraer imágenes de Amazon ECR. Esto se especifica mediante el valor `CODEBUILD` del atributo `imagePullCredentialsType` de `ProjectEnvironment`.
 - Su proyecto utiliza una imagen de Amazon ECR entre más de una cuenta. En este caso, el proyecto debe utilizar su rol de servicio para extraer imágenes de Amazon ECR. Para habilitar este comportamiento, establezca el atributo `imagePullCredentialsType` de `ProjectEnvironment` en `SERVICE_ROLE`.

1. Abra la consola de Amazon ECR en <https://console.aws.amazon.com/ecr/>.
2. En la lista de nombres de repositorio, elija el nombre del repositorio que ha creado o seleccionado.
3. En el panel de navegación, elija Permissions (Permisos), Edit (Editar) y Add statement (Agregar instrucción).
4. En Statement name (Nombre de instrucción), introduzca un identificador (por ejemplo, **CodeBuildAccess**).
5. En Effect (Efecto), deje seleccionado Allow (Permitir). Esto indica que desea permitir el acceso a otra cuenta de AWS .
6. En Principal, realice una de las siguientes acciones:
 - Si su proyecto utiliza CodeBuild credenciales para extraer una imagen de Amazon ECR, introduzca **codebuild.amazonaws.com** en Service principal.
 - Si el proyecto utiliza una imagen de Amazon ECR entre cuentas, en ID de cuentas de AWS introduzca los ID de las cuentas a las que desee dar acceso.
7. Omita la lista Todas las entidades de IAM.
8. En Acción, seleccione las acciones que solo se pueden extraer: ecr:GetDownloadUriForLayer, ecr: y ecr:BatchGetImage. BatchCheckLayerAvailability
9. En Condiciones, añada lo siguiente:

```
{
  "StringEquals":{
    "aws:SourceAccount": "<AWS-account-ID>",
    "aws:SourceArn": "arn:aws:codebuild:<region>:<AWS-account-ID>:project/<project-name>"
  }
}
```

10. Seleccione Guardar.

Esta política aparece en Permisos. La entidad principal es la especificada en Entidad principal en el paso 3 de este procedimiento:

- Si su proyecto utiliza CodeBuild credenciales para extraer una imagen de Amazon ECR, "codebuild.amazonaws.com" aparece en Principios de servicio.
- Si tu proyecto utiliza una imagen de Amazon ECR multicuenta, el ID de la AWS cuenta a la que quieres dar acceso aparece en los ID de AWS cuenta.

El siguiente ejemplo de política utiliza tanto CodeBuild las credenciales como una imagen de Amazon ECR multicuenta.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CodeBuildAccessPrincipal",
      "Effect": "Allow",
      "Principal": {
        "Service": "codebuild.amazonaws.com"
      },
      "Action": [
        "ecr:GetDownloadUrlForLayer",
        "ecr:BatchGetImage",
        "ecr:BatchCheckLayerAvailability"
      ],
      "Condition": {
        "StringEquals": {
          "aws:SourceArn": "arn:aws:codebuild:<region>:<aws-account-id>:project/<project-name>",
          "aws:SourceAccount": "<aws-account-id>"
        }
      }
    },
    {
      "Sid": "CodeBuildAccessCrossAccount",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::<AWS-account-ID>:root"
      },
      "Action": [
        "ecr:GetDownloadUrlForLayer",
        "ecr:BatchGetImage",
        "ecr:BatchCheckLayerAvailability"
      ]
    }
  ]
}
```

- Si sus proyectos utilizan CodeBuild credenciales y desea que tengan acceso abierto al repositorio de Amazon ECR, puede omitir `Condition` las claves y añadir la siguiente política de ejemplo. CodeBuild

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CodeBuildAccessPrincipal",
      "Effect": "Allow",
      "Principal": {
        "Service": "codebuild.amazonaws.com"
      },
      "Action": [
        "ecr:GetDownloadUrlForLayer",
        "ecr:BatchGetImage",
        "ecr:BatchCheckLayerAvailability"
      ]
    },
    {
      "Sid": "CodeBuildAccessCrossAccount",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::<AWS-account-ID>:root"
      },
      "Action": [
        "ecr:GetDownloadUrlForLayer",
        "ecr:BatchGetImage",
        "ecr:BatchCheckLayerAvailability"
      ]
    }
  ]
}
```

4. Cree un proyecto de compilación, ejecute la compilación y consulte la información de la compilación.

Si utilizas el AWS CLI para crear el proyecto de compilación, la entrada del `create-project` comando con formato JSON podría tener un aspecto similar al siguiente. (Sustituya los marcadores de posición por sus propios valores).

```
{
```

```

"name": "amazon-ecr-sample-project",
"source": {
  "type": "S3",
  "location": "codebuild-region-ID-account-ID-input-bucket/GoSample.zip"
},
"artifacts": {
  "type": "S3",
  "location": "codebuild-region-ID-account-ID-output-bucket",
  "packaging": "ZIP",
  "name": "GoOutputArtifact.zip"
},
"environment": {
  "type": "LINUX_CONTAINER",
  "image": "account-ID.dkr.ecr.region-ID.amazonaws.com/your-Amazon-ECR-repo-name:tag",
  "computeType": "BUILD_GENERAL1_SMALL"
},
"serviceRole": "arn:aws:iam::account-ID:role/role-name",
"encryptionKey": "arn:aws:kms:region-ID:account-ID:key/key-ID"
}

```

5. Para obtener el artefacto de salida de la compilación, abra el bucket de salida de S3.
6. Descargue el archivo *GoOutputArtifact*.zip en su equipo o instancia local y después extraiga el contenido del archivo *GoOutputArtifact*.zip. En el contenido extraído, obtenga el archivo hello.

Estructura de un proyecto de Go

En este ejemplo se presupone que existe esta estructura de directorios.

```

(root directory name)
### buildspec.yml
### hello.go

```

Archivos de un proyecto de Go

Este ejemplo usa los siguientes archivos.

buildspec.yml (in (*root directory name*))

```

version: 0.2

```

```
phases:
  install:
    runtime-versions:
      golang: 1.13
  build:
    commands:
      - echo Build started on `date`
      - echo Compiling the Go code
      - go build hello.go
  post_build:
    commands:
      - echo Build completed on `date`
artifacts:
  files:
    - hello
```

hello.go (in *(root directory name)*)

```
package main
import "fmt"

func main() {
  fmt.Println("hello world")
  fmt.Println("1+1 =", 1+1)
  fmt.Println("7.0/3.0 =", 7.0/3.0)
  fmt.Println(true && false)
  fmt.Println(true || false)
  fmt.Println(!true)
}
```

Recursos relacionados

- Para obtener información sobre cómo empezar AWS CodeBuild, consulte. [Empezar a trabajar con AWS CodeBuild utilizando la consola](#)
- Para obtener información sobre la solución de problemas en CodeBuild, consulte [Solución de problemas AWS CodeBuild](#).
- Para obtener información sobre las cuotas en CodeBuild, consulte [Cuotas para AWS CodeBuild](#).

Ejemplo de Amazon Elastic File System para AWS CodeBuild

Es posible que desee crear sus AWS CodeBuild compilaciones en Amazon Elastic File System, un servicio de archivos compartidos y escalable para instancias de Amazon EC2. La capacidad de almacenamiento con Amazon EFS es elástica, por lo que aumenta o disminuye a medida que se añaden y eliminan archivos. Tiene una interfaz de servicios web sencilla que puede utilizar para crear y configurar sistemas de archivos. También administra automáticamente toda la infraestructura de almacenamiento de archivos, por lo que no tiene que preocuparse por la implementación, aplicación de parches o el mantenimiento de configuraciones del sistema de archivos. Para obtener más información, consulte [¿Qué es Amazon Elastic File System?](#) en la Guía del usuario de Amazon Elastic File System.

En este ejemplo, se muestra cómo configurar un CodeBuild proyecto para que monte y, a continuación, cree una aplicación Java en un sistema de archivos Amazon EFS. Antes de empezar, debe tener una aplicación Java lista para ser compilada y cargada en un depósito de entrada de S3 o en un AWS CodeCommit GitHub repositorio de GitHub Enterprise Server o Bitbucket.

Los datos en tránsito para su sistema de archivos están cifrados. Para cifrar los datos en tránsito con una imagen distinta, consulte [Cifrado de datos en tránsito](#).

Pasos generales

En este ejemplo se describen los tres pasos de alto nivel necesarios para utilizar Amazon EFS con AWS CodeBuild:

1. Cree una nube privada virtual (VPC) en su AWS cuenta.
2. Crear un sistema de archivos que utilice esta VPC.
3. Cree y cree un CodeBuild proyecto que utilice la VPC. El CodeBuild proyecto utiliza lo siguiente para identificar el sistema de archivos:
 - Un identificador único del sistema de archivos. El identificador se elige al especificar el sistema de archivos en el proyecto de compilación.
 - El ID del sistema de archivos. El ID se muestra cuando ve el sistema de archivos en la consola de Amazon EFS.
 - Un punto de montaje. Se trata de un directorio en el contenedor Docker que monta el sistema de archivos.
 - Opciones de montaje. Incluyen detalles sobre cómo montar el sistema de archivos.

Note

Un sistema de archivos creado en Amazon EFS solo es compatible con las plataformas Linux.

Cree una VPC mediante AWS CloudFormation

Cree su VPC con una AWS CloudFormation plantilla.

1. Siga las instrucciones de uso [Plantilla de VPC de AWS CloudFormation](#) para AWS CloudFormation crear una VPC.

Note

La VPC creada por esta AWS CloudFormation plantilla tiene dos subredes privadas y dos subredes públicas. Solo debe usar subredes privadas cuando las utilice AWS CodeBuild para montar el sistema de archivos que creó en Amazon EFS. Si utiliza alguna de las subredes públicas, la compilación genera un error.

2. [Inicie sesión en la consola de Amazon VPC AWS Management Console y ábrala en https://console.aws.amazon.com/vpc/.](https://console.aws.amazon.com/vpc/)
3. Elija la VPC con la que creó. AWS CloudFormation
4. En la pestaña Description (Descripción), anote el nombre de la VPC y su ID. Ambas son necesarias para crear el AWS CodeBuild proyecto más adelante en este ejemplo.

Crear un sistema de archivos de Amazon Elastic File System con su VPC

Cree un sistema de archivos sencillo de Amazon EFS de ejemplo utilizando la VPC que creó anteriormente.

1. Inicie sesión en la consola Amazon EFS AWS Management Console y ábrala en [https://console.aws.amazon.com/efs/.](https://console.aws.amazon.com/efs/)
2. Seleccione Crear sistema de archivos.
3. Desde VPC, elija el nombre de VPC que anotó anteriormente en este ejemplo.
4. Deje las zonas de disponibilidad asociadas a las subredes seleccionadas.
5. Elija Paso siguiente.

6. En Añadir etiquetas, para la clave Nombre predeterminada, en Valor, introduzca el nombre de su sistema de archivos de Amazon EFS.
7. Mantenga Transmisión por ráfagas y Uso general seleccionados como sus modos de desempeño y rendimiento predeterminados, y luego elija Paso siguiente.
8. En Configurar acceso de cliente, seleccione Paso siguiente.
9. Seleccione Crear sistema de archivos.
10. (Opcional) Le recomendamos añadir una política a su sistema de archivos de Amazon EFS que imponga el cifrado de los datos en tránsito. En la consola de Amazon EFS, elija Política del sistema de archivos, elija Editar, seleccione la casilla Aplicar el cifrado en tránsito para todos los clientes y, a continuación, seleccione Guardar.

Crear un CodeBuild proyecto para usarlo con Amazon EFS

Cree un AWS CodeBuild proyecto que utilice la VPC que creó anteriormente en este ejemplo. Al ejecutar la compilación, se monta el sistema de archivos de Amazon EFS que se ha creado anteriormente. A continuación, se almacena el archivo .jar que la aplicación Java ha creado en el directorio de puntos de montaje del sistema de archivos.

1. Abra la AWS CodeBuild consola en <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. En el panel de navegación, elija Build projects (Proyectos de compilación) y después Create build project (Crear proyecto de compilación).
3. En Nombre del proyecto, introduzca un nombre para el proyecto.
4. En Source provider (Proveedor de código fuente), elija el repositorio que contiene la aplicación Java que desea compilar.
5. Introduzca la información, como la URL del repositorio, que se CodeBuild utiliza para localizar la aplicación. Las opciones son diferentes para cada proveedor de código fuente. Para obtener más información, consulte [Choose source provider](#).
6. En Imagen de entorno, elija Imagen administrada.
7. En Sistema operativo, elija Amazon Linux 2.
8. En Tiempo(s) de ejecución, elija Estándar.
9. En Imagen, elija aws/codebuild/amazonlinux2-x86_64-standard:4.0.
10. En Tipo de entorno, elija Linux.
11. En Rol de servicio, elija Nuevo rol de servicio. En Nombre del rol, introduzca un nombre para el rol que CodeBuild cree para usted.

12. Expanda Configuración adicional.
13. Seleccione Enable this flag if you want to build Docker images or want your builds to get elevated privileges (Habilite este indicador si desea compilar imágenes de Docker o que sus compilaciones tengan privilegios elevados).

 Note

De forma predeterminada, el daemon de Docker está habilitado para compilaciones que no son de VPC. Si quieres usar contenedores Docker para compilaciones de VPC, [consulta Privilegios de tiempo de ejecución y capacidades de Linux](#) en el sitio web de Docker Docs y habilita el modo privilegiado. Además, Windows no admite el modo privilegiado.

14. En VPC, elija el ID de VPC.
15. En Subredes, elija una o varias subredes privadas asociadas con su VPC. Debe utilizar subredes privadas en una compilación que monta un sistema de archivos de Amazon EFS. Si utiliza una subred pública, la compilación genera un error.
16. En Security groups (Grupos de seguridad), elija el grupo de seguridad predeterminado.
17. En File systems (Sistemas de archivos), escriba la siguiente información:
 - En Identifier (Identificador), introduzca un identificador único del sistema de archivos. El identificador debe tener menos de 129 caracteres y solo debe contener caracteres alfanuméricos y guiones bajos. CodeBuild utiliza este identificador para crear una variable de entorno que identifique el sistema de archivos elástico. El formato de la variable de entorno es `CODEBUILD_<file_system_identifier>`, en mayúsculas. Por ejemplo, si introduce `my_efs`, la variable de entorno es `CODEBUILD_MY_EFS`.
 - En ID, elija el ID del sistema de archivos.
 - (Opcional) Introduzca un directorio en el sistema de archivos. CodeBuild monta este directorio. Si deja la ruta del directorio en blanco, CodeBuild monta todo el sistema de archivos. La ruta es relativa a la raíz del sistema de archivos.
 - En Punto de montaje, introduzca la ruta absoluta del directorio en el contenedor de compilación donde se monta el sistema de archivos. Si este directorio no existe, lo CodeBuild crea durante la compilación.
 - (Opcional) Introduzca las opciones de montaje. Si deja las opciones de montaje en blanco, CodeBuild utiliza sus opciones de montaje predeterminadas:

```
nfsvers=4.1
rsiz=1048576
wsiz=1048576
hard
timeo=600
retrans=2
```

Para obtener más información, consulte [Opciones recomendadas de montaje NFS](#) en la Guía del usuario de Amazon Elastic File System.

18. En Build specification (Especificación de compilación), elija Insert build commands (Insertar comandos de compilación) y, a continuación, elija Switch to editor (Cambiar a editor).
19. Introduzca los siguientes comandos de especificaciones de compilación en el editor. Reemplace `<file_system_identifier>` por el identificador que ha especificado en el paso 17. Use letras mayúsculas (por ejemplo, CODEBUILD_MY_EFS).

```
version: 0.2
phases:
  install:
    runtime-versions:
      java: corretto11
  build:
    commands:
      - mvn compile -Dpgg.skip=true -Dmaven.repo.local=
        $CODEBUILD_<file_system_identifier>
```

20. Utilice los valores predeterminados para el resto de opciones y, a continuación, elija Create build project (Crear proyecto de compilación). Cuando la compilación se haya completado, se muestra la página de la consola para su proyecto.
21. Seleccione Iniciar la compilación.

CodeBuild y resumen de muestra de Amazon EFS

Una vez creado el AWS CodeBuild proyecto:

- Tiene un archivo .jar creado por su aplicación Java que se integra en el sistema de archivos de Amazon EFS, en el directorio del punto de montaje.

- Se crea una variable de entorno que identifica el sistema de archivos con el identificador de sistema de archivos que especificó al crear el proyecto.

Para obtener más información, consulte [Montaje de sistemas de archivos](#) en la Guía del usuario de Amazon Elastic File System.

Resolución de problemas

Los siguientes son errores que puede encontrar al configurar Amazon EFS con CodeBuild.

Temas

- [CLIENT_ERROR: no se ha podido montar '127.0.0.1: /'. Permiso denegado.](#)
- [CLIENT_ERROR: no se ha podido montar '127.0.0.1: /'. Un homólogo ha restablecido la conexión.](#)
- [VPC_CLIENT_ERROR: Error de EC2 inesperado: UnauthorizedOperation](#)

CLIENT_ERROR: no se ha podido montar '127.0.0.1: /'. Permiso denegado.

No se admite la autorización de IAM para montar Amazon EFS con CodeBuild. Si utiliza una política personalizada del sistema de archivos de Amazon EFS, tendrá que conceder acceso de lectura y escritura a todas las entidades principales de IAM. Por ejemplo:

```
"Principal": {
  "AWS": "*"
}
```

CLIENT_ERROR: no se ha podido montar '127.0.0.1: /'. Un homólogo ha restablecido la conexión.

Existen dos causas posibles para este error:

- La subred de CodeBuild VPC se encuentra en una zona de disponibilidad diferente a la del destino de montaje de Amazon EFS. Puede resolver este problema añadiendo una subred de VPC en la misma zona de disponibilidad que el destino de montaje de Amazon EFS.
- El grupo de seguridad no tiene permisos para comunicarse con Amazon EFS. Puede resolver este problema añadiendo una regla de entrada que permita todo el tráfico procedente de la VPC (añada el bloque CIDR principal de la VPC) o del grupo de seguridad en sí.

VPC_CLIENT_ERROR: Error de EC2 inesperado: UnauthorizedOperation

Este error se produce cuando todas las subredes de la configuración de VPC CodeBuild del proyecto son subredes públicas. Para garantizar la conectividad de la red, debe haber como mínimo una subred privada en la VPC.

CodePipeline muestras para CodeBuild

Temas

- [AWS CodePipeline integración con CodeBuild y compilaciones por lotes](#)
- [AWS CodePipeline ejemplo de integración con CodeBuild múltiples fuentes de entrada y artefactos de salida](#)

AWS CodePipeline integración con CodeBuild y compilaciones por lotes

AWS CodeBuild ahora admite compilaciones por lotes. En este ejemplo, se muestra cómo AWS CodePipeline crear un proyecto de compilación que utilice compilaciones por lotes.

Puedes usar un archivo con formato JSON que defina la estructura de tu canalización y luego usarlo con él AWS CLI para crear la canalización. Para obtener más información, consulte [Referencia de la estructura de canalización de AWS CodePipeline](#) en la Guía del usuario de AWS CodePipeline .

Compilación por lotes con artefactos individuales

Utilice el archivo JSON siguiente como ejemplo de una estructura de canalización que crea una compilación por lotes con artefactos individuales. Para habilitar las compilaciones por lotes CodePipeline, defina el BatchEnabled parámetro del objeto enconfiguration. true

```
{
  "pipeline": {
    "roleArn": "arn:aws:iam::account-id:role/my-AWS-CodePipeline-service-role-name",
    "stages": [
      {
        "name": "Source",
        "actions": [
          {
            "inputArtifacts": [],
            "name": "Source1",
            "actionTypeId": {
              "category": "Source",
              "owner": "AWS",
```

```
    "version": "1",
    "provider": "S3"
  },
  "outputArtifacts": [
    {
      "name": "source1"
    }
  ],
  "configuration": {
    "S3Bucket": "<my-input-bucket-name>",
    "S3ObjectKey": "my-source-code-file-name.zip"
  },
  "runOrder": 1
},
{
  "inputArtifacts": [],
  "name": "Source2",
  "actionTypeId": {
    "category": "Source",
    "owner": "AWS",
    "version": "1",
    "provider": "S3"
  },
  "outputArtifacts": [
    {
      "name": "source2"
    }
  ],
  "configuration": {
    "S3Bucket": "<my-other-input-bucket-name>",
    "S3ObjectKey": "my-other-source-code-file-name.zip"
  },
  "runOrder": 1
}
]
},
{
  "name": "Build",
  "actions": [
    {
      "inputArtifacts": [
        {
          "name": "source1"
        }
      ],
```

```
        {
          "name": "source2"
        }
      ],
      "name": "Build",
      "actionTypeId": {
        "category": "Build",
        "owner": "AWS",
        "version": "1",
        "provider": "CodeBuild"
      },
      "outputArtifacts": [
        {
          "name": "build1"
        },
        {
          "name": "build1_artifact1"
        },
        {
          "name": "build1_artifact2"
        },
        {
          "name": "build2_artifact1"
        },
        {
          "name": "build2_artifact2"
        }
      ],
      "configuration": {
        "ProjectName": "my-build-project-name",
        "PrimarySource": "source1",
        "BatchEnabled": "true"
      },
      "runOrder": 1
    }
  ]
}
],
"artifactStore": {
  "type": "S3",
  "location": "<AWS-CodePipeline-internal-bucket-name>"
},
"name": "my-pipeline-name",
"version": 1
```

```
}  
}
```

El siguiente es un ejemplo de un archivo CodeBuild buildspec que funcionará con esta configuración de canalización.

```
version: 0.2  
batch:  
  build-list:  
    - identifier: build1  
      env:  
        compute-type: BUILD_GENERAL1_SMALL  
    - identifier: build2  
      env:  
        compute-type: BUILD_GENERAL1_MEDIUM  
  
phases:  
  build:  
    commands:  
      - echo 'file' > output_file  
  
artifacts:  
  files:  
    - output_file  
  secondary-artifacts:  
    artifact1:  
      files:  
        - output_file  
    artifact2:  
      files:  
        - output_file
```

Los nombres de los artefactos de salida especificados en el archivo JSON de la canalización deben coincidir con el identificador de las compilaciones y los artefactos definidos en el archivo de especificación de la compilación. La sintaxis es *buildIdentifier* para los artefactos principales y *buildIdentifier_artifactIdentifier* para los artefactos secundarios.

Por ejemplo, para el nombre del artefacto de salida build1, CodeBuild cargará el artefacto principal de en la ubicación de build1. build1 Para el nombre de salida build1_artifact1, CodeBuild cargará el artefacto secundario artifact1 de build1 a la ubicación de build1_artifact1, y

así sucesivamente. Si solo se especifica una ubicación de salida, el nombre debe ser únicamente *buildIdentifier*.

Una vez creado el archivo JSON, puede crear la canalización. Utilice AWS CLI para ejecutar el comando `create-pipeline` y pasar el archivo al parámetro. `--cli-input-json` Para obtener más información, consulte [Creación de una canalización \(CLI\)](#) en la Guía del usuario de AWS CodePipeline .

Compilación por lotes con artefactos combinados

Utilice el archivo JSON siguiente como ejemplo de una estructura de canalización que crea una compilación por lotes con artefactos combinados. Para habilitar las compilaciones por lotes CodePipeline, defina el `BatchEnabled` parámetro del `configuration` objeto en `true` Para combinar los artefactos de compilación en la misma ubicación, defina el parámetro `CombineArtifacts` del objeto `configuration` como `true`.

```
{
  "pipeline": {
    "roleArn": "arn:aws:iam::account-id:role/my-AWS-CodePipeline-service-role-name",
    "stages": [
      {
        "name": "Source",
        "actions": [
          {
            "inputArtifacts": [],
            "name": "Source1",
            "actionTypeId": {
              "category": "Source",
              "owner": "AWS",
              "version": "1",
              "provider": "S3"
            },
            "outputArtifacts": [
              {
                "name": "source1"
              }
            ],
            "configuration": {
              "S3Bucket": "<my-input-bucket-name>",
              "S3ObjectKey": "my-source-code-file-name.zip"
            },
            "runOrder": 1
          }
        ]
      }
    ]
  }
}
```

```
    },
    {
      "inputArtifacts": [],
      "name": "Source2",
      "actionTypeId": {
        "category": "Source",
        "owner": "AWS",
        "version": "1",
        "provider": "S3"
      },
      "outputArtifacts": [
        {
          "name": "source2"
        }
      ],
      "configuration": {
        "S3Bucket": "<my-other-input-bucket-name>",
        "S3ObjectKey": "my-other-source-code-file-name.zip"
      },
      "runOrder": 1
    }
  ]
},
{
  "name": "Build",
  "actions": [
    {
      "inputArtifacts": [
        {
          "name": "source1"
        },
        {
          "name": "source2"
        }
      ],
      "name": "Build",
      "actionTypeId": {
        "category": "Build",
        "owner": "AWS",
        "version": "1",
        "provider": "CodeBuild"
      },
      "outputArtifacts": [
        {
```

```

        "name": "output1 "
      }
    ],
    "configuration": {
      "ProjectName": "my-build-project-name",
      "PrimarySource": "source1",
      "BatchEnabled": "true",
      "CombineArtifacts": "true"
    },
    "runOrder": 1
  }
]
}
],
"artifactStore": {
  "type": "S3",
  "location": "<AWS-CodePipeline-internal-bucket-name>"
},
"name": "my-pipeline-name",
"version": 1
}
}

```

El siguiente es un ejemplo de un archivo CodeBuild buildspec que funcionará con esta configuración de canalización.

```

version: 0.2
batch:
  build-list:
    - identifier: build1
      env:
        compute-type: BUILD_GENERAL1_SMALL
    - identifier: build2
      env:
        compute-type: BUILD_GENERAL1_MEDIUM

phases:
  build:
    commands:
      - echo 'file' > output_file

artifacts:
  files:

```

```
- output_file
```

Si los artefactos combinados están habilitados para la compilación por lotes, solo se permite una salida. CodeBuild combinará los artefactos principales de todas las compilaciones en un único archivo ZIP.

Una vez creado el archivo JSON, puede crear la canalización. Utilice el AWS CLI para ejecutar el comando `create-pipeline` y pasar el archivo al `--cli-input-json` parámetro. Para obtener más información, consulte [Creación de una canalización \(CLI\)](#) en la Guía del usuario de AWS CodePipeline .

AWS CodePipeline ejemplo de integración con CodeBuild múltiples fuentes de entrada y artefactos de salida

Un AWS CodeBuild proyecto puede tomar más de una fuente de entrada. También puede crear más de un artefacto de salida. En este ejemplo, se muestra cómo AWS CodePipeline crear un proyecto de compilación que utilice varias fuentes de entrada para crear varios artefactos de salida. Para obtener más información, consulte [Ejemplo de varios orígenes de entrada y varios artefactos de salida](#).

Puedes usar un archivo con formato JSON que defina la estructura de tu canalización y luego usarlo con él AWS CLI para crear la canalización. Utilice el archivo JSON siguiente como ejemplo de una estructura de canalización que crea una compilación con más de un origen de entrada y más de un artefacto de salida. Más adelante en este ejemplo podrá ver cómo dicho archivo especifica las distintas entradas y salidas. Para obtener más información, consulta la [referencia sobre la estructura de CodePipeline canalización](#) en la Guía del AWS CodePipeline usuario.

```
{
  "pipeline": {
    "roleArn": "arn:aws:iam::account-id:role/my-AWS-CodePipeline-service-role-name",
    "stages": [
      {
        "name": "Source",
        "actions": [
          {
            "inputArtifacts": [],
            "name": "Source1",
            "actionTypeId": {
              "category": "Source",
              "owner": "AWS",
              "version": "1",
```

```
    "provider": "S3"
  },
  "outputArtifacts": [
    {
      "name": "source1"
    }
  ],
  "configuration": {
    "S3Bucket": "my-input-bucket-name",
    "S3ObjectKey": "my-source-code-file-name.zip"
  },
  "runOrder": 1
},
{
  "inputArtifacts": [],
  "name": "Source2",
  "actionTypeId": {
    "category": "Source",
    "owner": "AWS",
    "version": "1",
    "provider": "S3"
  },
  "outputArtifacts": [
    {
      "name": "source2"
    }
  ],
  "configuration": {
    "S3Bucket": "my-other-input-bucket-name",
    "S3ObjectKey": "my-other-source-code-file-name.zip"
  },
  "runOrder": 1
}
]
},
{
  "name": "Build",
  "actions": [
    {
      "inputArtifacts": [
        {
          "name": "source1"
        },
        {
```

```
        "name": "source2"
      }
    ],
    "name": "Build",
    "actionTypeId": {
      "category": "Build",
      "owner": "AWS",
      "version": "1",
      "provider": "AWS CodeBuild"
    },
    "outputArtifacts": [
      {
        "name": "artifact1"
      },
      {
        "name": "artifact2"
      }
    ],
    "configuration": {
      "ProjectName": "my-build-project-name",
      "PrimarySource": "source1"
    },
    "runOrder": 1
  }
]
}
],
"artifactStore": {
  "type": "S3",
  "location": "AWS-CodePipeline-internal-bucket-name"
},
"name": "my-pipeline-name",
"version": 1
}
}
```

En este archivo JSON:

- Debe designar uno de los orígenes de entrada como `PrimarySource`. Esta fuente es el directorio en el que CodeBuild busca y ejecuta el archivo `buildspec`. La palabra clave `PrimarySource` se usa para especificar la fuente principal en la `configuration` sección de la CodeBuild etapa del archivo JSON.

- Cada origen de entrada se instala en su propio directorio. Este directorio se almacena en la variable de entorno integrado `$CODEBUILD_SRC_DIR` para el origen principal y `$CODEBUILD_SRC_DIR_yourInputArtifactName` para todos los demás orígenes. Para la canalización de este ejemplo, los dos directorios de origen de entrada son `$CODEBUILD_SRC_DIR` y `$CODEBUILD_SRC_DIR_source2`. Para obtener más información, consulte [Variables de entorno en los entornos de compilación](#).
- Los nombres de los artefactos de salida especificados en el archivo JSON de la canalización deben coincidir con los de los artefactos secundarios definidos en el archivo `buildspec`. Esta canalización utiliza el archivo `buildspec` siguiente. Para obtener más información, consulte [Sintaxis de buildspec](#).

```
version: 0.2

phases:
  build:
    commands:
      - touch source1_file
      - cd $CODEBUILD_SRC_DIR_source2
      - touch source2_file

artifacts:
  files:
    - '**/*'
  secondary-artifacts:
    artifact1:
      base-directory: $CODEBUILD_SRC_DIR
      files:
        - source1_file
    artifact2:
      base-directory: $CODEBUILD_SRC_DIR_source2
      files:
        - source2_file
```

Una vez creado el archivo JSON, puede crear la canalización. Utilice AWS CLI para ejecutar el comando `create-pipeline` y pasar el archivo al `--cli-input-json` parámetro. Para obtener más información, consulte [Creación de una canalización \(CLI\)](#) en la Guía del usuario de AWS CodePipeline .

Úselo AWS Config con una muestra CodeBuild

AWS Config proporciona un inventario de sus AWS recursos y un historial de los cambios de configuración de estos recursos. AWS Config ahora es compatible con AWS CodeBuild como AWS recurso, lo que significa que el servicio puede realizar un seguimiento de sus CodeBuild proyectos. Para obtener más información al respecto de AWS Config, consulte [¿Qué es AWS Config?](#) en la Guía para AWS Config desarrolladores.

Puedes ver la siguiente información sobre CodeBuild los recursos en la página de inventario de recursos de la AWS Config consola:

- Un cronograma de los cambios CodeBuild de configuración.
- Detalles de configuración de cada CodeBuild proyecto.
- Relaciones con otros AWS recursos.
- Una lista de los cambios en sus CodeBuild proyectos.

Los procedimientos de este tema muestran cómo configurar, buscar AWS Config y ver CodeBuild proyectos.

Temas

- [Requisitos previos](#)
- [Configura AWS Config](#)
- [Busca AWS CodeBuild proyectos](#)
- [Visualización AWS CodeBuild de los detalles de configuración en la AWS Config consola](#)

Requisitos previos

Cree su AWS CodeBuild proyecto. Para ver instrucciones, consulte [Creación de un proyecto de compilación](#).

Configura AWS Config

- [Configuración de AWS Config \(consola\)](#)
- [Configuración AWS Config \(AWS CLI\)](#)

Note

Una vez completada la configuración, es posible que pasen hasta 10 minutos antes de que pueda ver AWS CodeBuild los proyectos en la AWS Config consola.

Busca AWS CodeBuild proyectos

1. Inicie sesión en la consola AWS de administración y abra la AWS Config consola en <https://console.aws.amazon.com/config>.
2. En la página de inventario de recursos, seleccione AWS CodeBuild Proyecto en Tipo de recurso. Desplázate hacia abajo y selecciona la casilla CodeBuild del proyecto.
3. Elija Look up (Buscar).
4. Después de añadir la lista de CodeBuild proyectos, elige el enlace del nombre del CodeBuild proyecto en la columna Config timeline.

Visualización AWS CodeBuild de los detalles de configuración en la AWS Config consola

Al buscar los recursos en la página de inventario de recursos, puede elegir el AWS Config cronograma para ver los detalles CodeBuild del proyecto. La página de detalles de un recurso proporciona información sobre la configuración, las relaciones y el número de cambios realizados en dicho recurso.

Los bloques de la parte superior de la página se denominan colectivamente escala de tiempo. La escala de tiempo muestra la fecha y la hora a las que se realizó el registro.

Para obtener más información, consulta la sección [Visualización de los detalles de configuración en la AWS Config consola](#) en la Guía para AWS Config desarrolladores.

Cree un ejemplo de notificaciones para CodeBuild

Amazon CloudWatch Events tiene soporte integrado para AWS CodeBuild. CloudWatch Los eventos son una secuencia de eventos del sistema que describen los cambios en sus AWS recursos. Con CloudWatch Events, se escriben reglas declarativas para asociar los eventos de interés con las acciones automatizadas que se van a llevar a cabo. En este ejemplo, se utilizan Amazon CloudWatch Events y Amazon Simple Notification Service (Amazon SNS) para enviar notificaciones de compilación a los suscriptores siempre que las compilaciones se realicen correctamente, fallen, pasen de una fase de creación a otra o cualquier combinación de estos eventos.

⚠ Important

Al ejecutar este ejemplo, es posible que se le cobren cargos a su AWS cuenta. Estos incluyen posibles cargos por CodeBuild y por AWS los recursos y acciones relacionados con Amazon CloudWatch y Amazon SNS. Para obtener más información, consulte [CodeBuild los precios, los CloudWatch precios de Amazon](#) y los precios de [Amazon SNS](#).

Ejecución del ejemplo

Para ejecutar este ejemplo

1. Si ya tiene un tema configurado y suscrito a Amazon SNS que desee utilizar para este ejemplo, vaya directamente al paso 4. De lo contrario, si utiliza un usuario de IAM en lugar de una cuenta AWS raíz o un usuario administrador para trabajar con Amazon SNS, añada la siguiente declaración (*entre `### BEGIN ADDING STATEMENT HERE ###` y `### END ADDING STATEMENT HERE ###`) al usuario (o grupo de IAM al que está asociado el usuario). No se recomienda utilizar una cuenta raíz. AWS Esta instrucción permite visualizar, crear, suscribir y probar el envío de notificaciones a temas de Amazon SNS. Los puntos suspensivos (. . .) se usan por motivos de brevedad y para ayudarlo a encontrar el lugar donde debe añadir la instrucción. No elimine ninguna instrucción ni incluya estos puntos suspensivos en la política existente.*

```
{
  "Statement": [
    ### BEGIN ADDING STATEMENT HERE ###
    {
      "Action": [
        "sns:CreateTopic",
        "sns:GetTopicAttributes",
        "sns:List*",
        "sns:Publish",
        "sns:SetTopicAttributes",
        "sns:Subscribe"
      ],
      "Resource": "*",
      "Effect": "Allow"
    },
    ### END ADDING STATEMENT HERE ###
    ...
  ]
}
```

```
],  
  "Version": "2012-10-17"  
}
```

Note

La entidad de IAM que modifica esta política debe tener permiso en IAM para modificar políticas.

Para obtener más información, consulte [Edición de políticas administradas por el cliente](#) o la sección "Para editar o eliminar una política insertada de un grupo, un usuario o un rol" de [Uso de políticas insertadas \(consola\)](#) en la Guía del usuario de IAM.

2. Cree o identifique un tema en Amazon SNS. AWS CodeBuild usa CloudWatch Events para enviar notificaciones de compilación sobre este tema a través de Amazon SNS.

Para crear un tema:

1. [Abra la consola de Amazon SNS en https://console.aws.amazon.com/sns](https://console.aws.amazon.com/sns).
2. Elija Crear nuevo tema.
3. En el cuadro de diálogo Create new topic (Crear un nuevo tema), en Topic name (Nombre del tema), escriba un nombre para el tema (por ejemplo, **CodeBuildDemoTopic**). Si elige otro nombre, sustitúyalo en todo este ejemplo.
4. Elija Crear nuevo tema.
5. En la CodeBuildDemoTopic página Detalles del tema:, copie el valor del ARN del tema. Necesitará este valor para el siguiente paso.

Topic details: CodeBuildDemoTopic

[Publish to topic](#) [Other topic actions](#)

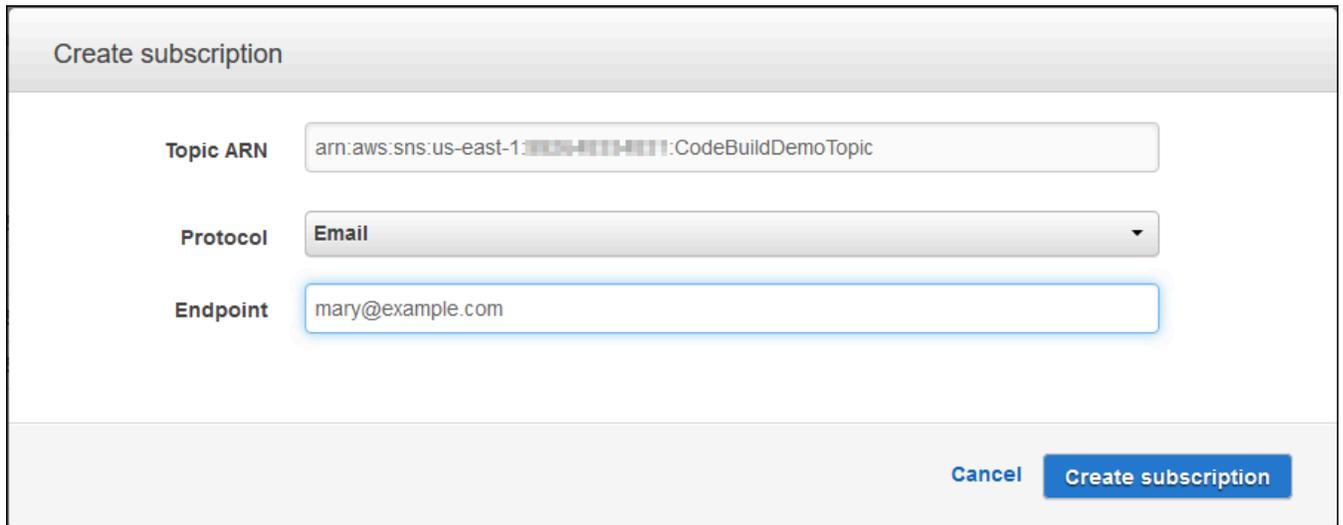
Topic ARN	arn:aws:sns:us-east-1:123456789012:CodeBuildDemoTopic
Topic owner	123456789012
Region	us-east-1
Display name	

Para obtener más información, consulte [Crear un tema](#) en la Guía para desarrolladores de Amazon SNS.

3. Suscriba a uno o varios destinatarios al tema para que reciban notificaciones por correo electrónico.

Para suscribir a un destinatario a un tema:

1. Con la consola de Amazon SNS abierta del paso anterior, en el panel de navegación, seleccione Suscripciones y, a continuación, elija Crear suscripción.
2. En Create subscription (Crear suscripción), en Topic ARN (ARN del tema), pegue el ARN del tema que copió en el paso anterior.
3. En Protocolo, elija Correo electrónico.
4. En Punto de conexión, escriba la dirección de correo electrónico completa del destinatario.



The screenshot shows the 'Create subscription' form in the AWS console. The form is titled 'Create subscription' and has three input fields: 'Topic ARN' with the value 'arn:aws:sns:us-east-1:123456789012:CodeBuildDemoTopic', 'Protocol' set to 'Email', and 'Endpoint' with the value 'mary@example.com'. At the bottom right, there are two buttons: 'Cancel' and 'Create subscription'.

5. Elija Create subscription (Crear suscripción).
6. Amazon SNS envía un correo electrónico de confirmación de suscripción al destinatario. Para empezar a recibir notificaciones por correo electrónico, el destinatario debe elegir el enlace Confirm subscription en el mensaje de confirmación de la suscripción. Cuando el destinatario hace clic en el enlace, si se ha suscrito correctamente, Amazon SNS muestra un mensaje de confirmación en el navegador web.

Para obtener más información, consulte [Suscribirse a un tema](#) en la Guía del desarrollador de Amazon SNS.

- Si utiliza un usuario en lugar de una cuenta AWS raíz o un usuario administrador para trabajar con CloudWatch Events, añada la siguiente instrucción (entre *### BEGIN ADDING STATEMENT HERE ###* y *### END ADDING STATEMENT HERE ###*) al usuario (o grupo de IAM al que está asociado el usuario). No AWS se recomienda usar una cuenta raíz. Esta declaración se utiliza para permitir al usuario trabajar con CloudWatch eventos. Los puntos suspensivos (. . .) se usan por motivos de brevedad y para ayudarle a encontrar el lugar donde debe añadir la instrucción. No elimine ninguna instrucción ni incluya estos puntos suspensivos en la política existente.

```
{
  "Statement": [
    ### BEGIN ADDING STATEMENT HERE ###
    {
      "Action": [
        "events:*",
        "iam:PassRole"
      ],
      "Resource": "*",
      "Effect": "Allow"
    },
    ### END ADDING STATEMENT HERE ###
    ...
  ],
  "Version": "2012-10-17"
}
```

Note

La entidad de IAM que modifica esta política debe tener permiso en IAM para modificar políticas.

Para obtener más información, consulte [Edición de políticas administradas por el cliente](#) o la sección "Para editar o eliminar una política insertada de un grupo, un usuario o un rol" de [Uso de políticas insertadas \(consola\)](#) en la Guía del usuario de IAM.

- Cree una regla en CloudWatch Eventos. Para ello, abra la CloudWatch consola en <https://console.aws.amazon.com/cloudwatch>.

6. En el panel de navegación, bajo Events, elija Rules y después elija Create rule.
7. En la página Step 1: Create rule (Paso 1: Crear regla), las opciones Event Pattern (Patrón de eventos) y Build event pattern to match events by service (Crear un patrón de eventos para encontrar coincidencias de eventos por servicio) ya deberían estar seleccionadas.
8. En Service Name (Nombre de servicio), seleccione CodeBuild, En Event Type (Tipo de evento), la opción All Events (Todos los eventos) ya debería estar seleccionada.
9. El código siguiente debería aparecer en Event Pattern Preview (Vista previa del patrón de eventos):

```
{
  "source": [
    "aws.codebuild"
  ]
}
```

10. Elija Edit (Editar) y sustituya el código de Event Pattern Preview (Vista previa de patrón del eventos) por uno de los dos patrones de reglas siguientes.

Este primer patrón de reglas desencadena un evento siempre que se inicia o se completa una compilación en los proyectos de compilación especificados en AWS CodeBuild.

```
{
  "source": [
    "aws.codebuild"
  ],
  "detail-type": [
    "CodeBuild Build State Change"
  ],
  "detail": {
    "build-status": [
      "IN_PROGRESS",
      "SUCCEEDED",
      "FAILED",
      "STOPPED"
    ],
    "project-name": [
      "my-demo-project-1",
      "my-demo-project-2"
    ]
  }
}
```

```
}

```

En la regla anterior, realice los siguientes cambios de código según sea necesario.

- Para desencadenar un evento cuando se inicia o se completa una compilación, deje todos los valores tal y como se muestran en la matriz `build-status` o quite completamente la matriz `build-status`.
- Para activar un evento solo cuando se complete una compilación, elimine `IN_PROGRESS` de la matriz `build-status`.
- Para activar un evento solo cuando se inicie una compilación, elimine todos los valores a excepción de `IN_PROGRESS` de la matriz `build-status`.
- Para activar eventos para todos los proyectos de compilación, elimine toda la matriz `project-name`.
- Para activar eventos solo para proyectos de compilación individuales, especifique el nombre de cada proyecto de compilación en la matriz `project-name`.

Este segundo patrón de reglas desencadena un evento siempre que una compilación pasa de una fase a otra en los proyectos de compilación especificados en AWS CodeBuild.

```
{
  "source": [
    "aws.codebuild"
  ],
  "detail-type": [
    "CodeBuild Build Phase Change"
  ],
  "detail": {
    "completed-phase": [
      "SUBMITTED",
      "PROVISIONING",
      "DOWNLOAD_SOURCE",
      "INSTALL",
      "PRE_BUILD",
      "BUILD",
      "POST_BUILD",
      "UPLOAD_ARTIFACTS",
      "FINALIZING"
    ],
    "completed-phase-status": [

```

```
    "TIMED_OUT",
    "STOPPED",
    "FAILED",
    "SUCCEEDED",
    "FAULT",
    "CLIENT_ERROR"
  ],
  "project-name": [
    "my-demo-project-1",
    "my-demo-project-2"
  ]
}
```

En la regla anterior, realice los siguientes cambios de código según sea necesario.

- Para desencadenar un evento con cada cambio de fase de compilación (lo que podría enviar hasta nueve notificaciones por cada compilación), deje todos los valores tal y como se muestran en la matriz `completed-phase` o elimine completamente la matriz `completed-phase`.
- Para activar eventos solo para cambios de fase de compilación individuales, elimine el nombre de cada fase de compilación en la matriz `completed-phase` para el que no desee que se active un evento.
- Para activar un evento para cada cambio de estado de fase de compilación, deje todos los valores tal y como se muestran en la matriz `completed-phase-status` o elimine completamente la matriz `completed-phase-status`.
- Para activar eventos solo para cambios de estado de fase de compilación individuales, elimine el nombre de cada estado de fase de compilación en la matriz `completed-phase-status` para el que no desee que se active un evento.
- Para activar eventos para todos los proyectos de compilación, elimine la matriz `project-name`.
- Para activar eventos para proyectos de compilación individuales, especifique el nombre de cada proyecto de compilación en la matriz `project-name`.

Para obtener más información sobre los patrones de eventos, consulta [Patrones de eventos](#) en la Guía del EventBridge usuario de Amazon.

Para obtener más información sobre el filtrado con patrones de eventos, consulta [Filtrado basado en contenido con patrones de eventos](#) en la Guía del EventBridge usuario de Amazon.

 Note

Si desea desencadenar eventos con los cambios de estado de compilación y los cambios de fase de compilación, debe crear dos reglas distintas: una para los cambios de estado y otra para los cambios de fase. Si intenta combinar las dos reglas en una sola, la regla combinada podría producir resultados inesperados o dejar de funcionar.

Cuando haya terminado de reemplazar el código, seleccione Save.

11. En Targets, seleccione Add target.
12. En la lista de destinos, elija SNS topic.
13. En Topic, elija el tema que identificó o creó anteriormente.
14. Expanda Configure input, a continuación, elija Input Transformer.
15. En el cuadro Input Path (Ruta de entrada), escriba una de las siguientes rutas de entrada.

Si en una regla el valor de detail-type es CodeBuild Build State Change, escriba lo siguiente.

```
{"build-id": "$.detail.build-id", "project-name": "$.detail.project-name", "build-status": "$.detail.build-status"}
```

Si en una regla el valor de detail-type es CodeBuild Build Phase Change, escriba lo siguiente.

```
{"build-id": "$.detail.build-id", "project-name": "$.detail.project-name", "completed-phase": "$.detail.completed-phase", "completed-phase-status": "$.detail.completed-phase-status"}
```

Para obtener otros tipos de información, consulte la [Referencia del formato de entrada de las notificaciones de compilación](#).

16. En el cuadro Input Template (Plantilla de entrada), escriba una de las siguientes plantillas de entrada.

Si en una regla el valor de `detail-type` es `CodeBuild Build State Change`, escriba lo siguiente.

```
"Build '<build-id>' for build project '<project-name>' has reached the build status of '<build-status>'."
```

Si en una regla el valor de `detail-type` es `CodeBuild Build Phase Change`, escriba lo siguiente.

```
"Build '<build-id>' for build project '<project-name>' has completed the build phase of '<completed-phase>' with a status of '<completed-phase-status>'."
```

17. Seleccione Configurar los detalles.
18. En la página Step 2: Configure rule details (Página 2: Configurar detalles de la regla), escriba el nombre y una descripción opcional. En State (Estado), deje seleccionado el valor Enabled (Habilitado).
19. Elija Crear regla.
20. Cree proyectos de compilación, ejecute las compilaciones y consulte la información de compilación.
21. Confirme que ahora CodeBuild se están enviando correctamente las notificaciones de compilación. Por ejemplo, compruebe si hay correos electrónicos de notificación de compilación en su bandeja de entrada.

Para cambiar el comportamiento de una regla, en la CloudWatch consola, elige la regla que quieres cambiar, selecciona Acciones y, a continuación, selecciona Editar. Realice los cambios que correspondan en la regla y elija Configure details (Configurar detalles) y Update rule (Actualizar regla).

Para dejar de usar una regla para enviar notificaciones de compilación, en la CloudWatch consola, elige la regla que quieres dejar de usar, selecciona Acciones y, a continuación, selecciona Inhabilitar.

Para eliminar una regla por completo, en la CloudWatch consola, elige la regla que quieres eliminar, selecciona Acciones y, a continuación, selecciona Eliminar.

Recursos relacionados

- Para obtener información sobre cómo empezar a AWS CodeBuild utilizarla, consulte [Empezar a trabajar con AWS CodeBuild utilizando la consola](#).
- Para obtener información sobre la solución de problemas en CodeBuild, consulte [Solución de problemas AWS CodeBuild](#).
- Para obtener información sobre las cuotas en CodeBuild, consulte [Cuotas para AWS CodeBuild](#).

Referencia del formato de entrada de las notificaciones de compilación

CloudWatch entrega las notificaciones en formato JSON.

Las notificaciones de cambio de estado de compilación usan el siguiente formato:

```
{
  "version": "0",
  "id": "c030038d-8c4d-6141-9545-00ff7b7153EX",
  "detail-type": "CodeBuild Build State Change",
  "source": "aws.codebuild",
  "account": "123456789012",
  "time": "2017-09-01T16:14:28Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:codebuild:us-west-2:123456789012:build/my-sample-project:8745a7a9-
c340-456a-9166-edf953571bEX"
  ],
  "detail": {
    "build-status": "SUCCEEDED",
    "project-name": "my-sample-project",
    "build-id": "arn:aws:codebuild:us-west-2:123456789012:build/my-sample-
project:8745a7a9-c340-456a-9166-edf953571bEX",
    "additional-information": {
      "artifact": {
        "md5sum": "da9c44c8a9a3cd4b443126e823168fEX",
        "sha256sum":
"6ccc2ae1df9d155ba83c597051611c42d60e09c6329dcb14a312cecc0a8e39EX",
        "location": "arn:aws:s3:::codebuild-123456789012-output-bucket/my-output-
artifact.zip"
      },
      "environment": {
        "image": "aws/codebuild/standard:5.0",
        "privileged-mode": false,

```

```
    "compute-type": "BUILD_GENERAL1_SMALL",
    "type": "LINUX_CONTAINER",
    "environment-variables": []
  },
  "timeout-in-minutes": 60,
  "build-complete": true,
  "initiator": "MyCodeBuildDemoUser",
  "build-start-time": "Sep 1, 2017 4:12:29 PM",
  "source": {
    "location": "codebuild-123456789012-input-bucket/my-input-artifact.zip",
    "type": "S3"
  },
  "logs": {
    "group-name": "/aws/codebuild/my-sample-project",
    "stream-name": "8745a7a9-c340-456a-9166-edf953571bEX",
    "deep-link": "https://console.aws.amazon.com/cloudwatch/home?region=us-west-2#logEvent:group=/aws/codebuild/my-sample-project;stream=8745a7a9-c340-456a-9166-edf953571bEX"
  },
  "phases": [
    {
      "phase-context": [],
      "start-time": "Sep 1, 2017 4:12:29 PM",
      "end-time": "Sep 1, 2017 4:12:29 PM",
      "duration-in-seconds": 0,
      "phase-type": "SUBMITTED",
      "phase-status": "SUCCEEDED"
    },
    {
      "phase-context": [],
      "start-time": "Sep 1, 2017 4:12:29 PM",
      "end-time": "Sep 1, 2017 4:13:05 PM",
      "duration-in-seconds": 36,
      "phase-type": "PROVISIONING",
      "phase-status": "SUCCEEDED"
    },
    {
      "phase-context": [],
      "start-time": "Sep 1, 2017 4:13:05 PM",
      "end-time": "Sep 1, 2017 4:13:10 PM",
      "duration-in-seconds": 4,
      "phase-type": "DOWNLOAD_SOURCE",
      "phase-status": "SUCCEEDED"
    }
  ],
```

```
{
  "phase-context": [],
  "start-time": "Sep 1, 2017 4:13:10 PM",
  "end-time": "Sep 1, 2017 4:13:10 PM",
  "duration-in-seconds": 0,
  "phase-type": "INSTALL",
  "phase-status": "SUCCEEDED"
},
{
  "phase-context": [],
  "start-time": "Sep 1, 2017 4:13:10 PM",
  "end-time": "Sep 1, 2017 4:13:10 PM",
  "duration-in-seconds": 0,
  "phase-type": "PRE_BUILD",
  "phase-status": "SUCCEEDED"
},
{
  "phase-context": [],
  "start-time": "Sep 1, 2017 4:13:10 PM",
  "end-time": "Sep 1, 2017 4:14:21 PM",
  "duration-in-seconds": 70,
  "phase-type": "BUILD",
  "phase-status": "SUCCEEDED"
},
{
  "phase-context": [],
  "start-time": "Sep 1, 2017 4:14:21 PM",
  "end-time": "Sep 1, 2017 4:14:21 PM",
  "duration-in-seconds": 0,
  "phase-type": "POST_BUILD",
  "phase-status": "SUCCEEDED"
},
{
  "phase-context": [],
  "start-time": "Sep 1, 2017 4:14:21 PM",
  "end-time": "Sep 1, 2017 4:14:21 PM",
  "duration-in-seconds": 0,
  "phase-type": "UPLOAD_ARTIFACTS",
  "phase-status": "SUCCEEDED"
},
{
  "phase-context": [],
  "start-time": "Sep 1, 2017 4:14:21 PM",
  "end-time": "Sep 1, 2017 4:14:26 PM",
```

```

        "duration-in-seconds": 4,
        "phase-type": "FINALIZING",
        "phase-status": "SUCCEEDED"
    },
    {
        "start-time": "Sep 1, 2017 4:14:26 PM",
        "phase-type": "COMPLETED"
    }
]
},
"current-phase": "COMPLETED",
"current-phase-context": "[]",
"version": "1"
}
}

```

Las notificaciones de cambio de fase de compilación usan el siguiente formato:

```

{
  "version": "0",
  "id": "43ddc2bd-af76-9ca5-2dc7-b695e15adeEX",
  "detail-type": "CodeBuild Build Phase Change",
  "source": "aws.codebuild",
  "account": "123456789012",
  "time": "2017-09-01T16:14:21Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:codebuild:us-west-2:123456789012:build/my-sample-project:8745a7a9-c340-456a-9166-edf953571bEX"
  ],
  "detail": {
    "completed-phase": "COMPLETED",
    "project-name": "my-sample-project",
    "build-id": "arn:aws:codebuild:us-west-2:123456789012:build/my-sample-project:8745a7a9-c340-456a-9166-edf953571bEX",
    "completed-phase-context": "[]",
    "additional-information": {
      "artifact": {
        "md5sum": "da9c44c8a9a3cd4b443126e823168fEX",
        "sha256sum":
"6ccc2ae1df9d155ba83c597051611c42d60e09c6329dcb14a312cecc0a8e39EX",
        "location": "arn:aws:s3:::codebuild-123456789012-output-bucket/my-output-artifact.zip"
      }
    }
  }
}

```

```
  },
  "environment": {
    "image": "aws/codebuild/standard:5.0",
    "privileged-mode": false,
    "compute-type": "BUILD_GENERAL1_SMALL",
    "type": "LINUX_CONTAINER",
    "environment-variables": []
  },
  "timeout-in-minutes": 60,
  "build-complete": true,
  "initiator": "MyCodeBuildDemoUser",
  "build-start-time": "Sep 1, 2017 4:12:29 PM",
  "source": {
    "location": "codebuild-123456789012-input-bucket/my-input-artifact.zip",
    "type": "S3"
  },
  "logs": {
    "group-name": "/aws/codebuild/my-sample-project",
    "stream-name": "8745a7a9-c340-456a-9166-edf953571bEX",
    "deep-link": "https://console.aws.amazon.com/cloudwatch/home?region=us-west-2#logEvent:group=/aws/codebuild/my-sample-project;stream=8745a7a9-c340-456a-9166-edf953571bEX"
  },
  "phases": [
    {
      "phase-context": [],
      "start-time": "Sep 1, 2017 4:12:29 PM",
      "end-time": "Sep 1, 2017 4:12:29 PM",
      "duration-in-seconds": 0,
      "phase-type": "SUBMITTED",
      "phase-status": "SUCCEEDED"
    },
    {
      "phase-context": [],
      "start-time": "Sep 1, 2017 4:12:29 PM",
      "end-time": "Sep 1, 2017 4:13:05 PM",
      "duration-in-seconds": 36,
      "phase-type": "PROVISIONING",
      "phase-status": "SUCCEEDED"
    },
    {
      "phase-context": [],
      "start-time": "Sep 1, 2017 4:13:05 PM",
      "end-time": "Sep 1, 2017 4:13:10 PM",
```

```
"duration-in-seconds": 4,
"phase-type": "DOWNLOAD_SOURCE",
"phase-status": "SUCCEEDED"
},
{
  "phase-context": [],
  "start-time": "Sep 1, 2017 4:13:10 PM",
  "end-time": "Sep 1, 2017 4:13:10 PM",
  "duration-in-seconds": 0,
  "phase-type": "INSTALL",
  "phase-status": "SUCCEEDED"
},
{
  "phase-context": [],
  "start-time": "Sep 1, 2017 4:13:10 PM",
  "end-time": "Sep 1, 2017 4:13:10 PM",
  "duration-in-seconds": 0,
  "phase-type": "PRE_BUILD",
  "phase-status": "SUCCEEDED"
},
{
  "phase-context": [],
  "start-time": "Sep 1, 2017 4:13:10 PM",
  "end-time": "Sep 1, 2017 4:14:21 PM",
  "duration-in-seconds": 70,
  "phase-type": "BUILD",
  "phase-status": "SUCCEEDED"
},
{
  "phase-context": [],
  "start-time": "Sep 1, 2017 4:14:21 PM",
  "end-time": "Sep 1, 2017 4:14:21 PM",
  "duration-in-seconds": 0,
  "phase-type": "POST_BUILD",
  "phase-status": "SUCCEEDED"
},
{
  "phase-context": [],
  "start-time": "Sep 1, 2017 4:14:21 PM",
  "end-time": "Sep 1, 2017 4:14:21 PM",
  "duration-in-seconds": 0,
  "phase-type": "UPLOAD_ARTIFACTS",
  "phase-status": "SUCCEEDED"
},
}
```

```
{
  "phase-context": [],
  "start-time": "Sep 1, 2017 4:14:21 PM",
  "end-time": "Sep 1, 2017 4:14:26 PM",
  "duration-in-seconds": 4,
  "phase-type": "FINALIZING",
  "phase-status": "SUCCEEDED"
},
{
  "start-time": "Sep 1, 2017 4:14:26 PM",
  "phase-type": "COMPLETED"
}
]
},
"completed-phase-status": "SUCCEEDED",
"completed-phase-duration-seconds": 4,
"version": "1",
"completed-phase-start": "Sep 1, 2017 4:14:21 PM",
"completed-phase-end": "Sep 1, 2017 4:14:26 PM"
}
```

Crea un ejemplo de insignias con CodeBuild

AWS CodeBuild ahora admite el uso de insignias de compilación, que proporcionan una imagen (insignia) integrable y generada dinámicamente que muestra el estado de la última compilación de un proyecto. Se puede acceder a esta imagen a través de una URL disponible públicamente generada para tu proyecto. CodeBuild Esto permite a cualquier persona ver el estado de un CodeBuild proyecto. Las insignias de compilación no contienen información de seguridad, por lo que no requieren autenticación.

Creación de un proyecto de compilación con insignias de compilación habilitadas (consola)

1. Abra la AWS CodeBuild consola en <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Si aparece una página de CodeBuild información, elija Crear proyecto de compilación. De lo contrario, en el panel de navegación, expanda Compilar, elija Proyectos de compilación y, a continuación, elija Crear proyecto de compilación.
3. En Project name (Nombre de proyecto), escriba un nombre para este proyecto de compilación. Los nombres de los proyectos de compilación deben ser únicos en cada AWS cuenta. También

puede introducir una descripción opcional del proyecto de compilación para ayudar a otros usuarios a entender para qué se utiliza el proyecto.

4. En Source (Código fuente), para Source provider (Proveedor de código fuente), elija el tipo de proveedor de código fuente y, a continuación, realice alguna de las siguientes operaciones:

 Note

CodeBuild no admite insignias de compilación con el proveedor de código fuente de Amazon S3. Como AWS CodePipeline utiliza Amazon S3 para las transferencias de artefactos, las insignias de compilación no se admiten en proyectos de compilación que formen parte de una canalización creada en CodePipeline

- Si lo ha elegido CodeCommit, en Repositorio, elija el nombre del repositorio. Seleccione Habilitar insignia de compilación para convertir el estado de compilación del proyecto en visible e incrustable.
- Si lo desea GitHub, siga las instrucciones para conectarse (o volver a conectarse) con GitHub. En la página GitHub Autorizar la solicitud, en Acceso a la organización, selecciona Solicitar acceso junto a cada repositorio AWS CodeBuild al que quieras acceder. Después de elegir Authorize application (Autorizar aplicación), de vuelta en la consola de AWS CodeBuild , para Repository (Repositorio), elija el nombre del repositorio que contiene el código fuente. Seleccione Habilitar insignia de compilación para convertir el estado de compilación del proyecto en visible e incrustable.
- Si eligió Bitbucket, siga las instrucciones para conectarse (o volver a conectarse) a Bitbucket. En la página Confirm access to your account de Bitbucket, para Organization access, elija Grant access. Tras seleccionar Conceder acceso, en la AWS CodeBuild consola, en Repositorio, elige el nombre del repositorio que contiene el código fuente. Seleccione Habilitar insignia de compilación para convertir el estado de compilación del proyecto en visible e incrustable.

 Important

La actualización del código fuente del proyecto podría afectar a la precisión de las insignias de compilación del proyecto.

5. En Environment (Entorno):

En Environment image (Imagen del entorno), realice alguna de las siguientes operaciones:

- Para usar una imagen de Docker gestionada por AWS CodeBuild, selecciona Imagen gestionada y, a continuación, selecciona Sistema operativo, Tiempo de ejecución, Imagen y Versión de la imagen. Realice una selección en Environment type (Tipo de entorno) si está disponible.
- Para usar otra imagen de Docker, elija Custom image (Imagen personalizada). En Tipo de entorno, elija ARM, Linux, GPU de Linux o Windows. Si elige Otro registro, en URL de registro externo, introduzca el nombre y etiqueta de la imagen de Docker en Docker Hub, usando el formato *docker repository/docker image name*. Si elige Amazon ECR, utilice el repositorio de Amazon ECR y la imagen de Amazon ECR para elegir la imagen de Docker en su cuenta. AWS
- Para usar una imagen de Docker privada, elija Imagen personalizada. En Tipo de entorno, elija ARM, Linux, GPU de Linux o Windows. En Image registry (Registro de imagen), elija Other registry (Otro registro) y, a continuación, introduzca el ARN de las credenciales para su imagen de Docker privada. Es necesario crear las credenciales con Secrets Manager. Para obtener más información, consulte [AWS Secrets Manager](#) en la Guía del usuario de AWS Secrets Manager .

6. En Service role (Rol de servicio), realice una de las operaciones siguientes:

- Si no tiene un rol de CodeBuild servicio, elija Nuevo rol de servicio. Escriba un nombre para el nuevo rol en el campo Role Name.
- Si tiene un rol CodeBuild de servicio, elija el rol de servicio existente. En ARN de rol, seleccione el rol de servicio.

 Note

Cuando utilizas la consola para crear o actualizar un proyecto de compilación, puedes crear un rol de CodeBuild servicio al mismo tiempo. De forma predeterminada, el rol funciona solamente con ese proyecto de compilación. Si utiliza la consola para asociar este rol de servicio a otro proyecto de compilación, el rol se actualizará para que funcione con el otro proyecto de compilación. Un rol de servicio puede funcionar con hasta diez proyectos de compilación.

7. En Archivo de especificación de compilación, realice alguna de las operaciones siguientes:

- Seleccione Usar un archivo de especificaciones de compilación para utilizar el archivo `buyildspec.yml` que figura en el directorio raíz del código fuente.
- Elija Insertar comandos de compilación para usar la consola para insertar comandos de compilación.

Para obtener más información, consulte [Referencia de la especificación de compilación](#).

8. En Artifacts (Artefactos), en Type (Tipo), realice una de las siguientes operaciones:
 - Si no desea crear ningún artefacto de salida de la compilación, elija No artifacts (Sin artefactos).
 - Para almacenar la salida de la compilación en un bucket de S3, seleccione Amazon S3 y, a continuación, haga lo siguiente:
 - Si desea utilizar su nombre de proyecto para el archivo ZIP o carpeta de salida de la compilación, deje en blanco Nombre. De lo contrario, escriba el nombre. De forma predeterminada, el nombre del artefacto es el nombre del proyecto. Si desea usar otro nombre, escríbalo en el cuadro de nombre de artefactos. Si desea producir un archivo ZIP, incluya la extensión `zip`.
 - En Bucket name (Nombre del bucket), seleccione el nombre del bucket de salida.
 - Si eligió Insert build commands (Insertar comandos de compilación) anteriormente en este procedimiento, en Output files (Archivos de salida), escriba las ubicaciones de los archivos de la compilación que desea incluir en el archivo ZIP o carpeta de salida de la compilación. Si hay varias ubicaciones, separe cada una de las ubicaciones con una coma (por ejemplo, `appspec.yml, target/my-app.jar`). Para obtener más información, consulte la descripción de files en [Sintaxis de buildspec](#).
9. Expanda Additional configuration (Configuración adicional) y elija las opciones que correspondan.
10. Elija Crear el proyecto de compilación. En la página Review (Revisar), elija Start build (Comenzar compilación) para ejecutar la compilación.

Creación de un proyecto de compilación con insignias de compilación habilitadas (CLI)

Para obtener información acerca de cómo crear un proyecto de compilación, consulte [Crear un proyecto de compilación \(AWS CLI\)](#). Para incluir insignias de compilación con su proyecto de AWS CodeBuild, debe especificar `badgeEnabled` con un valor de `true`.

Accede a tus insignias de AWS CodeBuild compilación

Puedes usar la AWS CodeBuild consola o la AWS CLI para acceder a las insignias de construcción.

- En la CodeBuild consola, en la lista de proyectos de construcción, en la columna Nombre, elija el enlace que corresponda al proyecto de construcción. En la página Build project (Proyecto de compilación): **nombre_proyecto**, en Configuration (Configuración), elija Copy badge URL (Copiar URL de insignia). Para obtener más información, consulte [Ver los detalles de un proyecto de compilación \(consola\)](#).
- En AWS CLI, ejecute el `batch-get-projects` comando. La URL de insignia de compilación se incluye en la sección de salida de detalles del entorno del proyecto. Para obtener más información, consulte [Ver los detalles de un proyecto de compilación \(AWS CLI\)](#).

La URL de la solicitud de la insignia de compilación se genera con una ramificación común predeterminada, pero es posible especificar cualquier ramificación del repositorio de fuentes que haya usado para ejecutar una compilación. Por ejemplo:

```
https://codebuild.us-east-1.amazon.com/badges?uuid=...&branch=<branch>
```

También es posible especificar una etiqueta del repositorio de origen sustituyendo el parámetro `branch` por el parámetro `tag` en la URL de la insignia. Por ejemplo:

```
https://codebuild.us-east-1.amazon.com/badges?uuid=...&tag=<tag>
```

Publica tus insignias de CodeBuild construcción

Es posible mostrar el estado de la última compilación en un archivo de marcado utilizando la URL de la insignia de compilación en una imagen de marcado. Esto resulta útil para mostrar el estado de la compilación más reciente del archivo `readme.md` del repositorio de código fuente (por ejemplo, o). GitHub CodeCommit Por ejemplo:

```

```

CodeBuild estados de insignias

- **PASSING** La compilación más reciente en la ramificación especificada se ha realizado correctamente.

- **FAILING** Se ha agotado el tiempo de espera, se ha producido un error o se ha detenido la compilación más reciente en la ramificación especificada.
- **IN_PROGRESS** La compilación más reciente en la ramificación especificada está en curso.
- **UNKNOWN** El proyecto aún no ha ejecutado una compilación en la ramificación especificada o en ninguna otra. Además, la característica de insignia de compilación podría estar deshabilitada.

Cree un informe de prueba CodeBuild utilizando el ejemplo AWS CLI

Las pruebas que especifique en su archivo buildspec se ejecutan durante la compilación. En este ejemplo, se muestra cómo utilizarlos AWS CLI para incorporar las pruebas en las versiones integradas CodeBuild. Puede usar JUnit para crear pruebas unitarias o puede usar otra herramienta para crear pruebas de configuración. Después, puede evaluar los resultados de las pruebas para solucionar problemas u optimizar la aplicación.

Puede usar la CodeBuild API o la AWS CodeBuild consola para acceder a los resultados de las pruebas. En este ejemplo se muestra cómo configurar el informe para que los resultados de las pruebas se exporten a un bucket de S3.

Temas

- [Requisitos previos](#)
- [Crear un grupo de informes](#)
- [Configurar un proyecto con un grupo de informes](#)
- [Ejecutar y ver los resultados de un informe](#)

Requisitos previos

- Cree los casos de prueba. Esta muestra se escribe bajo el supuesto de que tiene casos de prueba para incluirlos en su informe de pruebas de muestra. Especifique la ubicación de los archivos de prueba en el archivo buildspec.

Se admiten los siguientes formatos de archivo de informe de prueba:

- Cucumber JSON (.json)
- JUnit XML (.xml)
- NUnit XML (.xml)
- NUnit3 XML (.xml)

- TestNG XML (.xml)
- Visual Studio TRX (.trx)
- XML de Visual Studio TRX (.xml)

Cree sus casos de prueba con cualquier marco de pruebas que pueda crear archivos de informes en uno de esos formatos (por ejemplo, el complemento Surefire JUnit, TestNG o Cucumber).

- Cree un bucket de S3 y anote su nombre. Para obtener más información, consulte [¿Cómo puedo crear un bucket de S3?](#) en la Guía del usuario de Amazon S3.
- Cree un rol de IAM y anote su ARN. Necesita el ARN cuando cree su proyecto de compilación.
- Si el rol no tiene los permisos siguientes, añádalos.

```
{
  "Effect": "Allow",
  "Resource": [
    "*"
  ],
  "Action": [
    "codebuild:CreateReportGroup",
    "codebuild:CreateReport",
    "codebuild:UpdateReport",
    "codebuild:BatchPutTestCases"
  ]
}
```

Para obtener más información, consulte [Permisos para las operaciones de informes de pruebas](#).

Crear un grupo de informes

1. Cree un archivo denominado `CreateReportGroupInput.json`.
2. Cree una carpeta en el bucket de S3 donde se exportarán los resultados de las pruebas.
3. Copie lo siguiente en `CreateReportGroupInput.json`. Para *<bucket-name>*, use el nombre del bucket de S3. Para *<path-to-folder>*, escriba la ruta de acceso a la carpeta en el bucket de S3.

```
{
  "name": "<report-name>",
  "type": "TEST",
  "exportConfig": {
```

```

    "exportConfigType": "S3",
    "s3Destination": {
      "bucket": "<bucket-name>",
      "path": "<path-to-folder>",
      "packaging": "NONE"
    }
  }
}

```

4. Ejecute el siguiente comando en el directorio que contiene `CreateReportGroupInput.json`:

```

aws codebuild create-report-group --cli-input-json file://
CreateReportGroupInput.json

```

El resultado es similar al siguiente. Anote el ARN de `reportGroup`. Se utiliza al crear un proyecto que utiliza este grupo de informes.

```

{
  "reportGroup": {
    "arn": "arn:aws:codebuild:us-west-2:123456789012:report-group/<report-name>",
    "name": "<report-name>",
    "type": "TEST",
    "exportConfig": {
      "exportConfigType": "S3",
      "s3Destination": {
        "bucket": "<s3-bucket-name>",
        "path": "<folder-path>",
        "packaging": "NONE",
        "encryptionKey": "arn:aws:kms:us-west-2:123456789012:alias/aws/s3"
      }
    },
    "created": 1570837165.885,
    "lastModified": 1570837165.885
  }
}

```

Configurar un proyecto con un grupo de informes

Para ejecutar un informe, primero debe crear un proyecto de CodeBuild compilación que esté configurado con su grupo de informes. Los casos de prueba especificados para su grupo de informes se ejecutan cuando se ejecuta una compilación.

1. Cree un archivo buildspec denominado `buildspec.yml`.
2. Utilice el siguiente código YAML como plantilla para su archivo `buildspec.yml`. Asegúrese de incluir los comandos que ejecutan las pruebas. En la sección `reports`, especifique los archivos que contienen los resultados de los casos de prueba. Estos archivos almacenan los resultados de las pruebas a los que puede acceder CodeBuild. Caducan 30 días después de su creación. Estos archivos son distintos de los archivos de resultados de casos de prueba sin procesar que se exportan a un bucket de S3.

```
version: 0.2
  phases:
    install:
      runtime-versions:
        java: openjdk8
    build:
      commands:
        - echo Running tests
        - <enter commands to run your tests>

  reports:
    <report-name-or-arn>: #test file information
    files:
      - '<test-result-files>'
    base-directory: '<optional-base-directory>'
    discard-paths: false #do not remove file paths from test result files
```

Note

En lugar del ARN de un grupo de informes existente, también puede especificar un nombre para un grupo de informes que no se haya creado. Si especifica un nombre en lugar de un ARN, CodeBuild crea un grupo de informes cuando ejecuta una compilación. Su nombre contiene el nombre de proyecto y el nombre especificado en el archivo buildspec en este formato: `project-name-report-group-name`. Para obtener más información, consulte [Crear un informe de pruebas](#) y [Nomenclatura de grupos de informes](#).

3. Cree un archivo denominado `project.json`. Este archivo contiene la entrada para el comando `create-project`.

4. Copie el siguiente JSON en `project.json`. Para `source`, introduzca el tipo y la ubicación del repositorio que contiene los archivos de origen. Para `serviceRole`, especifique el ARN del rol que está utilizando.

```
{
  "name": "test-report-project",
  "description": "sample-test-report-project",
  "source": {
    "type": "CODECOMMIT|CODEPIPELINE|GITHUB|S3|BITBUCKET|GITHUB_ENTERPRISE|
NO_SOURCE",
    "location": "<your-source-url>"
  },
  "artifacts": {
    "type": "NO_ARTIFACTS"
  },
  "cache": {
    "type": "NO_CACHE"
  },
  "environment": {
    "type": "LINUX_CONTAINER",
    "image": "aws/codebuild/standard:5.0",
    "computeType": "small"
  },
  "serviceRole": "arn:aws:iam::<your-aws-account-id>:role/service-role/<your-role-
name>"
}
```

5. Ejecute el siguiente comando en el directorio que contiene `project.json`: De este modo se crea un proyecto llamado `test-project`.

```
aws codebuild create-project --cli-input-json file://project.json
```

Ejecutar y ver los resultados de un informe

En esta sección, se ejecuta una compilación del proyecto creado anteriormente. Durante el proceso de creación, CodeBuild crea un informe con los resultados de los casos de prueba. El informe se incluye en el grupo de informes especificado.

1. Para iniciar una compilación, ejecute el comando siguiente. `test-report-project` es el nombre del proyecto de compilación creado anteriormente. Anote el ID de compilación que aparece en la salida.

```
aws codebuild start-build --project-name test-report-project
```

2. Ejecute el siguiente comando para obtener información sobre su compilación, incluido el ARN del informe. Para `<build-id>`, especifique su ID de compilación. Anote el ARN del informe en la propiedad `reportArns` de la salida.

```
aws codebuild batch-get-builds --ids <build-id>
```

3. Ejecute el comando siguiente para obtener detalles sobre su informe. Para `<report-arn>`, especifique el ARN del informe.

```
aws codebuild batch-get-reports --report-arns <report-arn>
```

El resultado es similar al siguiente. Esta salida de muestra presenta cuántas pruebas fueron correctas, tuvieron errores, se omitieron, produjeron un error o devolvieron un estado desconocido.

```
{
  "reports": [
    {
      "status": "FAILED",
      "reportGroupArn": "<report-group-arn>",
      "name": "<report-group-name>",
      "created": 1573324770.154,
      "exportConfig": {
        "exportConfigType": "S3",
        "s3Destination": {
          "bucket": "<your-S3-bucket>",
          "path": "<path-to-your-report-results>",
          "packaging": "NONE",
          "encryptionKey": "<encryption-key>"
        }
      },
      "expired": 1575916770.0,
      "truncated": false,
      "executionId": "arn:aws:codebuild:us-west-2:123456789012:build/<name-of-build-project>:2c254862-ddf6-4831-a53f-6839a73829c1",

```

```

    "type": "TEST",
    "arn": "<report-arn>",
    "testSummary": {
      "durationInNanoSeconds": 6657770,
      "total": 11,
      "statusCounts": {
        "FAILED": 3,
        "SKIPPED": 7,
        "ERROR": 0,
        "SUCCEEDED": 1,
        "UNKNOWN": 0
      }
    }
  },
  "reportsNotFound": []
}

```

- Ejecute el comando siguiente para enumerar la información sobre los casos de prueba para su informe. Para *<report-arn>*, especifique el ARN del informe. Para el parámetro `--filter` opcional, puede especificar un resultado de estado (SUCCEEDED, FAILED, SKIPPED, ERROR o UNKNOWN).

```

aws codebuild describe-test-cases \
  --report-arn <report-arn> \
  --filter status=SUCCEEDED|FAILED|SKIPPED|ERROR|UNKNOWN

```

El resultado es similar al siguiente.

```

{
  "testCases": [
    {
      "status": "FAILED",
      "name": "Test case 1",
      "expired": 1575916770.0,
      "reportArn": "<report-arn>",
      "prefix": "Cucumber tests for agent",
      "message": "A test message",
      "durationInNanoSeconds": 1540540,
      "testRawDataPath": "<path-to-output-report-files>"
    },
  ],
}

```

```
"status": "SUCCEEDED",
"name": "Test case 2",
"expired": 1575916770.0,
"reportArn": "<report-arn>",
"prefix": "Cucumber tests for agent",
"message": "A test message",
"durationInNanoSeconds": 1540540,
"testRawDataPath": "<path-to-output-report-files>"
}
]
}
```

Muestras de Docker para CodeBuild

Temas

- [Ejemplo de imagen personalizada de Docker para CodeBuild](#)
- [Publicar una imagen de Docker en un ejemplo de repositorio de imágenes de Amazon Elastic Container Registry para CodeBuild](#)
- [Registro privado con AWS Secrets Manager una muestra para CodeBuild](#)

Ejemplo de imagen personalizada de Docker para CodeBuild

En este ejemplo, se crea y ejecuta una imagen de Docker mediante AWS CodeBuild una imagen de compilación de Docker personalizada (docker:dinden Docker Hub).

Para aprender a crear una imagen de Docker utilizando en su lugar una imagen de compilación proporcionada por CodeBuild el soporte de Docker, consulta nuestra [Ejemplo de publicación de una imagen de Docker en un repositorio de imágenes de Amazon ECR](#)

Important

Si ejecutas este ejemplo, es posible que se produzcan cargos en tu cuenta. AWS Estos incluyen posibles cargos por CodeBuild y por AWS los recursos y acciones relacionados con Amazon S3 y CloudWatch Logs. AWS KMS Para obtener más información, consulte [CodeBuild precios](#), [precios de Amazon S3](#), [AWS Key Management Service precios](#) y [CloudWatchprecios de Amazon](#).

Temas

- [Ejecución del ejemplo](#)
- [Estructura de directorios](#)
- [Archivos](#)
- [Recursos relacionados](#)

Ejecución del ejemplo

Para ejecutar este ejemplo

1. Cree los archivos tal y como se describe en las secciones «Estructura de directorios» y «Archivos» de este tema y, a continuación, cárguelos en un depósito de entrada de S3 o en un AWS CodeCommit repositorio de Bitbucket. GitHub

Important

No cargue (*root directory name*), solo los archivos incluidos en (*root directory name*).

Si utiliza un bucket de entrada de S3, no olvide crear un archivo ZIP que contenga los archivos y cárguelo en el bucket de entrada. No añada (*root directory name*) al archivo ZIP, solo los archivos incluidos en (*root directory name*).

2. Crea un proyecto de compilación, ejecuta la compilación y consulta la información de compilación relacionada.

Si utilizas el AWS CLI para crear el proyecto de compilación, la entrada del `create-project` comando con formato JSON podría tener un aspecto similar al siguiente. (Sustituya los marcadores de posición por sus propios valores).

```
{
  "name": "sample-docker-custom-image-project",
  "source": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-input-
bucket/DockerCustomImageSample.zip"
  },
  "artifacts": {
    "type": "NO_ARTIFACTS"
  }
}
```

```
},
"environment": {
  "type": "LINUX_CONTAINER",
  "image": "docker:dind",
  "computeType": "BUILD_GENERAL1_SMALL",
  "privilegedMode": false
},
"serviceRole": "arn:aws:iam::account-ID:role/role-name",
"encryptionKey": "arn:aws:kms:region-ID:account-ID:key/key-ID"
}
```

Note

De forma predeterminada, el daemon de Docker está habilitado para compilaciones que no son de VPC. Si quieres usar contenedores Docker para compilaciones de VPC, [consulta Privilegios de tiempo de ejecución y capacidades de Linux](#) en el sitio web de Docker Docs y habilita el modo privilegiado. Además, Windows no admite el modo privilegiado.

3. Para ver los resultados de compilación, busque en el log de la compilación la cadena Hello, World! Para obtener más información, consulte [Ver detalles de las compilaciones](#).

Estructura de directorios

En este ejemplo se presupone que existe esta estructura de directorios.

```
(root directory name)
### buildspec.yml
### Dockerfile
```

Archivos

La imagen base del sistema operativo utilizado en este ejemplo es Ubuntu. El ejemplo usa los siguientes archivos.

buildspec.yml (in (*root directory name*))

```
version: 0.2

phases:
```

```
pre_build:
  commands:
    - docker build -t helloworld .
build:
  commands:
    - docker images
    - docker run helloworld echo "Hello, World!"
```

Dockerfile (in *(root directory name)*)

```
FROM maven:3.3.9-jdk-8

RUN echo "Hello World"
```

Recursos relacionados

- Para obtener información sobre cómo empezar a utilizarlos, consulte [AWS CodeBuild Empezar a trabajar con AWS CodeBuild utilizando la consola](#)
- Para obtener información sobre la solución de problemas en CodeBuild, consulte [Solución de problemas AWS CodeBuild](#).
- Para obtener información sobre las cuotas en CodeBuild, consulte [Cuotas para AWS CodeBuild](#).

Publicar una imagen de Docker en un ejemplo de repositorio de imágenes de Amazon Elastic Container Registry para CodeBuild

Este ejemplo produce como salida de la compilación una imagen de Docker y la inserta después en un repositorio de imágenes de Amazon Elastic Container Registry (Amazon ECR). Puede adaptar este ejemplo para insertar la imagen de Docker en Docker Hub. Para obtener más información, consulte [Adaptación del ejemplo para insertar la imagen en Docker Hub](#).

Para obtener información acerca de cómo crear una imagen de Docker utilizando una imagen de compilación de Docker personalizada (`docker:dind` en Docker Hub), consulte nuestro [Ejemplo de Docker en una imagen personalizada](#).

Este ejemplo se ha probado con `golang:1.12`.

Utiliza la nueva característica de fases múltiples de Docker, que produce una imagen de Docker como salida de la compilación. A continuación, envía la imagen de Docker a un repositorio de imágenes de Amazon ECR. Las compilaciones de imágenes de fases múltiples de Docker ayudan a

reducir el tamaño de la imagen de Docker final. Para obtener más información, consulte [Use multi-stage builds with Docker](#).

Important

La ejecución de este ejemplo podría generar cargos en su cuenta. AWS Estos incluyen los posibles cargos por AWS CodeBuild AWS los recursos y las acciones relacionados con Amazon S3 AWS KMS, CloudWatch Logs y Amazon ECR. Para obtener más información, consulte [CodeBuild los precios, precios](#) de [Amazon S3, precios](#), [AWS Key Management Service precios](#) de [Amazon y CloudWatch precios](#) de [Amazon Elastic Container Registry](#).

Temas

- [Ejecución del ejemplo](#)
- [Estructura de directorios](#)
- [Archivos](#)
- [Adaptación del ejemplo para insertar la imagen en Docker Hub](#)
- [Recursos relacionados](#)

Ejecución del ejemplo

Para ejecutar este ejemplo

1. Si ya tiene un repositorio de imágenes en Amazon ECR que desee utilizar, vaya al paso 3. De lo contrario, si utiliza un usuario en lugar de una cuenta AWS raíz o un usuario administrador para trabajar con Amazon ECR, añada esta declaración (entre **### BEGIN ADDING STATEMENT HERE ###** y **### END ADDING STATEMENT HERE ###**) *al usuario (o grupo de IAM al que está asociado el usuario)*. No se recomienda usar una cuenta AWS root. Esta declaración permite la creación de repositorios de Amazon ECR para almacenar imágenes de Docker. Los puntos suspensivos (. . .) se usan por motivos de brevedad y para ayudarle a encontrar el lugar donde debe añadir la instrucción. No elimine ninguna instrucción ni incluya estos puntos suspensivos en la política. Para obtener más información, consulte el artículo relacionado con el [uso de políticas integradas a través de la AWS Management Console](#) en la Guía del usuario.

```
{  
  "Statement": [  

```

```

    ### BEGIN ADDING STATEMENT HERE ###
    {
      "Action": [
        "ecr:CreateRepository"
      ],
      "Resource": "*",
      "Effect": "Allow"
    },
    ### END ADDING STATEMENT HERE ###
    ...
  ],
  "Version": "2012-10-17"
}

```

Note

La entidad de IAM que modifica esta política debe tener permiso en IAM para modificar políticas.

2. Cree un repositorio de imágenes en Amazon ECR. Asegúrese de crear el repositorio en la misma AWS región en la que creó el entorno de compilación y en la que ejecutó la compilación. Para obtener más información, consulte [Crear un repositorio](#) en la Guía del usuario de Amazon ECR. El nombre del repositorio debe coincidir con el que especificará más adelante en este procedimiento, representado por la variable de entorno IMAGE_REPO_NAME. Asegúrese de que la política de repositorios de Amazon ECR conceda acceso push de imágenes para su función de IAM CodeBuild de servicio.
3. Añada esta declaración (entre *### BEGIN ADDING STATEMENT HERE ###* y *### END ADDING STATEMENT HERE ###*) a la política que adjuntó a su función de servicio. AWS CodeBuild Esta declaración permite cargar imágenes de Docker CodeBuild a los repositorios de Amazon ECR. Los puntos suspensivos (. . .) se usan por motivos de brevedad y para ayudarle a encontrar el lugar donde debe añadir la instrucción. No elimine ninguna instrucción ni incluya estos puntos suspensivos en la política.

```

{
  "Statement": [
    ### BEGIN ADDING STATEMENT HERE ###
    {
      "Action": [
        "ecr:BatchCheckLayerAvailability",
        "ecr:CompleteLayerUpload",

```

```

    "ecr:GetAuthorizationToken",
    "ecr:InitiateLayerUpload",
    "ecr:PutImage",
    "ecr:UploadLayerPart"
  ],
  "Resource": "*",
  "Effect": "Allow"
},
### END ADDING STATEMENT HERE ###
...
],
"Version": "2012-10-17"
}

```

Note

La entidad de IAM que modifica esta política debe tener permiso en IAM para modificar políticas.

4. Cree los archivos tal y como se describe en las secciones «Estructura de directorios» y «Archivos» de este tema y, a continuación, cárguelos en un depósito de entrada de S3 o en un repositorio AWS CodeCommit de GitHub Bitbucket. Para obtener más información, consulte [Referencia del archivo de definiciones de imágenes](#) en la Guía del usuario de AWS CodePipeline .

Important

No cargue (*root directory name*), solo los archivos incluidos en (*root directory name*).

Si utiliza un bucket de entrada de S3, no olvide crear un archivo ZIP que contenga los archivos y cárguelo en el bucket de entrada. No añada (*root directory name*) al archivo ZIP, solo los archivos incluidos en (*root directory name*).

5. Crea un proyecto de compilación, ejecuta la compilación y consulta la información de la compilación.

Si utiliza la consola para crear un proyecto:

- a. En Operating system (Sistema operativo), elija Ubuntu.
- b. En Runtime, elija Standard (Estándar).

- c. En Imagen, elija `aws/codebuild/standard:5.0`.
- d. Añada las siguientes variables de entorno:
 - `AWS_DEFAULT_REGION` con un valor de *region-ID*
 - `AWS_ACCOUNT_ID` con un valor de ID de *account-ID*
 - `IMAGE_TAG` con un valor de Latest
 - `IMAGE_REPO_NAME` con un valor de *Amazon-ECR-repo-name*

Si utilizas el AWS CLI para crear el proyecto de compilación, la entrada del `create-project` comando con formato JSON podría tener un aspecto similar al siguiente. (Sustituya los marcadores de posición por sus propios valores).

```
{
  "name": "sample-docker-project",
  "source": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-input-bucket/DockerSample.zip"
  },
  "artifacts": {
    "type": "NO_ARTIFACTS"
  },
  "environment": {
    "type": "LINUX_CONTAINER",
    "image": "aws/codebuild/standard:5.0",
    "computeType": "BUILD_GENERAL1_SMALL",
    "environmentVariables": [
      {
        "name": "AWS_DEFAULT_REGION",
        "value": "region-ID"
      },
      {
        "name": "AWS_ACCOUNT_ID",
        "value": "account-ID"
      },
      {
        "name": "IMAGE_REPO_NAME",
        "value": "Amazon-ECR-repo-name"
      },
      {
        "name": "IMAGE_TAG",
```

```

        "value": "latest"
      }
    ],
  },
  "serviceRole": "arn:aws:iam::account-ID:role/role-name",
  "encryptionKey": "arn:aws:kms:region-ID:account-ID:key/key-ID"
}

```

6. Confirma que la imagen de Docker se insertó CodeBuild correctamente en el repositorio:

1. Abra la consola de Amazon ECR en <https://console.aws.amazon.com/ecr/>.
2. Elija el nombre del repositorio. La imagen debe mostrarse en la columna Image tag (Etiqueta de imagen).

Estructura de directorios

En este ejemplo se presupone que existe esta estructura de directorios.

```

(root directory name)
### buildspec.yml
### Dockerfile

```

Archivos

Este ejemplo usa los siguientes archivos.

buildspec.yml (in (*root directory name*))

```

version: 0.2

phases:
  pre_build:
    commands:
      - echo Logging in to Amazon ECR...
      - aws ecr get-login-password --region $AWS_DEFAULT_REGION | docker login --
username AWS --password-stdin $AWS_ACCOUNT_ID.dkr.ecr.$AWS_DEFAULT_REGION.amazonaws.com
  build:
    commands:
      - echo Build started on `date`
      - echo Building the Docker image...
      - docker build -t $IMAGE_REPO_NAME:$IMAGE_TAG .

```

```
- docker tag $IMAGE_REPO_NAME:$IMAGE_TAG $AWS_ACCOUNT_ID.dkr.ecr.
$AWS_DEFAULT_REGION.amazonaws.com/$IMAGE_REPO_NAME:$IMAGE_TAG
post_build:
  commands:
    - echo Build completed on `date`
    - echo Pushing the Docker image...
    - docker push $AWS_ACCOUNT_ID.dkr.ecr.$AWS_DEFAULT_REGION.amazonaws.com/
$IMAGE_REPO_NAME:$IMAGE_TAG
```

Dockerfile (in *(root directory name)*)

```
FROM golang:1.12-alpine AS build
#Install git
RUN apk add --no-cache git
#Get the hello world package from a GitHub repository
RUN go get github.com/golang/example/hello
WORKDIR /go/src/github.com/golang/example/hello
# Build the project and send the output to /bin/HelloWorld
RUN go build -o /bin/HelloWorld

FROM golang:1.12-alpine
#Copy the build's output binary from the previous build container
COPY --from=build /bin/HelloWorld /bin/HelloWorld
ENTRYPOINT ["/bin/HelloWorld"]
```

Note

CodeBuild anula la de las imágenes ENTRYPOINT de Docker personalizadas.

Adaptación del ejemplo para insertar la imagen en Docker Hub

Para enviar la imagen de Docker a Docker Hub en lugar de Amazon ECR, modifique el código de este ejemplo.

Note

Si está utilizando una versión de Docker anterior a la 17.06, quite la opción `--no-include-email`.

1. Sustituya estas líneas de código específicas de Amazon ECR en el archivo `buildspec.yml`:

```
...
pre_build:
  commands:
    - echo Logging in to Amazon ECR...
    - aws ecr get-login-password --region $AWS_DEFAULT_REGION |
docker login --username AWS --password-stdin $AWS_ACCOUNT_ID.dkr.ecr.
$AWS_DEFAULT_REGION.amazonaws.com
build:
  commands:
    - echo Build started on `date`
    - echo Building the Docker image...
    - docker build -t $IMAGE_REPO_NAME:$IMAGE_TAG .
    - docker tag $IMAGE_REPO_NAME:$IMAGE_TAG $AWS_ACCOUNT_ID.dkr.ecr.
$AWS_DEFAULT_REGION.amazonaws.com/$IMAGE_REPO_NAME:$IMAGE_TAG
post_build:
  commands:
    - echo Build completed on `date`
    - echo Pushing the Docker image...
    - docker push $AWS_ACCOUNT_ID.dkr.ecr.$AWS_DEFAULT_REGION.amazonaws.com/
$IMAGE_REPO_NAME:$IMAGE_TAG
...
```

Por estas líneas de código específicas de Docker Hub:

```
...
pre_build:
  commands:
    - echo Logging in to Docker Hub...
    # Type the command to log in to your Docker Hub account here.
build:
  commands:
    - echo Build started on `date`
    - echo Building the Docker image...
    - docker build -t $IMAGE_REPO_NAME:$IMAGE_TAG .
    - docker tag $IMAGE_REPO_NAME:$IMAGE_TAG $IMAGE_REPO_NAME:$IMAGE_TAG
post_build:
  commands:
    - echo Build completed on `date`
    - echo Pushing the Docker image...
    - docker push $IMAGE_REPO_NAME:$IMAGE_TAG
```

```
...
```

2. Sube el código editado a un depósito de entrada de S3 o a un AWS CodeCommit repositorio GitHub de Bitbucket.

⚠ Important

No cargue (*root directory name*), solo los archivos incluidos en (*root directory name*).

Si utiliza un bucket de entrada de S3, no olvide crear un archivo ZIP que contenga los archivos y cárguelo en el bucket de entrada. No añada (*root directory name*) al archivo ZIP, solo los archivos incluidos en (*root directory name*).

3. Sustituya estas líneas de código de la entrada con formato JSON del comando `create-project`:

```
...
  "environmentVariables": [
    {
      "name": "AWS_DEFAULT_REGION",
      "value": "region-ID"
    },
    {
      "name": "AWS_ACCOUNT_ID",
      "value": "account-ID"
    },
    {
      "name": "IMAGE_REPO_NAME",
      "value": "Amazon-ECR-repo-name"
    },
    {
      "name": "IMAGE_TAG",
      "value": "latest"
    }
  ]
}
```

Por estas líneas de código:

```
...
  "environmentVariables": [
```

```
{
  "name": "IMAGE_REPO_NAME",
  "value": "your-Docker-Hub-repo-name"
},
{
  "name": "IMAGE_TAG",
  "value": "latest"
}
]
...

```

4. Crea un entorno de compilación, ejecuta la compilación y consulta la información de compilación relacionada.
5. Confirme que la imagen de Docker se insertó AWS CodeBuild correctamente en el repositorio. Inicie sesión en Docker Hub, vaya al repositorio y elija la pestaña Tags. La etiqueta `latest` debe contener un valor de Last Updated muy reciente.

Recursos relacionados

- Para obtener información sobre cómo empezar AWS CodeBuild, consulte [Empezar a trabajar con AWS CodeBuild utilizando la consola](#).
- Para obtener información sobre la solución de problemas en CodeBuild, consulte [Solución de problemas AWS CodeBuild](#).
- Para obtener información sobre las cuotas en CodeBuild, consulte [Cuotas para AWS CodeBuild](#).

Registro privado con AWS Secrets Manager una muestra para CodeBuild

En este ejemplo, se muestra cómo utilizar una imagen de Docker almacenada en un registro privado como entorno AWS CodeBuild de ejecución. Las credenciales del registro privado están almacenadas en AWS Secrets Manager. Cualquier registro privado funciona con CodeBuild. En esta muestra se utiliza Docker Hub.

Note

Los secretos son visibles para las acciones y no se ocultan cuando se escriben en un archivo.

Requisitos del ejemplo de un registro privado

Para usar un registro privado con AWS CodeBuild, debe tener lo siguiente:

- Un secreto de Secrets Manager que almacena sus credenciales de Docker Hub. Las credenciales se utilizan para acceder al repositorio privado.

Note

Se le cobrará por los secretos que cree.

- Un repositorio privado o una cuenta.
- Una política de IAM de rol de CodeBuild servicio que otorga acceso a tu secreto de Secrets Manager.

Siga estos pasos para crear estos recursos y, a continuación, cree un proyecto de CodeBuild compilación con las imágenes de Docker almacenadas en su registro privado.

Crea un CodeBuild proyecto con un registro privado

1. Para obtener información sobre cómo crear un repositorio privado gratuito, consulte [Repositorios en Docker Hub](#). También puede ejecutar los siguientes comandos en un terminal para extraer una imagen, obtener su ID y enviarla a un nuevo repositorio.

```
docker pull amazonlinux
docker images amazonlinux --format {{.ID}}
docker tag image-id your-username/repository-name:tag
docker login
docker push your-username/repository-name
```

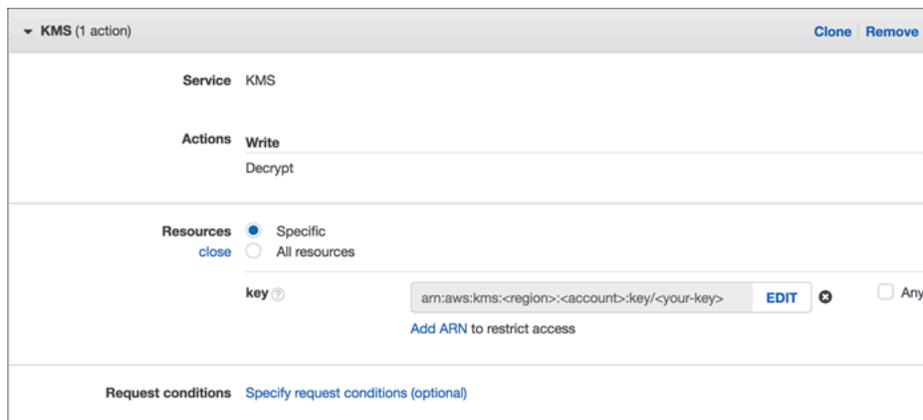
2. Siga los pasos que se indican en [Crear un AWS Secrets Manager secreto](#) en la Guía del AWS Secrets Manager usuario.
 - a. En el paso 3, en Elija tipo de secreto, seleccione Otro tipo de secreto.
 - b. En Pares clave/valor, cree un par clave-valor para el nombre de usuario de Docker Hub y otro par clave-valor para la contraseña de Docker Hub.
 - c. Sigue los pasos de [Crear un AWS Secrets Manager secreto](#).

- d. En el paso 5, en la página Configurar la rotación automática, desactívala porque las claves corresponden a sus credenciales de Docker Hub.
- e. Termina de seguir los pasos de [Crear un AWS Secrets Manager secreto](#).

Para obtener más información, consulte [¿Qué es AWS Secrets Manager?](#)

3. Cuando creas un AWS CodeBuild proyecto en la consola, te CodeBuild adjunta el permiso necesario. Si utiliza una AWS KMS clave que no sea `DefaultEncryptionKey`, debe añadirla al rol de servicio. Para obtener más información, consulte [Modificación de un rol \(consola\)](#) en la Guía del usuario de IAM.

Para que su rol de servicio funcione con Secrets Manager, debe tener, como mínimo, el permiso `secretsmanager:GetSecretValue`.



▼ KMS (1 action) Clone Remove

Service KMS

Actions Write
Decrypt

Resources Specific All resources close

key EDIT Any

[Add ARN to restrict access](#)

Request conditions [Specify request conditions \(optional\)](#)

4. Para usar la consola para crear un proyecto con un entorno almacenado en un registro privado, haga lo siguiente mientras crea un proyecto. Para obtener más información, consulte [Creación de un proyecto de compilación \(consola\)](#).

Note

Si su registro privado está en su VPC, debe tener acceso público a Internet. CodeBuild no puede extraer una imagen de una dirección IP privada en una VPC.

- a. En Imagen de entorno, elija Imagen personalizada.
- b. En Environment type (Tipo de entorno), elija Linux o Windows.
- c. En Registro de imágenes, elija Otro registro.

- d. En URL de registro externa, introduzca la ubicación de la imagen y en Credencial de registro (opcional), introduzca el ARN o el nombre de sus credenciales de Secrets Manager.

 Note

Si las credenciales no existen en la región actual, debe usar el ARN. No puede utilizar el nombre de la credencial si las credenciales existen en otra región.

Crear un sitio web estático con la salida de compilación alojada en un bucket de S3

Puede deshabilitar el cifrado de los artefactos en una compilación, por ejemplo, para publicar artefactos en una ubicación configurada para alojar un sitio web. (Los artefactos cifrados no se pueden publicar). En este ejemplo, se muestra cómo utilizar los webhooks para desencadenar una compilación y publicar sus artefactos en un bucket de S3 configurado como un sitio web.

1. Siga las instrucciones de [configuración para sitio web estático](#) si quiere configurar un bucket de S3 para que funcione como un sitio web.
2. Abra la AWS CodeBuild consola en <https://console.aws.amazon.com/codesuite/codebuild/home>.
3. Si aparece una página de CodeBuild información, elija Crear proyecto de compilación. De lo contrario, en el panel de navegación, expanda Compilar, elija Proyectos de compilación y, a continuación, elija Crear proyecto de compilación.
4. En Project name (Nombre de proyecto), escriba un nombre para este proyecto de compilación. Los nombres de los proyectos de compilación deben ser únicos en cada AWS cuenta. También puede introducir una descripción opcional del proyecto de compilación para ayudar a otros usuarios a entender para qué se utiliza el proyecto.
5. En Fuente, en Proveedor de fuentes, elija GitHub. Sigue las instrucciones para conectarte (o volver a conectarte) y GitHub, a continuación, selecciona Autorizar.

En Webhook, seleccione Rebuild every time a code change is pushed to this repository (Volver a compilar cada vez que un cambio de código se inserte en el repositorio). Solo puede seleccionar esta casilla de verificación si eligió Use a repository in my account (Usar un repositorio de mi cuenta).

Source Add source

Source 1 - Primary

Source provider
GitHub

Repository
 Public repository Repository in my GitHub account

GitHub repository

▼ **Additional configuration**

Git clone depth

Git clone depth - *optional*

Build Status - *optional*
 Report build statuses to source provider when your builds start and finish

Webhook - *optional*
 Rebuild every time a code change is pushed to this repository

Branch filter - *optional*

Enter a regular expression

6. En Environment (Entorno):

En Environment image (Imagen del entorno), realice alguna de las siguientes operaciones:

- Para usar una imagen de Docker gestionada por AWS CodeBuild, selecciona Imagen gestionada y selecciona Sistema operativo, Tiempo de ejecución, Imagen y Versión de la imagen. Realice una selección en Environment type (Tipo de entorno) si está disponible.

- Para usar otra imagen de Docker, elija Custom image (Imagen personalizada). En Tipo de entorno, elija ARM, Linux, GPU de Linux o Windows. Si elige Otro registro, en URL de registro externo, introduzca el nombre y etiqueta de la imagen de Docker en Docker Hub, usando el formato *docker repository/docker image name*. Si elige Amazon ECR, utilice el repositorio de Amazon ECR y la imagen de Amazon ECR para elegir la imagen de Docker en su cuenta. AWS
 - Para usar una imagen de Docker privada, elija Imagen personalizada. En Tipo de entorno, elija ARM, Linux, GPU de Linux o Windows. En Image registry (Registro de imagen), elija Other registry (Otro registro) y, a continuación, introduzca el ARN de las credenciales para su imagen de Docker privada. Es necesario crear las credenciales con Secrets Manager. Para obtener más información, consulte [AWS Secrets Manager](#) en la Guía del usuario de AWS Secrets Manager .
7. En Service role (Rol de servicio), realice una de las operaciones siguientes:
- Si no tiene un rol de CodeBuild servicio, elija Nuevo rol de servicio. Escriba un nombre para el nuevo rol en el campo Role Name.
 - Si tiene un rol CodeBuild de servicio, elija el rol de servicio existente. En ARN de rol, seleccione el rol de servicio.

 Note

Cuando utilizas la consola para crear o actualizar un proyecto de compilación, puedes crear un rol de CodeBuild servicio al mismo tiempo. De forma predeterminada, el rol funciona solamente con ese proyecto de compilación. Si utiliza la consola para asociar este rol de servicio a otro proyecto de compilación, el rol se actualizará para que funcione con el otro proyecto de compilación. Un rol de servicio puede funcionar con hasta diez proyectos de compilación.

8. En Archivo de especificación de compilación, realice alguna de las operaciones siguientes:
- Seleccione Usar un archivo de especificaciones de compilación para utilizar el archivo `buyildspec.yml` que figura en el directorio raíz del código fuente.
 - Elija Insertar comandos de compilación para usar la consola para insertar comandos de compilación.

Para obtener más información, consulte [Referencia de la especificación de compilación](#).

9. En Artefactos, en Tipo, elija Amazon S3 para almacenar la salida de la compilación en un bucket de S3.
10. En Bucket name (Nombre del bucket), elija el nombre del bucket de S3 que ha configurado para que funcione como un sitio web en el paso 1.
11. Si eligió Insert build commands (Insertar comandos de compilación) en Environment: How to build (Entorno: cómo compilar), escriba las ubicaciones de los archivos de la compilación que desea incluir en el bucket de salida en Output files (Archivos de salida). Si tiene varias ubicaciones, utilice una coma para separarlas (por ejemplo, **appspect.yml**, **target/my-app.jar**). Para obtener más información, consulte [Artifacts reference-key in the buildspec file](#).
12. Seleccione Disable artifacts encryption (Deshabilitar cifrado de artefactos).
13. Expanda Additional configuration (Configuración adicional) y elija las opciones que correspondan.
14. Elija Crear el proyecto de compilación. En la página del proyecto de compilación, en Build history (Historial de compilaciones), elija Start build (Comenzar compilación) para ejecutar la compilación.
15. (Opcional) Siga las instrucciones de [Ejemplo: acelere su sitio web con Amazon CloudFront](#) en la Guía para desarrolladores de Amazon S3.

Ejemplo de varios orígenes de entrada y varios artefactos de salida

Puede crear un proyecto de AWS CodeBuild compilación con más de una fuente de entrada y más de un conjunto de artefactos de salida. En este ejemplo se muestra cómo configurar un proyecto de compilación que:

- Utiliza varios orígenes y repositorios de distintos tipos.
- Publica los artefactos de compilación en varios buckets de S3 en una sola compilación.

En este ejemplo se crea un proyecto de compilación que se utiliza para ejecutar una compilación. El ejemplo usa el archivo buildspec del proyecto de compilación para mostrar cómo incorporar más de un origen y crear más de un conjunto de artefactos.

1. Sube tus fuentes a uno o más repositorios de S3 buckets CodeCommit GitHub, GitHub Enterprise Server o Bitbucket.
2. Elija cuál es el origen principal. Esta es la fuente en la que CodeBuild busca y ejecuta tu archivo buildspec.

3. Cree un proyecto de compilación. Para obtener más información, consulte [Creación de un proyecto de compilación en AWS CodeBuild](#).
4. Crea tu proyecto de compilación, ejecuta la compilación y obtén información sobre la compilación.
5. Si utilizas el AWS CLI para crear el proyecto de compilación, la entrada del `create-project` comando con formato JSON podría tener un aspecto similar al siguiente:

```
{
  "name": "sample-project",
  "source": {
    "type": "S3",
    "location": "<bucket/sample.zip>"
  },
  "secondarySources": [
    {
      "type": "CODECOMMIT",
      "location": "https://git-codecommit.us-west-2.amazonaws.com/v1/repos/repo",
      "sourceIdentifier": "source1"
    },
    {
      "type": "GITHUB",
      "location": "https://github.com/awslabs/aws-codebuild-jenkins-plugin",
      "sourceIdentifier": "source2"
    }
  ],
  "secondaryArtifacts": [
    {
      "type": "S3",
      "location": "<output-bucket>",
      "artifactIdentifier": "artifact1"
    },
    {
      "type": "S3",
      "location": "<other-output-bucket>",
      "artifactIdentifier": "artifact2"
    }
  ],
  "environment": {
    "type": "LINUX_CONTAINER",
    "image": "aws/codebuild/standard:5.0",
    "computeType": "BUILD_GENERAL1_SMALL"
  },
}
```

```
"serviceRole": "arn:aws:iam::account-ID:role/role-name",
"encryptionKey": "arn:aws:kms:region-ID:account-ID:key/key-ID"
}
```

El origen principal se define en el atributo `source`. El resto de orígenes se denominan secundarios y aparecen en `secondarySources`. Todos los orígenes secundarios se instalan en su propio directorio. Este directorio se almacena en la variable de entorno integrada `CODEBUILD_SRC_DIR_`*sourceIdentifer*. Para obtener más información, consulte [Variables de entorno en los entornos de compilación](#).

El atributo `secondaryArtifacts` contiene una lista de definiciones de artefacto. Estos artefactos usan el bloque `secondary-artifacts` del archivo `buildspec` anidado en el bloque `artifacts`.

Los artefactos secundarios del archivo `buildspec` tienen la misma estructura que el resto de artefactos y se separan mediante el identificador correspondiente.

Note

En la [CodeBuild API](#), el atributo `artifactIdentifier` sobre un artefacto secundario es obligatorio en `y.CreateProject` `UpdateProject`. Debe utilizarse para hacer referencia a un artefacto secundario.

Si se utiliza la entrada con formato JSON anterior, el archivo `buildspec` del proyecto puede tener este aspecto:

```
version: 0.2

phases:
  install:
    runtime-versions:
      java: openjdk11
  build:
    commands:
      - cd $CODEBUILD_SRC_DIR_source1
      - touch file1
      - cd $CODEBUILD_SRC_DIR_source2
      - touch file2

artifacts:
```

```

files:
  - '**.*'
secondary-artifacts:
  artifact1:
    base-directory: $CODEBUILD_SRC_DIR_source1
    files:
      - file1
  artifact2:
    base-directory: $CODEBUILD_SRC_DIR_source2
    files:
      - file2

```

Puede reemplazar la versión del origen principal mediante la API con el atributo `sourceVersion` en `StartBuild`. Para reemplazar una o varias versiones de origen secundario, utilice el atributo `secondarySourceVersionOverride`.

La entrada con formato JSON del `start-build` comando en el podría tener el siguiente aspecto:
AWS CLI

```

{
  "projectName": "sample-project",
  "secondarySourcesVersionOverride": [
    {
      "sourceIdentifier": "source1",
      "sourceVersion": "codecommit-branch"
    },
    {
      "sourceIdentifier": "source2",
      "sourceVersion": "github-branch"
    }
  ]
}

```

Ejemplo de proyecto sin un origen

Puede configurar un CodeBuild proyecto eligiendo el tipo de **NO_SOURCE** fuente al configurar la fuente. Cuando el tipo de origen es **NO_SOURCE**, no puede especificar un archivo de especificación de compilación porque su proyecto no tiene un origen. En su lugar, debe especificar una cadena de especificación de compilación con formato YAML en el atributo `buildspec` de la entrada con formato JSON al comando de la CLI `create-project`. Podría ser como el siguiente:

```
{
```

```
"name": "project-name",
"source": {
  "type": "NO_SOURCE",
  "buildspec": "version: 0.2\n\nphases:\n  build:\n    commands:\n      - command"
},
"environment": {
  "type": "LINUX_CONTAINER",
  "image": "aws/codebuild/standard:5.0",
  "computeType": "BUILD_GENERAL1_SMALL",
},
"serviceRole": "arn:aws:iam::account-ID:role/role-name",
"encryptionKey": "arn:aws:kms:region-ID:account-ID:key/key-ID"
}
```

Para obtener más información, consulte [Crear un proyecto de compilación \(AWS CLI\)](#).

Para obtener información sobre cómo crear una canalización que utilice varias entradas de origen CodeBuild para crear varios artefactos de salida, consulte [AWS CodePipeline ejemplo de integración con CodeBuild múltiples fuentes de entrada y artefactos de salida](#).

Versiones en tiempo de ejecución incluidas en el ejemplo del archivo buildspec para CodeBuild

Si utiliza la versión de imagen de Amazon Linux 2 (AL2) estándar 1.0 u otra posterior, o la versión de imagen de Ubuntu estándar 2.0 o posterior, puede especificar uno o más entornos de tiempo de ejecución en la sección `runtime-versions` del archivo de especificación de compilación. Este ejemplo muestra cómo puede cambiar el runtime de su proyecto, especificar más de un runtime y especificar un runtime que depende de otro. Para obtener información sobre los runtimes compatibles, consulte [Imágenes de Docker proporcionadas por CodeBuild](#).

Note

Si utiliza Docker en el contenedor de compilación, la compilación debe ejecutarse en un modo con privilegios. Para obtener más información, consulte [Ejecutar una compilación en AWS CodeBuild](#) y [Creación de un proyecto de compilación en AWS CodeBuild](#).

Actualización de la versión del entorno de ejecución

Puede modificar el entorno de tiempo de ejecución que utiliza su proyecto por una nueva versión actualizando la sección `runtime-versions` del archivo de especificación de compilación. Los siguientes ejemplos muestran cómo se especifican las versiones 8 y 11 de Java.

- Una sección `runtime-versions` que especifica la versión 8 de Java:

```
phases:
  install:
    runtime-versions:
      java: corretto8
```

- Una sección `runtime-versions` que especifica la versión 11 de Java:

```
phases:
  install:
    runtime-versions:
      java: corretto11
```

Los siguientes ejemplos muestran cómo especificar diferentes versiones de Python utilizando la imagen de Ubuntu estándar 5.0 o la imagen de Amazon Linux 2 estándar 3.0:

- Una sección de `runtime-versions` que especifica Python versión 3.7:

```
phases:
  install:
    runtime-versions:
      python: 3.7
```

- Una sección de `runtime-versions` que especifica Python versión 3.8:

```
phases:
  install:
    runtime-versions:
      python: 3.8
```

Esta muestra presenta un proyecto que comienza con el tiempo de ejecución de la versión 8 de Java y, a continuación, se actualiza al tiempo de ejecución de la versión 10 de Java.

1. Descargue e instale Maven. Para obtener información, consulte [Downloading Apache Maven e Installing Apache Maven](#) en el sitio web de Apache Maven.
2. Cambie a un directorio vacío del equipo o instancia local y, a continuación, ejecute este comando Maven.

```
mvn archetype:generate "-DgroupId=com.mycompany.app" "-DartifactId=ROOT" "-DarchetypeArtifactId=maven-archetype-webapp" "-DinteractiveMode=false"
```

Si el comando se ejecuta correctamente, se creará esta estructura de directorios y archivos.

```
.
### ROOT
  ### pom.xml
  ### src
    ### main
      ### resources
      ### webapp
        ### WEB-INF
        #   ### web.xml
        ### index.jsp
```

3. Cree un archivo denominado `buildspec.yml` con el siguiente contenido. Guarde el archivo en el directorio *(root directory name)/my-web-app*.

```
version: 0.2

phases:
  install:
    runtime-versions:
      java: corretto8
  build:
    commands:
      - java -version
      - mvn package
artifacts:
  files:
    - '**/*'
  base-directory: 'target/my-web-app'
```

En el archivo `buildspec`:

- La sección `runtime-versions` especifica que el proyecto utiliza la versión 8 del tiempo de ejecución de Java.
- El comando `- java -version` muestra la versión de Java que utiliza su proyecto cuando se compila.

La estructura de archivos debe ser como la siguiente.

```
(root directory name)
### my-web-app
  ### src
  #   ### main
  #   ### resources
  #   ### webapp
  #       ### WEB-INF
  #           ### web.xml
  #               ### index.jsp
  ### buildspec.yml
  ### pom.xml
```

4. Sube el contenido del `my-web-app` directorio a un depósito de entrada de S3 o a un repositorio de CodeCommit GitHub Bitbucket.

Important

No cargue *(root directory name)* o *(root directory name)/my-web-app*, solo los directorios y archivos incluidos en *(root directory name)/my-web-app*. Si utiliza un bucket de entrada de S3, no olvide crear un archivo ZIP que contenga la estructura de directorios y los archivos para cargarlo después en el bucket de entrada. No añada *(root directory name)* o *(root directory name)/my-web-app* al archivo ZIP, solo los directorios y archivos incluidos en *(root directory name)/my-web-app*.

5. Abra la AWS CodeBuild consola en <https://console.aws.amazon.com/codesuite/codebuild/home>.
6. Cree un proyecto de compilación. Para obtener más información, consulte [Creación de un proyecto de compilación \(consola\)](#) y [Ejecutar una compilación \(consola\)](#). Deje todas las opciones con sus valores predeterminados, excepto las siguientes opciones.
 - Para Environment (Entorno):

- En Environment image (Imagen de entorno), elija Managed image (Imagen administrada).
 - En Operating system (Sistema operativo), elija Amazon Linux 2.
 - En Runtime(s) (Tiempo de ejecución), elija Standard (Estándar).
 - En Imagen, elija aws/codebuild/amazonlinux2-x86_64-standard:4.0.
7. Seleccione Iniciar la compilación.
 8. En Build configuration (Configuración de la compilación), acepte los valores predeterminados y elija Start build (Comenzar compilación).
 9. Una vez finalizada la compilación, consulte la salida de compilación en la pestaña Build logs (Registros de compilación). Debería ver una salida similar a esta:

```
[Container] Date Time Phase is DOWNLOAD_SOURCE
[Container] Date Time CODEBUILD_SRC_DIR=/codebuild/output/src460614277/src
[Container] Date Time YAML location is /codebuild/output/src460614277/src/buildspec.yml
[Container] Date Time Processing environment variables
[Container] Date Time Selecting 'java' runtime version 'corretto8' based on manual selections...
[Container] Date Time Running command echo "Installing Java version 8 ..."
Installing Java version 8 ...

[Container] Date Time Running command export JAVA_HOME="$JAVA_8_HOME"

[Container] Date Time Running command export JRE_HOME="$JRE_8_HOME"

[Container] Date Time Running command export JDK_HOME="$JDK_8_HOME"

[Container] Date Time Running command for tool_path in "$JAVA_8_HOME"/bin/*
"$JRE_8_HOME"/bin/*;
```

10. Actualice la sección runtime-versions con la versión 11 de Java:

```
install:
  runtime-versions:
    java: corretto11
```

11. Después de guardar el cambio, ejecute la compilación de nuevo y consulte la salida de compilación. Debería ver que la versión instalada de Java es la 11. Debería ver una salida similar a esta:

```
[Container] Date Time Phase is DOWNLOAD_SOURCE
[Container] Date Time CODEBUILD_SRC_DIR=/codebuild/output/src460614277/src
[Container] Date Time YAML location is /codebuild/output/src460614277/src/
buildspec.yml
[Container] Date Time Processing environment variables
[Container] Date Time Selecting 'java' runtime version 'corretto11' based on manual
selections...
Installing Java version 11 ...

[Container] Date Time Running command export JAVA_HOME="$JAVA_11_HOME"

[Container] Date Time Running command export JRE_HOME="$JRE_11_HOME"

[Container] Date Time Running command export JDK_HOME="$JDK_11_HOME"

[Container] Date Time Running command for tool_path in "$JAVA_11_HOME"/bin/*
"$JRE_11_HOME"/bin/*;
```

Especificación de dos entornos de ejecución

Puede especificar más de un tiempo de ejecución en el mismo proyecto de CodeBuild compilación. Este proyecto de muestra utiliza dos archivos de código fuente: uno que utiliza el tiempo de ejecución de Go y otro que utiliza el tiempo de ejecución de Node.js.

1. Cree un directorio llamado `my-source`.
2. En el directorio `my-source`, cree un directorio llamado `golang-app`.
3. Cree un archivo denominado `hello.go` con el siguiente contenido. Guarde el archivo en el directorio `golang-app`.

```
package main
import "fmt"

func main() {
    fmt.Println("hello world from golang")
    fmt.Println("1+1 =", 1+1)
    fmt.Println("7.0/3.0 =", 7.0/3.0)
    fmt.Println(true && false)
    fmt.Println(true || false)
    fmt.Println(!true)
    fmt.Println("good bye from golang")
}
```

```
}
```

4. En el directorio `my-source`, cree un directorio llamado `nodejs-app`. Debe estar al mismo nivel que el directorio `golang-app`.
5. Cree un archivo denominado `index.js` con el siguiente contenido. Guarde el archivo en el directorio `nodejs-app`.

```
console.log("hello world from nodejs");
console.log("1+1 =" + (1+1));
console.log("7.0/3.0 =" + 7.0/3.0);
console.log(true && false);
console.log(true || false);
console.log(!true);
console.log("good bye from nodejs");
```

6. Cree un archivo denominado `package.json` con el siguiente contenido. Guarde el archivo en el directorio `nodejs-app`.

```
{
  "name": "mycompany-app",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"run some tests here\""
  },
  "author": "",
  "license": "ISC"
}
```

7. Cree un archivo denominado `buildspec.yml` con el siguiente contenido. Guarde el archivo en el directorio `my-source`, al mismo nivel que los directorios `nodejs-app` y `golang-app`. La sección `runtime-versions` especifica los tiempos de ejecución de la versión 12 de Node.js y la versión 1.13 de Go.

```
version: 0.2

phases:
  install:
    runtime-versions:
      golang: 1.13
      nodejs: 12
```

```
build:
  commands:
    - echo Building the Go code...
    - cd $CODEBUILD_SRC_DIR/golang-app
    - go build hello.go
    - echo Building the Node code...
    - cd $CODEBUILD_SRC_DIR/nodejs-app
    - npm run test
artifacts:
  secondary-artifacts:
    golang_artifacts:
      base-directory: golang-app
      files:
        - hello
    nodejs_artifacts:
      base-directory: nodejs-app
      files:
        - index.js
        - package.json
```

8. La estructura de archivos debe ser como la siguiente.

```
my-source
### golang-app
#   ### hello.go
### nodejs.app
#   ### index.js
#   ### package.json
### buildspec.yml
```

9. Sube el contenido del `my-source` directorio a un depósito de entrada de S3 o a un CodeCommit repositorio de Bitbucket. GitHub

Important

Si utiliza un bucket de entrada de S3, no olvide crear un archivo ZIP que contenga la estructura de directorios y los archivos para cargarlo después en el bucket de entrada. No añada `my-source` al archivo ZIP, solo los directorios y archivos incluidos en `my-source`.

10. Abre la AWS CodeBuild consola en <https://console.aws.amazon.com/codesuite/codebuild/home>.

11. Cree un proyecto de compilación. Para obtener más información, consulte [Creación de un proyecto de compilación \(consola\)](#) y [Ejecutar una compilación \(consola\)](#). Deje todas las opciones con sus valores predeterminados, excepto las siguientes opciones.
 - Para Environment (Entorno):
 - En Environment image (Imagen de entorno), elija Managed image (Imagen administrada).
 - En Operating system (Sistema operativo), elija Amazon Linux 2.
 - En Runtime(s) (Tiempo de ejecución), elija Standard (Estándar).
 - En Imagen, elija aws/codebuild/amazonlinux2-x86_64-standard:4.0.
12. Elija Crear el proyecto de compilación.
13. Seleccione Iniciar la compilación.
14. En Build configuration (Configuración de la compilación), acepte los valores predeterminados y elija Start build (Comenzar compilación).
15. Una vez finalizada la compilación, consulte la salida de compilación en la pestaña Build logs (Registros de compilación). Debería ver un resultado similar a este. Muestra la salida de los tiempos de ejecución de Go y Node.js. También muestra la salida de las aplicaciones de Go y Node.js.

```
[Container] Date Time Processing environment variables
[Container] Date Time Selecting 'golang' runtime version '1.13' based on manual
  selections...
[Container] Date Time Selecting 'nodejs' runtime version '12' based on manual
  selections...
[Container] Date Time Running command echo "Installing Go version 1.13 ..."
Installing Go version 1.13 ...

[Container] Date Time Running command echo "Installing Node.js version 12 ..."
Installing Node.js version 12 ...

[Container] Date Time Running command n $NODE_12_VERSION
  installed : v12.20.1 (with npm 6.14.10)

[Container] Date Time Moving to directory /codebuild/output/src819694850/src
[Container] Date Time Registering with agent
[Container] Date Time Phases found in YAML: 2
[Container] Date Time  INSTALL: 0 commands
[Container] Date Time  BUILD: 1 commands
[Container] Date Time Phase complete: DOWNLOAD_SOURCE State: SUCCEEDED
[Container] Date Time Phase context status code:  Message:
```

```
[Container] Date Time Entering phase INSTALL
[Container] Date Time Phase complete: INSTALL State: SUCCEEDED
[Container] Date Time Phase context status code: Message:
[Container] Date Time Entering phase PRE_BUILD
[Container] Date Time Phase complete: PRE_BUILD State: SUCCEEDED
[Container] Date Time Phase context status code: Message:
[Container] Date Time Entering phase BUILD
[Container] Date Time Running command echo Building the Go code...
Building the Go code...

[Container] Date Time Running command cd $CODEBUILD_SRC_DIR/golang-app

[Container] Date Time Running command go build hello.go

[Container] Date Time Running command echo Building the Node code...
Building the Node code...

[Container] Date Time Running command cd $CODEBUILD_SRC_DIR/nodejs-app

[Container] Date Time Running command npm run test

> mycompany-app@1.0.0 test /codebuild/output/src924084119/src/nodejs-app
> echo "run some tests here"

run some tests here
```

Ejemplo de la versión fuente con AWS CodeBuild

En este ejemplo se muestra cómo especificar una versión de su origen con un formato que no sea un ID de confirmación (también conocido como SHA de confirmación). Puede especificar la versión del origen de las siguientes formas:

- En el caso de un proveedor de origen de Amazon S3, utilice el ID de versión del objeto que representa el archivo ZIP de entrada de la compilación.
- Para CodeCommit Bitbucket y GitHub Enterprise Server GitHub, usa una de las siguientes opciones:
 - Solicitud de extracción como referencia de solicitud de extracción (por ejemplo, `refs/pull/1/head`).
 - Ramificación como nombre de ramificación.

- ID de confirmación.
- Etiqueta.
- Referencia e ID de confirmación. La referencia puede ser uno de los siguientes elementos:
 - Una etiqueta (por ejemplo, `refs/tags/mytagv1.0^{full-commit-SHA}`).
 - Una ramificación (por ejemplo, `refs/heads/mydevbranch^{full-commit-SHA}`).
 - Una solicitud de extracción (por ejemplo, `refs/pull/1/head^{full-commit-SHA}`).
- Para autogestionados GitLab y GitLab autogestionados, usa uno de los siguientes:
 - Ramificación como nombre de ramificación.
 - ID de confirmación.
 - Etiqueta.

Note

Puedes especificar la versión de una fuente de solicitudes de extracción solo si tu repositorio es GitHub GitHub Enterprise Server.

Si utiliza una referencia y un ID de confirmación para especificar una versión, la fase `DOWNLOAD_SOURCE` de la compilación será más rápida que si proporciona solo la versión. Esto se debe a que, al añadir una referencia, CodeBuild no es necesario descargar todo el repositorio para encontrar la confirmación.

- Puede especificar una versión de origen con solo un ID de confirmación, como por ejemplo `12345678901234567890123467890123456789`. Si lo haces, CodeBuild debes descargar todo el repositorio para encontrar la versión.
- Puede especificar una versión de origen con una referencia y un ID de confirmación en este formato: `refs/heads/branchname^{full-commit-SHA}` (por ejemplo, `refs/heads/main^{12345678901234567890123467890123456789}`). Si lo hace, CodeBuild descarga solo la rama especificada para buscar la versión.

Note

Para acelerar la `DOWNLOAD_SOURCE` fase de compilación, también puedes establecer la profundidad de clonación de Git en un número bajo. CodeBuild descarga menos versiones de tu repositorio.

Para especificar una versión de GitHub repositorio con un ID de confirmación

1. Abre la AWS CodeBuild consola en <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Cree un proyecto de compilación. Para obtener más información, consulte [Creación de un proyecto de compilación \(consola\)](#) y [Ejecutar una compilación \(consola\)](#). Deje todas las opciones con sus valores predeterminados, excepto las siguientes opciones:
 - En Source (Código fuente):
 - En Proveedor de código fuente, elija GitHub. Si no está conectado a GitHub, siga las instrucciones para conectarse.
 - En Repository (Repositorio), elija Public repository (Repositorio público).
 - En URL de repositorio, especifique **`https://github.com/aws/aws-sdk-ruby.git`**
 - En Environment (Entorno):
 - En Environment image (Imagen de entorno), elija Managed image (Imagen administrada).
 - En Operating system (Sistema operativo), elija Amazon Linux 2.
 - En Runtime(s) (Tiempo de ejecución), elija Standard (Estándar).
 - En Imagen, elija `aws/codebuild/amazonlinux2-x86_64-standard:4.0`.
3. En Build specifications (Especificaciones de compilación), elija Insert build commands (Insertar comandos de compilación) y, a continuación, elija Switch to editor (Cambiar a editor).
4. En Build commands (Comandos de compilación), reemplace el texto del marcador de posición por lo siguiente:

```
version: 0.2

phases:
  install:
    runtime-versions:
      ruby: 2.6
  build:
```

```
commands:
  - echo $CODEBUILD_RESOLVED_SOURCE_VERSION
```

La sección `runtime-versions` es obligatoria si utiliza la imagen estándar de Ubuntu 2.0. Aquí, se especifica el tiempo de ejecución de la versión de Ruby 2.6, pero puede usar cualquier tiempo de ejecución. El comando `echo` muestra la versión del código de origen almacenado en la variable de entorno de `CODEBUILD_RESOLVED_SOURCE_VERSION`.

5. En Build configuration (Configuración de la compilación), acepte los valores predeterminados y elija Start build (Comenzar compilación).
6. En Versión de origen, especifique **046e8b67481d53bdc86c3f6affdd5d1afae6d369**. Este es el SHA de una confirmación en el repositorio `https://github.com/aws/aws-sdk-ruby.git`.
7. Seleccione Iniciar la compilación.
8. Cuando se complete la compilación, debería ver lo siguiente:
 - En la pestaña Registros de compilación, la versión del origen de proyecto que se ha utilizado. A continuación se muestra un ejemplo.

```
[Container] Date Time Running command echo $CODEBUILD_RESOLVED_SOURCE_VERSION
046e8b67481d53bdc86c3f6affdd5d1afae6d369
```

```
[Container] Date Time Phase complete: BUILD State: SUCCEEDED
```

- En la pestaña Variables de entorno, Versión de origen resuelta coincide con el ID de confirmación utilizado para crear la compilación.
- En la pestaña Detalles de fase, la duración de la fase `DOWNLOAD_SOURCE`.

Estos pasos muestran cómo crear una compilación con la misma versión del origen. Esta vez, la versión del origen se especifica mediante una referencia con el ID de confirmación.

Para especificar una versión de GitHub repositorio con un ID de confirmación y una referencia

1. En el panel de navegación izquierdo, elija Proyectos de compilación y, a continuación, elija el proyecto que creó anteriormente.
2. Seleccione Iniciar la compilación.
3. En Versión de origen, especifique **refs/heads/main^{046e8b67481d53bdc86c3f6affdd5d1afae6d369}**. Es el mismo ID de confirmación

y una referencia a una ramificación en el formato `refs/heads/branchname^{full-commit-SHA}`.

4. Seleccione Iniciar la compilación.
5. Cuando se complete la compilación, debería ver lo siguiente:
 - En la pestaña Registros de compilación, la versión del origen de proyecto que se ha utilizado. A continuación se muestra un ejemplo.

```
[Container] Date Time Running command echo $CODEBUILD_RESOLVED_SOURCE_VERSION
046e8b67481d53bdc86c3f6affdd5d1afae6d369
```

```
[Container] Date Time Phase complete: BUILD State: SUCCEEDED
```

- En la pestaña Variables de entorno, Versión de origen resuelta coincide con el ID de confirmación utilizado para crear la compilación.
- En la pestaña Detalles de fase, la duración de la fase `DOWNLOAD_SOURCE` debe ser inferior a la duración cuando utilizó solo el ID de confirmación para especificar la versión del origen.

Ejemplos de repositorios fuente de terceros para CodeBuild

Temas

- [Ejemplo de filtro webhook y pull request de Bitbucket para CodeBuild](#)
- [GitHub Ejemplo de Enterprise Server para CodeBuild](#)
- [GitHub ejemplo de filtro pull request y webhook para CodeBuild](#)

Ejemplo de filtro webhook y pull request de Bitbucket para CodeBuild

AWS CodeBuild admite webhooks cuando el repositorio de origen es Bitbucket. Esto significa que, en el caso de un proyecto de CodeBuild compilación cuyo código fuente esté almacenado en un repositorio de Bitbucket, los webhooks se pueden usar para reconstruir el código fuente cada vez que se introduzca un cambio de código en el repositorio. Para obtener más información, consulte [Eventos de webhooks de Bitbucket](#).

En este ejemplo se muestra cómo crear una solicitud de extracción mediante un repositorio de Bitbucket. También te muestra cómo usar un webhook de Bitbucket para activar CodeBuild la creación de una compilación de un proyecto.

 Note

Al usar webhooks, es posible que un usuario active una compilación de forma inesperada. Para mitigar este riesgo, consulte [Prácticas recomendadas para utilizar webhooks](#).

Temas

- [Requisitos previos](#)
- [Creación de un proyecto de compilación con Bitbucket como repositorio de origen y habilitación de webhooks](#)
- [Activación de una compilación con un webhook de Bitbucket](#)

Requisitos previos

Para ejecutar este ejemplo, debes conectar tu AWS CodeBuild proyecto con tu cuenta de Bitbucket.

 Note

CodeBuild ha actualizado sus permisos con Bitbucket. Si anteriormente habías conectado tu proyecto a Bitbucket y ahora recibes un error de conexión con Bitbucket, debes volver a conectarte para conceder el CodeBuild permiso de gestión de tus webhooks.

Creación de un proyecto de compilación con Bitbucket como repositorio de origen y habilitación de webhooks

En los siguientes pasos, se describe cómo crear un AWS CodeBuild proyecto con Bitbucket como repositorio de origen y cómo habilitar los webhooks.

1. [Abre la AWS CodeBuild consola en https://console.aws.amazon.com/codesuite/codebuild/home](https://console.aws.amazon.com/codesuite/codebuild/home).
2. Si aparece una página de CodeBuild información, elija Crear proyecto de compilación. De lo contrario, en el panel de navegación, expanda Compilar, elija Proyectos de compilación y, a continuación, elija Crear proyecto de compilación.
3. Elija Crear el proyecto de compilación.
4. En Project configuration (Configuración del proyecto):

Nombre del proyecto

Introduzca un nombre para este proyecto de compilación. Los nombres de los proyectos de compilación deben ser únicos en cada AWS cuenta. También puede introducir una descripción opcional del proyecto de compilación para ayudar a otros usuarios a entender para qué se utiliza el proyecto.

5. En Source (Código fuente):

Proveedor de fuentes

Seleccione Bitbucket. Siga las instrucciones para conectarse (o volver a conectarse) a Bitbucket y, a continuación, elija Autorizar.

Repositorio

Seleccione Repositorio en mi cuenta de Bitbucket.

Si no se ha conectado previamente a su cuenta de Bitbucket, introduzca su nombre de usuario y contraseña de aplicación de Bitbucket y seleccione Guardar las credenciales de Bitbucket.

Repositorio de Bitbucket

Introduzca la URL de su repositorio de Bitbucket.

6. En Eventos de webhook de origen principal, seleccione lo siguiente.

Note

La sección Eventos de webhook de origen principal solo está visible si se ha seleccionado Repositorio en mi cuenta de Bitbucket en el paso anterior.

1. Cuando cree el proyecto, seleccione Rebuild every time a code change is pushed to this repository (Volver a compilar cada vez que se inserte un cambio de código en este repositorio).
2. En Event type (Tipo de evento), seleccione uno o varios eventos.
3. Para filtrar en función de cuándo un evento va a desencadenar una compilación, en Start a build under these conditions (Iniciar una compilación en estas condiciones), añada uno o varios filtros opcionales.

4. Para filtrar en función de cuándo no se va a desencadenar un evento, en Don't start a build under these conditions (No iniciar una compilación en estas condiciones), añada uno o varios filtros opcionales.
5. Seleccione Añadir grupo de filtros para añadir otro grupo de filtros.

Para obtener más información sobre los filtros y tipos de eventos de webhook de Bitbucket, consulte [Eventos de webhooks de Bitbucket](#).

7. En Environment (Entorno):

Imagen del entorno

Seleccione una de las siguientes opciones:

Para usar una imagen de Docker gestionada por AWS CodeBuild:

Elija Imagen administrada y después elija opciones en Sistema operativo, Tiempo de ejecución, Imagen y Versión de imagen. Realice una selección en Environment type (Tipo de entorno) si está disponible.

Para utilizar otra imagen de Docker:

Elija Imagen personalizada. En Tipo de entorno, elija ARM, Linux, GPU de Linux o Windows. Si elige Otro registro, en URL de registro externo, introduzca el nombre y etiqueta de la imagen de Docker en Docker Hub, usando el formato *docker repository/docker image name*. Si ha seleccionado Amazon ECR, use Repositorio de Amazon ECR e Imagen de Amazon ECR para elegir la imagen de Docker en su cuenta de AWS .

Para usar una imagen de Docker privada:

Elija Imagen personalizada. En Tipo de entorno, elija ARM, Linux, GPU de Linux o Windows. En Image registry (Registro de imagen), elija Other registry (Otro registro) y, a continuación, introduzca el ARN de las credenciales para su imagen de Docker privada. Es necesario crear las credenciales con Secrets Manager. Para obtener más información, consulte [¿Qué es? AWS Secrets Manager](#) en la Guía AWS Secrets Manager del usuario.

Rol de servicio

Seleccione una de las siguientes opciones:

- Si no tiene un rol de CodeBuild servicio, elija Nuevo rol de servicio. Escriba un nombre para el nuevo rol en el campo Role Name.

- Si tiene un rol CodeBuild de servicio, elija el rol de servicio existente. En ARN de rol, seleccione el rol de servicio.

 Note

Cuando utilizas la consola para crear o actualizar un proyecto de compilación, puedes crear un rol de CodeBuild servicio al mismo tiempo. De forma predeterminada, el rol funciona solamente con ese proyecto de compilación. Si utiliza la consola para asociar este rol de servicio a otro proyecto de compilación, el rol se actualizará para que funcione con el otro proyecto de compilación. Un rol de servicio puede funcionar con hasta diez proyectos de compilación.

8. En Archivo de especificación de compilación, realice alguna de las operaciones siguientes:
 - Seleccione Usar un archivo de especificaciones de compilación para utilizar el archivo `buyildspec.yml` que figura en el directorio raíz del código fuente.
 - Elija Insertar comandos de compilación para usar la consola para insertar comandos de compilación.

Para obtener más información, consulte [Referencia de la especificación de compilación](#).

9. En Artifacts (Artefactos):

Tipo

Seleccione una de las siguientes opciones:

- Si no desea crear ningún artefacto de salida de la compilación, elija No artifacts (Sin artefactos).
- Para almacenar la salida de la compilación en un bucket de S3, seleccione Amazon S3 y, a continuación, haga lo siguiente:
 - Si desea utilizar su nombre de proyecto para el archivo ZIP o carpeta de salida de la compilación, deje en blanco Nombre. De lo contrario, escriba el nombre. De forma predeterminada, el nombre del artefacto es el nombre del proyecto. Si desea usar otro nombre, escríbalo en el cuadro de nombre de artefactos. Si desea producir un archivo ZIP, incluya la extensión zip.
 - En Bucket name (Nombre del bucket), seleccione el nombre del bucket de salida.

- Si eligió Insert build commands (Insertar comandos de compilación) anteriormente en este procedimiento, en Output files (Archivos de salida), escriba las ubicaciones de los archivos de la compilación que desea incluir en el archivo ZIP o carpeta de salida de la compilación. Si hay varias ubicaciones, separe cada una de las ubicaciones con una coma (por ejemplo, `appspec.yml`, `target/my-app.jar`). Para obtener más información, consulte la descripción de files en [Sintaxis de buildspec](#).

Configuración adicional

Expande Additional configuration (Configuración adicional) y establezca las opciones según sea necesario.

10. Elija Crear el proyecto de compilación. En la página Review (Revisar), elija Start build (Comenzar compilación) para ejecutar la compilación.

Activación de una compilación con un webhook de Bitbucket

En el caso de un proyecto que usa webhooks de Bitbucket, AWS CodeBuild crea una compilación cuando el repositorio de Bitbucket detecta un cambio en el código fuente.

1. [Abre la AWS CodeBuild consola en https://console.aws.amazon.com/codesuite/codebuild/home](https://console.aws.amazon.com/codesuite/codebuild/home).
2. En el panel de navegación, elija Build projects (Proyectos de compilación) y, a continuación, elija un proyecto asociado a un repositorio de Bitbucket con webhooks. Para obtener información sobre cómo crear un proyecto con webhook de Bitbucket, consulte [the section called "Creación de un proyecto de compilación con Bitbucket como repositorio de origen y habilitación de webhooks"](#).
3. Realice algunos cambios en el código en el repositorio de Bitbucket del proyecto.
4. Cree una solicitud de extracción en su repositorio de Bitbucket. Para obtener más información, consulte el artículo sobre [cómo crear una solicitud de extracción](#).
5. En la página de webhooks de Bitbucket, elija View request (Ver solicitud) para ver una lista de eventos recientes.
6. Seleccione Ver detalles para ver los detalles de la respuesta devuelta por CodeBuild. Debe ser similar al siguiente:

```
"response": "Webhook received and build started: https://us-east-1.console.aws.amazon.com/codebuild/home..."
"statusCode": 200
```

7. Vaya a la página de solicitudes de extracción de Bitbucket para ver el estado de la compilación.

GitHub Ejemplo de Enterprise Server para CodeBuild

AWS CodeBuild admite GitHub Enterprise Server como repositorio de origen. En este ejemplo se muestra cómo configurar CodeBuild los proyectos cuando el repositorio de GitHub Enterprise Server tiene un certificado instalado. También se muestra cómo habilitar los webhooks para que CodeBuild se reconstruya el código fuente cada vez que se introduzca un cambio de código en el repositorio de GitHub Enterprise Server.

Requisitos previos

1. Genera un token de acceso personal para tu proyecto. CodeBuild Le recomendamos que cree un usuario GitHub empresarial y genere un token de acceso personal para este usuario. Cópelo en el portapapeles para que pueda usarlo al crear su CodeBuild proyecto. Para obtener más información, consulte [Crear un token de acceso personal para la línea de comandos](#) en el sitio web de GitHub ayuda.

Cuando cree el token de acceso personal, incluya el ámbito del repositorio en la definición.

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes.](#)

<input checked="" type="checkbox"/>	repo	Full control of private repositories
<input checked="" type="checkbox"/>	repo:status	Access commit status
<input checked="" type="checkbox"/>	repo_deployment	Access deployment status
<input checked="" type="checkbox"/>	public_repo	Access public repositories

2. Descargue su certificado desde GitHub Enterprise Server. CodeBuild utiliza el certificado para establecer una conexión SSL de confianza con el repositorio.

Clientes Linux/macOS:

En una ventana de terminal de , ejecute el siguiente comando:

```
echo -n | openssl s_client -connect HOST:PORTNUMBER \  
| sed -ne '/-BEGIN CERTIFICATE-/,/-END CERTIFICATE-/p' > /folder/filename.pem
```

Sustituya los marcadores de posición del comando por los siguientes valores:

HOST. La dirección IP de su repositorio de GitHub Enterprise Server.

PORTNUMBER. El número de puerto que se utiliza para conectarse (por ejemplo, 443).

folder. La carpeta en la que se ha descargado el certificado.

nombre de archivo. El nombre de archivo del archivo de certificado.

 Important

Guarde el certificado como un archivo .pem.

Clientes de Windows:

Utilice su navegador para descargar el certificado de GitHub Enterprise Server. Para ver los detalles del certificado del sitio, seleccione el icono del candado. Para obtener información sobre cómo exportar el certificado, consulte la documentación del navegador.

 Important

Guarde el certificado como un archivo .pem.

3. Cargue el archivo de certificado en un bucket de S3. Para obtener información acerca de cómo crear un bucket de S3, consulte [¿Cómo puedo crear un bucket de S3?](#) Para obtener información acerca de cómo cargar objetos en un bucket de S3, consulte [¿Cómo puedo cargar archivos y carpetas en un bucket?](#)

 Note

Este depósito debe estar en la misma AWS región que sus compilaciones. Por ejemplo, si indicas CodeBuild que se ejecute una compilación en la región EE.UU. Este (Ohio), el bucket debe estar en la región EE.UU. Este (Ohio).

Cree un proyecto de compilación con GitHub Enterprise Server como repositorio de origen y habilite los webhooks (consola)

1. Abra la AWS CodeBuild consola en <https://console.aws.amazon.com/codesuite/codebuild/home>.

2. Si aparece una página de CodeBuild información, elija Crear proyecto de compilación. De lo contrario, en el panel de navegación, expanda Compilar, elija Proyectos de compilación y, a continuación, elija Crear proyecto de compilación.
3. En Project name (Nombre de proyecto), escriba un nombre para este proyecto de compilación. Los nombres de los proyectos de compilación deben ser únicos en cada AWS cuenta. También puede introducir una descripción opcional del proyecto de compilación para ayudar a otros usuarios a entender para qué se utiliza el proyecto.
4. En Source, en Source provider, selecciona GitHub Enterprise.
 - En Personal Access Token, pegue el token que ha copiado en el portapapeles y elija Save Token. En URL del repositorio, introduzca la URL del repositorio de GitHub Enterprise Server.

 Note

Solo es necesario introducir y guardar el token de acceso personal una vez. Todos los AWS CodeBuild proyectos futuros utilizan este token.

- En Repository URL (URL del repositorio), escriba la ruta al repositorio, incluido el nombre del repositorio.
- Expanda Configuración adicional.
- Seleccione Rebuild every time a code change is pushed to this repository (Volver a compilar cada vez que un cambio de código se inserte en el repositorio) para recompilar cada vez que se inserte un cambio de código en este repositorio.
- Seleccione Habilitar SSL inseguro para ignorar las advertencias de SSL mientras se conecta al repositorio de proyectos de GitHub Enterprise Server.

 Note

Le recomendamos que solamente utilice Enable insecure SSL (Habilitar SSL no seguro) para realizar pruebas. No debe utilizarse en un entorno de producción.

Source Add source

Source 1 - Primary

Source provider

GitHub Enterprise ▼

Repository URL

https://<host-name>/<user-name>/<repository-name>

Disconnect GitHub Enterprise account

▼ **Additional configuration**
Git clone depth, Insecure SSL

Git clone depth - *optional*

1 ▼

Webhook - *optional*

Rebuild every time a code change is pushed to this repository

Branch filter - *optional*

Enter a regular expression

Insecure SSL - *optional*
Enable this flag to ignore SSL warnings while connecting to project source.

Enable insecure SSL

5. En Environment (Entorno):

En Environment image (Imagen del entorno), realice alguna de las siguientes operaciones:

- Para usar una imagen de Docker gestionada por AWS CodeBuild, selecciona Imagen gestionada y, a continuación, selecciona Sistema operativo, Tiempo de ejecución, Imagen y Versión de la imagen. Realice una selección en Environment type (Tipo de entorno) si está disponible.

- Para usar otra imagen de Docker, elija Custom image (Imagen personalizada). En Tipo de entorno, elija ARM, Linux, GPU de Linux o Windows. Si elige Otro registro, en URL de registro externo, introduzca el nombre y etiqueta de la imagen de Docker en Docker Hub, usando el formato *docker repository/docker image name*. Si elige Amazon ECR, utilice el repositorio de Amazon ECR y la imagen de Amazon ECR para elegir la imagen de Docker en su cuenta. AWS
 - Para usar una imagen de Docker privada, elija Imagen personalizada. En Tipo de entorno, elija ARM, Linux, GPU de Linux o Windows. En Image registry (Registro de imagen), elija Other registry (Otro registro) y, a continuación, introduzca el ARN de las credenciales para su imagen de Docker privada. Es necesario crear las credenciales con Secrets Manager. Para obtener más información, consulte [AWS Secrets Manager](#) en la Guía del usuario de AWS Secrets Manager .
6. En Service role (Rol de servicio), realice una de las operaciones siguientes:
- Si no tiene un rol de CodeBuild servicio, elija Nuevo rol de servicio. Escriba un nombre para el nuevo rol en el campo Role Name.
 - Si tiene un rol CodeBuild de servicio, elija el rol de servicio existente. En ARN de rol, seleccione el rol de servicio.

 Note

Cuando utilizas la consola para crear o actualizar un proyecto de compilación, puedes crear un rol de CodeBuild servicio al mismo tiempo. De forma predeterminada, el rol funciona solamente con ese proyecto de compilación. Si utiliza la consola para asociar este rol de servicio a otro proyecto de compilación, el rol se actualizará para que funcione con el otro proyecto de compilación. Un rol de servicio puede funcionar con hasta diez proyectos de compilación.

7. Expanda Configuración adicional.

Si quieres CodeBuild trabajar con tu VPC:

- Para la VPC, elija el ID de VPC que utilice. CodeBuild
- Para las subredes de VPC, elija las subredes que incluyen los recursos que utiliza. CodeBuild
- Para los grupos de seguridad de VPC, elija los grupos de seguridad que se CodeBuild utilizan para permitir el acceso a los recursos de las VPC.

Para obtener más información, consulte [Úselo AWS CodeBuild con Amazon Virtual Private Cloud](#).

8. En Archivo de especificación de compilación, realice alguna de las operaciones siguientes:

- Seleccione Usar un archivo de especificaciones de compilación para utilizar el archivo `buyildspec.yml` que figura en el directorio raíz del código fuente.
- Elija Insertar comandos de compilación para usar la consola para insertar comandos de compilación.

Para obtener más información, consulte [Referencia de la especificación de compilación](#).

9. En Artifacts (Artefactos), en Type (Tipo), realice una de las siguientes operaciones:

- Si no desea crear ningún artefacto de salida de la compilación, elija No artifacts (Sin artefactos).
- Para almacenar la salida de la compilación en un bucket de S3, seleccione Amazon S3 y, a continuación, haga lo siguiente:
 - Si desea utilizar su nombre de proyecto para el archivo ZIP o carpeta de salida de la compilación, deje en blanco Nombre. De lo contrario, escriba el nombre. De forma predeterminada, el nombre del artefacto es el nombre del proyecto. Si desea usar otro nombre, escríbalo en el cuadro de nombre de artefactos. Si desea producir un archivo ZIP, incluya la extensión `zip`.
 - En Bucket name (Nombre del bucket), seleccione el nombre del bucket de salida.
 - Si eligió Insert build commands (Insertar comandos de compilación) anteriormente en este procedimiento, en Output files (Archivos de salida), escriba las ubicaciones de los archivos de la compilación que desea incluir en el archivo ZIP o carpeta de salida de la compilación. Si hay varias ubicaciones, separe cada una de las ubicaciones con una coma (por ejemplo, `appspect.yml, target/my-app.jar`). Para obtener más información, consulte la descripción de `files` en [Sintaxis de buildspec](#).

10. En Tipo de caché, seleccione una de las opciones siguientes:

- Si no desea usar una memoria caché, seleccione Sin caché.
- Si prefiere utilizar una caché de Amazon S3, seleccione Amazon S3 y haga lo siguiente:
 - En Bucket, señeccopme el nombre del bucket de S3 donde se almacena la caché.

- (Opcional) Para Prefijo de ruta de caché, escriba un prefijo de ruta de Amazon S3. El valor Prefijo de ruta de caché es similar a un nombre de directorio. Le permite almacenar la caché en el mismo directorio en un bucket.

 Important

No añada una barra (/) al final del prefijo de ruta.

- Si desea utilizar una caché local, seleccione Local y elija uno o varios modos de caché local.

 Note

El modo de caché de capas de Docker solo está disponible para Linux. Si lo selecciona, el proyecto deberá ejecutarse en modo con privilegios.

El uso de memoria caché ahorra mucho tiempo de compilación, ya que algunas partes reutilizables del entorno de compilación se almacenan en ella y se usan en las distintas compilaciones. Para obtener información acerca de cómo especificar una caché en el archivo de especificación de compilación, consulte [Sintaxis de buildspec](#). Para obtener más información acerca del almacenamiento en caché, consulte [Almacenamiento en caché de compilaciones de AWS CodeBuild](#).

11. Elija Crear el proyecto de compilación. En la página del proyecto de compilación, elija Start build (Iniciar compilación).
12. Si ha habilitado los webhooks en Source (Código fuente), aparecerá el cuadro de diálogo Create webhook (Crear webhook) con los valores para Payload URL (URL de carga) y Secret (Secreto).

 Important

El cuadro de diálogo Create webhook solo aparecerá una vez. Copie la URL de carga y la clave secreta. Los necesitará cuando añada un webhook a Enterprise Server. GitHub Si necesitas volver a generar una URL de carga útil y una clave secreta, primero debes eliminar el webhook del repositorio de GitHub Enterprise Server. En el CodeBuild proyecto, desactive la casilla de verificación Webhook y, a continuación, seleccione Guardar. A continuación, puedes crear o actualizar un CodeBuild proyecto con la casilla

de verificación Webhook seleccionada. El cuadro de diálogo Create webhook aparecerá de nuevo.

13. En GitHub Enterprise Server, elige el repositorio en el que está almacenado tu CodeBuild proyecto.
14. Elija Settings (Configuración), elija Hooks & services (Enlaces y servicios) y, a continuación, elija Add webhook (Añadir webhook).
15. Introduzca la URL de carga y la clave secreta, acepte los valores predeterminados para el resto de los campos y elija Add webhook.

requests 0 Projects 0 Wiki Pulse Graphs Settings

Webhooks / Add webhook

We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, x-www-form-urlencoded, etc). More information can be found in [our developer documentation](#).

Payload URL *

Content type

application/json

Secret

By default, we verify SSL certificates when delivering payloads. [Disable SSL verification](#)

Which events would you like to trigger this webhook?

Just the push event.

Send me everything.

Let me select individual events.

Active
We will deliver event details when this hook is triggered.

[Add webhook](#)

16. Regrese a su CodeBuild proyecto. Cierre el cuadro de diálogo Create webhook y elija Start build.

GitHub ejemplo de filtro pull request y webhook para CodeBuild

AWS CodeBuild admite webhooks cuando el repositorio fuente lo es GitHub. Esto significa que, en el caso de un proyecto de CodeBuild compilación que tiene su código fuente almacenado en un

GitHub repositorio, los webhooks se pueden usar para reconstruir el código fuente cada vez que se introduce un cambio de código en el repositorio. Para ver CodeBuild ejemplos, consulta [AWS CodeBuild Ejemplos](#).

 Note

Al usar webhooks, es posible que un usuario active una compilación de forma inesperada. Para mitigar este riesgo, consulte [Prácticas recomendadas para utilizar webhooks](#).

Cree un proyecto de compilación GitHub como repositorio de origen y habilite los webhooks (consola)

1. Abra la AWS CodeBuild consola en <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Si aparece una página de CodeBuild información, elija Crear proyecto de compilación. De lo contrario, en el panel de navegación, expanda Compilar, elija Proyectos de compilación y, a continuación, elija Crear proyecto de compilación.
3. Elija Crear el proyecto de compilación.
4. En Project configuration (Configuración del proyecto):

Nombre del proyecto

Introduzca un nombre para este proyecto de compilación. Los nombres de los proyectos de compilación deben ser únicos en cada AWS cuenta. También puede introducir una descripción opcional del proyecto de compilación para ayudar a otros usuarios a entender para qué se utiliza el proyecto.

5. En Source (Código fuente):

Proveedor de fuentes

Elige GitHub. Sigue las instrucciones para conectarte (o volver a conectarte) GitHub y, a continuación, selecciona Autorizar.

Repositorio

Selecciona Repositorio en mi GitHub cuenta.

GitHub repositorio

Introduce la URL de tu GitHub repositorio.

6. En Eventos de webhook de origen principal, seleccione lo siguiente.

Note

La sección de eventos de webhook de la fuente principal solo está visible si seleccionaste Repositorio en mi GitHub cuenta en el paso anterior.

1. Cuando cree el proyecto, seleccione Rebuild every time a code change is pushed to this repository (Volver a compilar cada vez que se inserte un cambio de código en este repositorio).
2. En Event type (Tipo de evento), seleccione uno o varios eventos.
3. Para filtrar en función de cuándo un evento va a desencadenar una compilación, en Start a build under these conditions (Iniciar una compilación en estas condiciones), añada uno o varios filtros opcionales.
4. Para filtrar en función de cuándo no se va a desencadenar un evento, en Don't start a build under these conditions (No iniciar una compilación en estas condiciones), añada uno o varios filtros opcionales.
5. Seleccione Añadir grupo de filtros para añadir otro grupo de filtros.

Para obtener más información sobre los filtros y tipos de eventos de GitHub webhook, consulte.

[GitHub eventos de webhook](#)

7. En Environment (Entorno):

Imagen del entorno

Seleccione una de las siguientes opciones:

Para usar una imagen de Docker gestionada por: AWS CodeBuild

Elija Imagen administrada y después elija opciones en Sistema operativo, Tiempo de ejecución, Imagen y Versión de imagen. Realice una selección en Environment type (Tipo de entorno) si está disponible.

Para utilizar otra imagen de Docker:

Elija Imagen personalizada. En Tipo de entorno, elija ARM, Linux, GPU de Linux o Windows. Si elige Otro registro, en URL de registro externo, introduzca el nombre

y etiqueta de la imagen de Docker en Docker Hub, usando el formato *docker repository/docker image name*. Si ha seleccionado Amazon ECR, use Repositorio de Amazon ECR e Imagen de Amazon ECR para elegir la imagen de Docker en su cuenta de AWS .

Para usar una imagen de Docker privada:

Elija Imagen personalizada. En Tipo de entorno, elija ARM, Linux, GPU de Linux o Windows. En Image registry (Registro de imagen), elija Other registry (Otro registro) y, a continuación, introduzca el ARN de las credenciales para su imagen de Docker privada. Es necesario crear las credenciales con Secrets Manager. Para obtener más información, consulte [¿Qué es? AWS Secrets Manager](#) en la Guía AWS Secrets Manager del usuario.

Rol de servicio

Seleccione una de las siguientes opciones:

- Si no tiene un rol de CodeBuild servicio, elija Nuevo rol de servicio. Escriba un nombre para el nuevo rol en el campo Role Name.
- Si tiene un rol CodeBuild de servicio, elija el rol de servicio existente. En ARN de rol, seleccione el rol de servicio.

Note

Cuando utilizas la consola para crear o actualizar un proyecto de compilación, puedes crear un rol de CodeBuild servicio al mismo tiempo. De forma predeterminada, el rol funciona solamente con ese proyecto de compilación. Si utiliza la consola para asociar este rol de servicio a otro proyecto de compilación, el rol se actualizará para que funcione con el otro proyecto de compilación. Un rol de servicio puede funcionar con hasta diez proyectos de compilación.

8. En Archivo de especificación de compilación, realice alguna de las operaciones siguientes:

- Seleccione Usar un archivo de especificaciones de compilación para utilizar el archivo `buyildspec.yml` que figura en el directorio raíz del código fuente.
- Elija Insertar comandos de compilación para usar la consola para insertar comandos de compilación.

Para obtener más información, consulte [Referencia de la especificación de compilación](#).

9. En Artifacts (Artefactos):

Tipo

Seleccione una de las siguientes opciones:

- Si no desea crear ningún artefacto de salida de la compilación, elija No artifacts (Sin artefactos).
- Para almacenar la salida de la compilación en un bucket de S3, seleccione Amazon S3 y, a continuación, haga lo siguiente:
 - Si desea utilizar su nombre de proyecto para el archivo ZIP o carpeta de salida de la compilación, deje en blanco Nombre. De lo contrario, escriba el nombre. De forma predeterminada, el nombre del artefacto es el nombre del proyecto. Si desea usar otro nombre, escríbalo en el cuadro de nombre de artefactos. Si desea producir un archivo ZIP, incluya la extensión zip.
 - En Bucket name (Nombre del bucket), seleccione el nombre del bucket de salida.
 - Si eligió Insert build commands (Insertar comandos de compilación) anteriormente en este procedimiento, en Output files (Archivos de salida), escriba las ubicaciones de los archivos de la compilación que desea incluir en el archivo ZIP o carpeta de salida de la compilación. Si hay varias ubicaciones, separe cada una de las ubicaciones con una coma (por ejemplo, `appspec.yml`, `target/my-app.jar`). Para obtener más información, consulte la descripción de files en [Sintaxis de buildspec](#).

Configuración adicional

Expanda Additional configuration (Configuración adicional) y establezca las opciones según sea necesario.

10. Elija Crear el proyecto de compilación. En la página Review (Revisar), elija Start build (Comenzar compilación) para ejecutar la compilación.

Comprobaciones de verificación

1. Abra la AWS CodeBuild consola en <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. En el panel de navegación, elija Proyectos de compilación.
3. Realice una de las siguientes acciones siguientes:
 - Elija el enlace del proyecto de compilación con webhooks que desea verificar y, a continuación, seleccione Build details (Detalles de compilación).

- Seleccione el botón de opción situado junto al proyecto de compilación con webhooks que desea verificar, elija Ver detalles y, a continuación, elija la pestaña Detalles de compilación.
4. En los eventos de webhook de la fuente principal, seleccione el enlace a la URL del Webhook.
 5. En tu GitHub repositorio, en la página de configuración, en Webhooks, comprueba que estén seleccionadas las solicitudes de extracción e inserción.
 6. En la configuración GitHub de tu perfil, en Configuración personal, Aplicaciones y Aplicaciones OAuth autorizadas, verás que tu aplicación está autorizada a acceder a la AWS región que has seleccionado.

Ejemplo para utilizar el control de versiones semánticas para asignar nombre a artefactos de compilación

El ejemplo contiene archivos buildspec que muestran cómo especificar un nombre de artefacto que se crea en el momento de la compilación. El nombre que se especifica en un archivo buildspec puede incorporar comandos de shell y variables de entorno para hacerlo único. Dicho nombre anulará otro nombre que escriba en la consola al crear el proyecto.

Si compila varias veces, el uso de un nombre de artefacto especificado en el archivo buildspec le ayuda a garantizar que los nombres de los archivos de artefactos de salida son únicos. Por ejemplo, puede utilizar una marca de fecha y hora para que se inserte en el nombre del artefacto en el momento de la compilación.

Si desea reemplazar el nombre de artefacto que ha escrito en la consola por un nombre del archivo buildspec, haga lo siguiente:

1. Establezca que el proyecto de compilación reemplace el nombre del artefacto por un nombre incluido en el archivo buildspec.
 - Si utiliza la consola para crear su proyecto de compilación, seleccione Enable semantic versioning (Habilitar control semántico de versiones). Para obtener más información, consulte [Creación de un proyecto de compilación \(consola\)](#).
 - Si utilizas la AWS CLI, establézcala en true en el archivo overrideArtifactName con formato JSON transferido a. create-project Para obtener más información, consulte [Crear un proyecto de compilación \(AWS CLI\)](#).

- Si utilizas la AWS CodeBuild API, establece la `overrideArtifactName` marca en el `ProjectArtifacts` objeto cuando se cree o actualice un proyecto o cuando se inicie una compilación.
2. Especifique un nombre en el archivo `buildspec`. Utilice los siguientes archivos `buildspec` de ejemplo como guía.

Este ejemplo de Linux muestra cómo especificar un nombre de artefacto que incluye la fecha de creación de la compilación:

```
version: 0.2
phases:
  build:
    commands:
      - rspec HelloWorld_spec.rb
artifacts:
  files:
    - '**/*'
  name: myname-${date +%Y-%m-%d}
```

En este ejemplo de Linux, se muestra cómo especificar el nombre de un artefacto que utilice una variable de CodeBuild entorno. Para obtener más información, consulte [Variables de entorno en los entornos de compilación](#).

```
version: 0.2
phases:
  build:
    commands:
      - rspec HelloWorld_spec.rb
artifacts:
  files:
    - '**/*'
  name: myname-${AWS_REGION}
```

Este ejemplo de Windows muestra cómo especificar un nombre de artefacto que incluye la fecha y la hora de creación de la compilación:

```
version: 0.2
env:
  variables:
```

```
TEST_ENV_VARIABLE: myArtifactName
phases:
  build:
    commands:
      - cd samples/helloworld
      - dotnet restore
      - dotnet run
artifacts:
  files:
    - '**/*'
name: $Env:TEST_ENV_VARIABLE-$(Get-Date -UFormat "%Y%m%d-%H%M%S")
```

En este ejemplo de Windows, se muestra cómo especificar un nombre de artefacto que utilice una variable declarada en el archivo buildspec y una variable de entorno. CodeBuild Para obtener más información, consulte [Variables de entorno en los entornos de compilación](#).

```
version: 0.2
env:
  variables:
    TEST_ENV_VARIABLE: myArtifactName
phases:
  build:
    commands:
      - cd samples/helloworld
      - dotnet restore
      - dotnet run
artifacts:
  files:
    - '**/*'
name: $Env:TEST_ENV_VARIABLE-$Env:AWS_REGION
```

Para obtener más información, consulte [Referencia de especificación de compilación para CodeBuild](#).

Ejemplos de Microsoft Windows para CodeBuild

Estos ejemplos utilizan un entorno de AWS CodeBuild compilación que ejecuta Microsoft Windows Server 2019, .NET Framework y .NET Core SDK para crear archivos de tiempo de ejecución a partir de código escrito en F# y Visual Basic.

⚠ Important

La ejecución de estos ejemplos podría generar cargos en su AWS cuenta. Estos incluyen posibles cargos por CodeBuild y por AWS los recursos y acciones relacionados con Amazon S3 y CloudWatch Logs. AWS KMS Para obtener más información, consulte [CodeBuildprecios](#), [precios de Amazon S3](#), [AWS Key Management Service precios](#) y [CloudWatch precios de Amazon](#).

Ejecución de los ejemplos

Para ejecutar estos ejemplos:

1. Cree los archivos tal y como se describe en las secciones «Estructura de directorios» y «Archivos» de este tema y, a continuación, cárguelos en un depósito de entrada de S3 o en un CodeCommit GitHub repositorio.

⚠ Important

No cargue (*root directory name*), solo los archivos incluidos en (*root directory name*).

Si utiliza un bucket de entrada de S3, no olvide crear un archivo ZIP que contenga los archivos y cárguelo en el bucket de entrada. No añada (*root directory name*) al archivo ZIP, solo los archivos incluidos en (*root directory name*).

2. Cree un proyecto de compilación. El proyecto de compilación debe usar la imagen `mcr.microsoft.com/dotnet/framework/sdk:4.8` para crear proyectos de .NET Framework.

Si utilizas el AWS CLI para crear el proyecto de compilación, la entrada del `create-project` comando con formato JSON podría tener un aspecto similar al siguiente. (Sustituya los marcadores de posición por sus propios valores).

```
{
  "name": "sample-windows-build-project",
  "source": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-input-bucket/windows-build-input-artifact.zip"
  }
}
```

```

},
"artifacts": {
  "type": "S3",
  "location": "codebuild-region-ID-account-ID-output-bucket",
  "packaging": "ZIP",
  "name": "windows-build-output-artifact.zip"
},
"environment": {
  "type": "WINDOWS_SERVER_2019_CONTAINER",
  "image": "mcr.microsoft.com/dotnet/framework/sdk:4.8",
  "computeType": "BUILD_GENERAL1_MEDIUM"
},
"serviceRole": "arn:aws:iam::account-ID:role/role-name",
"encryptionKey": "arn:aws:kms:region-ID:account-ID:key/key-ID"
}

```

3. Ejecute la compilación y siga los pasos que se indican en [Ejecutar una compilación](#).
4. Para obtener el artefacto de salida de la compilación, en el bucket de salida de S3, descargue el archivo *windows-build-output-artifact.zip* en el equipo o la instancia local. Extraiga el contenido para obtener los archivos de tiempo de ejecución y otros archivos.
 - El archivo de tiempo de ejecución del ejemplo de F# que utiliza .NET Framework, FSharpHelloWorld.exe, se encuentra en el directorio FSharpHelloWorld\bin\Debug.
 - El archivo de tiempo de ejecución del ejemplo de Visual Basic que utiliza .NET Framework, VBHelloWorld.exe, se encuentra en el directorio VBHelloWorld\bin\Debug.

Estructura de directorios

Estos ejemplos suponen que se utilizan las siguientes estructuras de directorio.

F # y .NET Framework

```

(root directory name)
### buildspec.yml
### FSharpHelloWorld.sln
### FSharpHelloWorld
### App.config
### AssemblyInfo.fs
### FSharpHelloWorld.fsproj
### Program.fs

```

Visual Basic y .NET Framework

```
(root directory name)
### buildspec.yml
### VBHelloWorld.sln
### VBHelloWorld
### App.config
### HelloWorld.vb
### VBHelloWorld.vbproj
### My Project
### Application.Designer.vb
### Application.myapp
### AssemblyInfo.vb
### Resources.Designer.vb
### Resources.resx
### Settings.Designer.vb
### Settings.settings
```

Archivos

En estos ejemplos, se utilizan los siguientes archivos:

F # y .NET Framework

buildspec.yml (in *(root directory name)*):

```
version: 0.2

env:
  variables:
    SOLUTION: .\FSharpHelloWorld.sln
    PACKAGE_DIRECTORY: .\packages
    DOTNET_FRAMEWORK: 4.8

phases:
  build:
    commands:
      - '& nuget restore $env:SOLUTION -PackagesDirectory $env:PACKAGE_DIRECTORY'
      - '& msbuild -p:FrameworkPathOverride="C:\Program Files (x86)\Reference
Assemblies\Microsoft\Framework\.NETFramework\v$env:DOTNET_FRAMEWORK" $env:SOLUTION'
artifacts:
  files:
```

```
- .\FSharpHelloWorld\bin\Debug\*
```

FSharpHelloWorld.sln (in *(root directory name)*):

```
Microsoft Visual Studio Solution File, Format Version 12.00
# Visual Studio 14
VisualStudioVersion = 14.0.25420.1
MinimumVisualStudioVersion = 10.0.40219.1
Project("{F2A71F9B-5D33-465A-A702-920D77279786}") = "FSharpHelloWorld",
  "FSharpHelloWorld\FSharpHelloWorld.fsproj", "{D60939B6-526D-43F4-9A89-577B2980DF62}"
EndProject
Global
  GlobalSection(SolutionConfigurationPlatforms) = preSolution
    Debug|Any CPU = Debug|Any CPU
    Release|Any CPU = Release|Any CPU
  EndGlobalSection
  GlobalSection(ProjectConfigurationPlatforms) = postSolution
    {D60939B6-526D-43F4-9A89-577B2980DF62}.Debug|Any CPU.ActiveCfg = Debug|Any CPU
    {D60939B6-526D-43F4-9A89-577B2980DF62}.Debug|Any CPU.Build.0 = Debug|Any CPU
    {D60939B6-526D-43F4-9A89-577B2980DF62}.Release|Any CPU.ActiveCfg = Release|Any CPU
    {D60939B6-526D-43F4-9A89-577B2980DF62}.Release|Any CPU.Build.0 = Release|Any CPU
  EndGlobalSection
  GlobalSection(SolutionProperties) = preSolution
    HideSolutionNode = FALSE
  EndGlobalSection
EndGlobal
```

App.config (in *(root directory name)*\FSharpHelloWorld):

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.8" />
  </startup>
</configuration>
```

AssemblyInfo.fs (in *(root directory name)*\FSharpHelloWorld):

```
namespace FSharpHelloWorld.AssemblyInfo

open System.Reflection
open System.Runtime.CompilerServices
```

```
open System.Runtime.InteropServices

// General Information about an assembly is controlled through the following
// set of attributes. Change these attribute values to modify the information
// associated with an assembly.
[<assembly: AssemblyTitle("FSharpHelloWorld")>]
[<assembly: AssemblyDescription("")>]
[<assembly: AssemblyConfiguration("")>]
[<assembly: AssemblyCompany("")>]
[<assembly: AssemblyProduct("FSharpHelloWorld")>]
[<assembly: AssemblyCopyright("Copyright © 2017")>]
[<assembly: AssemblyTrademark("")>]
[<assembly: AssemblyCulture("")>]

// Setting ComVisible to false makes the types in this assembly not visible
// to COM components.  If you need to access a type in this assembly from
// COM, set the ComVisible attribute to true on that type.
[<assembly: ComVisible(false)>]

// The following GUID is for the ID of the typelib if this project is exposed to COM
[<assembly: Guid("d60939b6-526d-43f4-9a89-577b2980df62")>]

// Version information for an assembly consists of the following four values:
//
// Major Version
// Minor Version
// Build Number
// Revision
//
// You can specify all the values or you can default the Build and Revision Numbers
// by using the '*' as shown below:
// [assembly: AssemblyVersion("1.0.*")]
[<assembly: AssemblyVersion("1.0.0.0")>]
[<assembly: AssemblyFileVersion("1.0.0.0")>]

do
    ()
```

FSharpHelloWorld.fsproj (in *(root directory name)*\FSharpHelloWorld):

```
<?xml version="1.0" encoding="utf-8"?>
<Project ToolsVersion="14.0" DefaultTargets="Build" xmlns="http://
schemas.microsoft.com/developer/msbuild/2003">
```

```

<Import Project="$(MSBuildExtensionsPath)\
$(MSBuildToolsVersion)\Microsoft.Common.props"
Condition="Exists('$(MSBuildExtensionsPath)\
$(MSBuildToolsVersion)\Microsoft.Common.props')" />
<PropertyGroup>
  <Configuration Condition=" '$(Configuration)' == '' ">Debug</Configuration>
  <Platform Condition=" '$(Platform)' == '' ">AnyCPU</Platform>
  <SchemaVersion>2.0</SchemaVersion>
  <ProjectGuid>d60939b6-526d-43f4-9a89-577b2980df62</ProjectGuid>
  <OutputType>Exe</OutputType>
  <RootNamespace>FSharpHelloWorld</RootNamespace>
  <AssemblyName>FSharpHelloWorld</AssemblyName>
  <TargetFrameworkVersion>v4.8</TargetFrameworkVersion>
  <AutoGenerateBindingRedirects>true</AutoGenerateBindingRedirects>
  <TargetFSharpCoreVersion>4.4.0.0</TargetFSharpCoreVersion>
  <Name>FSharpHelloWorld</Name>
</PropertyGroup>
<PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Debug|AnyCPU' ">
  <DebugSymbols>true</DebugSymbols>
  <DebugType>full</DebugType>
  <Optimize>>false</Optimize>
  <Tailcalls>>false</Tailcalls>
  <OutputPath>bin\Debug\</OutputPath>
  <DefineConstants>DEBUG;TRACE</DefineConstants>
  <WarningLevel>3</WarningLevel>
  <PlatformTarget>AnyCPU</PlatformTarget>
  <DocumentationFile>bin\Debug\FSharpHelloWorld.XML</DocumentationFile>
  <Prefer32Bit>true</Prefer32Bit>
</PropertyGroup>
<PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Release|AnyCPU' ">
  <DebugType>pdbonly</DebugType>
  <Optimize>true</Optimize>
  <Tailcalls>true</Tailcalls>
  <OutputPath>bin\Release\</OutputPath>
  <DefineConstants>TRACE</DefineConstants>
  <WarningLevel>3</WarningLevel>
  <PlatformTarget>AnyCPU</PlatformTarget>
  <DocumentationFile>bin\Release\FSharpHelloWorld.XML</DocumentationFile>
  <Prefer32Bit>true</Prefer32Bit>
</PropertyGroup>
<ItemGroup>
  <Reference Include="mscorlib" />
  <Reference Include="FSharp.Core, Version=$(TargetFSharpCoreVersion),
Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a">

```

```

    <Private>True</Private>
  </Reference>
  <Reference Include="System" />
  <Reference Include="System.Core" />
  <Reference Include="System.Numerics" />
</ItemGroup>
<ItemGroup>
  <Compile Include="AssemblyInfo.fs" />
  <Compile Include="Program.fs" />
  <None Include="App.config" />
</ItemGroup>
<PropertyGroup>
  <MinimumVisualStudioVersion Condition="'$(MinimumVisualStudioVersion)' == ''>11</
MinimumVisualStudioVersion>
</PropertyGroup>
<Choose>
  <When Condition="'$(VisualStudioVersion)' == '11.0'">
    <PropertyGroup Condition="Exists('$(MSBuildExtensionsPath32)\..\Microsoft SDKs\F#
\3.0\Framework\v4.0\Microsoft.FSharp.Targets')">
      <FSharpTargetsPath>$(MSBuildExtensionsPath32)\..\Microsoft SDKs\F#
\3.0\Framework\v4.0\Microsoft.FSharp.Targets</FSharpTargetsPath>
    </PropertyGroup>
  </When>
  <Otherwise>
    <PropertyGroup Condition="Exists('$(MSBuildExtensionsPath32)\Microsoft
\VisualStudio\v$(VisualStudioVersion)\FSharp\Microsoft.FSharp.Targets')">
      <FSharpTargetsPath>$(MSBuildExtensionsPath32)\Microsoft\VisualStudio\v
$(VisualStudioVersion)\FSharp\Microsoft.FSharp.Targets</FSharpTargetsPath>
    </PropertyGroup>
  </Otherwise>
</Choose>
<Import Project="$(FSharpTargetsPath)" />
<!-- To modify your build process, add your task inside one of the targets below and
uncomment it.
    Other similar extension points exist, see Microsoft.Common.targets.
  <Target Name="BeforeBuild">
  </Target>
  <Target Name="AfterBuild">
  </Target>
  -->
</Project>

```

Program.fs (in *(root directory name)*\FSharpHelloWorld):

```
// Learn more about F# at http://fsharp.org
// See the 'F# Tutorial' project for more help.

[<EntryPoint>]
let main argv =
    printfn "Hello World"
    0 // return an integer exit code
```

Visual Basic y .NET Framework

buildspec.yml (in *(root directory name)*):

```
version: 0.2

env:
  variables:
    SOLUTION: .\VBHelloWorld.sln
    PACKAGE_DIRECTORY: .\packages
    DOTNET_FRAMEWORK: 4.8

phases:
  build:
    commands:
      - '& "C:\ProgramData\chocolatey\bin\NuGet.exe" restore $env:SOLUTION -
        PackagesDirectory $env:PACKAGE_DIRECTORY'
      - '& "C:\Program Files (x86)\MSBuild\14.0\Bin\MSBuild.exe" -
        p:FrameworkPathOverride="C:\Program Files (x86)\Reference Assemblies\Microsoft
        \Framework\.NETFramework\v$env:DOTNET_FRAMEWORK" $env:SOLUTION'
    artifacts:
      files:
        - .\VBHelloWorld\bin\Debug\*
```

VBHelloWorld.sln (in *(root directory name)*):

```
Microsoft Visual Studio Solution File, Format Version 12.00
# Visual Studio 14
VisualStudioVersion = 14.0.25420.1
MinimumVisualStudioVersion = 10.0.40219.1
Project("{F184B08F-C81C-45F6-A57F-5ABD9991F28F}") = "VBHelloWorld", "VBHelloWorld
\VBHelloWorld.vbproj", "{4DCEC446-7156-4FE6-8CCC-219E34DD409D}"
EndProject
Global
```

```

GlobalSection(SolutionConfigurationPlatforms) = preSolution
    Debug|Any CPU = Debug|Any CPU
    Release|Any CPU = Release|Any CPU
EndGlobalSection
GlobalSection(ProjectConfigurationPlatforms) = postSolution
    {4DCEC446-7156-4FE6-8CCC-219E34DD409D}.Debug|Any CPU.ActiveCfg = Debug|Any CPU
    {4DCEC446-7156-4FE6-8CCC-219E34DD409D}.Debug|Any CPU.Build.0 = Debug|Any CPU
    {4DCEC446-7156-4FE6-8CCC-219E34DD409D}.Release|Any CPU.ActiveCfg = Release|Any CPU
    {4DCEC446-7156-4FE6-8CCC-219E34DD409D}.Release|Any CPU.Build.0 = Release|Any CPU
EndGlobalSection
GlobalSection(SolutionProperties) = preSolution
    HideSolutionNode = FALSE
EndGlobalSection
EndGlobal

```

App.config (in *(root directory name)*\VBHelloWorld):

```

<?xml version="1.0" encoding="utf-8" ?>
<configuration>
    <startup>
        <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.8" />
    </startup>
</configuration>

```

HelloWorld.vb (in *(root directory name)*\VBHelloWorld):

```

Module HelloWorld

    Sub Main()
        MsgBox("Hello World")
    End Sub

End Module

```

VBHelloWorld.vbproj (in *(root directory name)*\VBHelloWorld):

```

<?xml version="1.0" encoding="utf-8"?>
<Project ToolsVersion="14.0" DefaultTargets="Build" xmlns="http://
schemas.microsoft.com/developer/msbuild/2003">
    <Import Project="$(MSBuildExtensionsPath)\
$(MSBuildToolsVersion)\Microsoft.Common.props"

```

```

Condition="Exists('$$(MSBuildExtensionsPath)\
$(MSBuildToolsVersion)\Microsoft.Common.props')" />
  <PropertyGroup>
    <Configuration Condition=" '$(Configuration)' == '' ">Debug</Configuration>
    <Platform Condition=" '$(Platform)' == '' ">AnyCPU</Platform>
    <ProjectGuid>{4DCEC446-7156-4FE6-8CCC-219E34DD409D}</ProjectGuid>
    <OutputType>Exe</OutputType>
    <StartupObject>VBHelloWorld.HelloWorld</StartupObject>
    <RootNamespace>VBHelloWorld</RootNamespace>
    <AssemblyName>VBHelloWorld</AssemblyName>
    <FileAlignment>512</FileAlignment>
    <MyType>Console</MyType>
    <TargetFrameworkVersion>v4.8</TargetFrameworkVersion>
    <AutoGenerateBindingRedirects>true</AutoGenerateBindingRedirects>
  </PropertyGroup>
  <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Debug|AnyCPU' ">
    <PlatformTarget>AnyCPU</PlatformTarget>
    <DebugSymbols>true</DebugSymbols>
    <DebugType>full</DebugType>
    <DefineDebug>true</DefineDebug>
    <DefineTrace>true</DefineTrace>
    <OutputPath>bin\Debug\</OutputPath>
    <DocumentationFile>VBHelloWorld.xml</DocumentationFile>
    <NoWarn>42016,41999,42017,42018,42019,42032,42036,42020,42021,42022</NoWarn>
  </PropertyGroup>
  <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Release|AnyCPU' ">
    <PlatformTarget>AnyCPU</PlatformTarget>
    <DebugType>pdbonly</DebugType>
    <DefineDebug>>false</DefineDebug>
    <DefineTrace>true</DefineTrace>
    <Optimize>true</Optimize>
    <OutputPath>bin\Release\</OutputPath>
    <DocumentationFile>VBHelloWorld.xml</DocumentationFile>
    <NoWarn>42016,41999,42017,42018,42019,42032,42036,42020,42021,42022</NoWarn>
  </PropertyGroup>
  <PropertyGroup>
    <OptionExplicit>On</OptionExplicit>
  </PropertyGroup>
  <PropertyGroup>
    <OptionCompare>Binary</OptionCompare>
  </PropertyGroup>
  <PropertyGroup>
    <OptionStrict>Off</OptionStrict>
  </PropertyGroup>

```

```
<PropertyGroup>
  <OptionInfer>On</OptionInfer>
</PropertyGroup>
<ItemGroup>
  <Reference Include="System" />
  <Reference Include="System.Data" />
  <Reference Include="System.Deployment" />
  <Reference Include="System.Xml" />
  <Reference Include="System.Core" />
  <Reference Include="System.Xml.Linq" />
  <Reference Include="System.Data.DataSetExtensions" />
  <Reference Include="System.Net.Http" />
</ItemGroup>
<ItemGroup>
  <Import Include="Microsoft.VisualBasic" />
  <Import Include="System" />
  <Import Include="System.Collections" />
  <Import Include="System.Collections.Generic" />
  <Import Include="System.Data" />
  <Import Include="System.Diagnostics" />
  <Import Include="System.Linq" />
  <Import Include="System.Xml.Linq" />
  <Import Include="System.Threading.Tasks" />
</ItemGroup>
<ItemGroup>
  <Compile Include="HelloWorld.vb" />
  <Compile Include="My Project\AssemblyInfo.vb" />
  <Compile Include="My Project\Application.Designer.vb">
    <AutoGen>True</AutoGen>
    <DependentUpon>Application.myapp</DependentUpon>
  </Compile>
  <Compile Include="My Project\Resources.Designer.vb">
    <AutoGen>True</AutoGen>
    <DesignTime>True</DesignTime>
    <DependentUpon>Resources.resx</DependentUpon>
  </Compile>
  <Compile Include="My Project\Settings.Designer.vb">
    <AutoGen>True</AutoGen>
    <DependentUpon>Settings.settings</DependentUpon>
    <DesignTimeSharedInput>True</DesignTimeSharedInput>
  </Compile>
</ItemGroup>
<ItemGroup>
  <EmbeddedResource Include="My Project\Resources.resx">
```

```

    <Generator>VbMyResourcesResXFileCodeGenerator</Generator>
    <LastGenOutput>Resources.Designer.vb</LastGenOutput>
    <CustomToolNamespace>My.Resources</CustomToolNamespace>
    <SubType>Designer</SubType>
  </EmbeddedResource>
</ItemGroup>
<ItemGroup>
  <None Include="My Project\Application.myapp">
    <Generator>MyApplicationCodeGenerator</Generator>
    <LastGenOutput>Application.Designer.vb</LastGenOutput>
  </None>
  <None Include="My Project\Settings.settings">
    <Generator>SettingsSingleFileGenerator</Generator>
    <CustomToolNamespace>My</CustomToolNamespace>
    <LastGenOutput>Settings.Designer.vb</LastGenOutput>
  </None>
  <None Include="App.config" />
</ItemGroup>
<Import Project="$(MSBuildToolsPath)\Microsoft.VisualBasic.targets" />
<!-- To modify your build process, add your task inside one of the targets below and
uncomment it.
    Other similar extension points exist, see Microsoft.Common.targets.
<Target Name="BeforeBuild">
</Target>
<Target Name="AfterBuild">
</Target>
-->
</Project>

```

Application.Designer.vb (in *(root directory name)\VBHelloWorld\My Project*):

```

'-----
' <auto-generated>
'   This code was generated by a tool.
'   Runtime Version:4.0.30319.42000
'
'   Changes to this file may cause incorrect behavior and will be lost if
'   the code is regenerated.
' </auto-generated>
'-----

Option Strict On
Option Explicit On

```

Application.myapp (in *(root directory name)*\VBHelloWorld\My Project):

```
<?xml version="1.0" encoding="utf-8"?>
<MyApplicationData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <MySubMain>false</MySubMain>
  <SingleInstance>false</SingleInstance>
  <ShutdownMode>0</ShutdownMode>
  <EnableVisualStyles>true</EnableVisualStyles>
  <AuthenticationMode>0</AuthenticationMode>
  <ApplicationType>2</ApplicationType>
  <SaveMySettingsOnExit>true</SaveMySettingsOnExit>
</MyApplicationData>
```

AssemblyInfo.vb (in *(root directory name)*\VBHelloWorld\My Project):

```
Imports System
Imports System.Reflection
Imports System.Runtime.InteropServices

' General Information about an assembly is controlled through the following
' set of attributes. Change these attribute values to modify the information
' associated with an assembly.

' Review the values of the assembly attributes

<Assembly: AssemblyTitle("VBHelloWorld")>
<Assembly: AssemblyDescription("")>
<Assembly: AssemblyCompany("")>
<Assembly: AssemblyProduct("VBHelloWorld")>
<Assembly: AssemblyCopyright("Copyright © 2017")>
<Assembly: AssemblyTrademark("")>

<Assembly: ComVisible(False)>

'The following GUID is for the ID of the typelib if this project is exposed to COM
<Assembly: Guid("137c362b-36ef-4c3e-84ab-f95082487a5a")>

' Version information for an assembly consists of the following four values:
'
' Major Version
' Minor Version
' Build Number
```

```
' Revision
'
' You can specify all the values or you can default the Build and Revision Numbers
' by using the '*' as shown below:
' <Assembly: AssemblyVersion("1.0.*")>

<Assembly: AssemblyVersion("1.0.0.0")>
<Assembly: AssemblyFileVersion("1.0.0.0")>
```

Resources.Designer.vb (in *(root directory name)*\VBHelloWorld\My Project):

```
' -----
' <auto-generated>
'   This code was generated by a tool.
'   Runtime Version:4.0.30319.42000
'
'   Changes to this file may cause incorrect behavior and will be lost if
'   the code is regenerated.
' </auto-generated>
' -----

Option Strict On
Option Explicit On

Namespace My.Resources

    'This class was auto-generated by the StronglyTypedResourceBuilder
    'class via a tool like ResGen or Visual Studio.
    'To add or remove a member, edit your .ResX file then rerun ResGen
    'with the /str option, or rebuild your VS project.
    '''<summary>
    ''' A strongly-typed resource class, for looking up localized strings, etc.
    '''</summary>

    <Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Resources.Tools.StronglyTypedRe
    "4.0.0.0"), _
    Global.System.Diagnostics.DebuggerNonUserCodeAttribute(), _
    Global.System.Runtime.CompilerServices.CompilerGeneratedAttribute(), _
    Global.Microsoft.VisualBasic.HideModuleNameAttribute(> _
    Friend Module Resources

        Private resourceMan As Global.System.Resources.ResourceManager
```

```

Private resourceCulture As Global.System.Globalization.CultureInfo

'''<summary>
''' Returns the cached ResourceManager instance used by this class.
'''</summary>

<Global.System.ComponentModel.EditorBrowsableAttribute(Global.System.ComponentModel.EditorBrow
-
Friend ReadOnly Property ResourceManager() As
Global.System.Resources.ResourceManager
    Get
        If Object.ReferenceEquals(resourceMan, Nothing) Then
            Dim temp As Global.System.Resources.ResourceManager = New
Global.System.Resources.ResourceManager("VBHelloWorld.Resources",
GetType(Resources).Assembly)
            resourceMan = temp
        End If
        Return resourceMan
    End Get
End Property

'''<summary>
''' Overrides the current thread's CurrentUICulture property for all
''' resource lookups using this strongly typed resource class.
'''</summary>

<Global.System.ComponentModel.EditorBrowsableAttribute(Global.System.ComponentModel.EditorBrow
-
Friend Property Culture() As Global.System.Globalization.CultureInfo
    Get
        Return resourceCulture
    End Get
    Set(ByVal value As Global.System.Globalization.CultureInfo)
        resourceCulture = value
    End Set
End Property
End Module
End Namespace

```

Resources.resx (in *(root directory name)*\VBHelloWorld\My Project):

```

<?xml version="1.0" encoding="utf-8"?>
<root>

```

```
<!--
```

```
Microsoft ResX Schema
```

```
Version 2.0
```

The primary goals of this format is to allow a simple XML format that is mostly human readable. The generation and parsing of the various data types are done through the `TypeConverter` classes associated with the data types.

Example:

```
... ado.net/XML headers & schema ...
<resheader name="resmimetype">text/microsoft-resx</resheader>
<resheader name="version">2.0</resheader>
<resheader name="reader">System.Resources.ResXResourceReader,
System.Windows.Forms, ...</resheader>
<resheader name="writer">System.Resources.ResXResourceWriter,
System.Windows.Forms, ...</resheader>
<data name="Name1"><value>this is my long string</value><comment>this is a
comment</comment></data>
<data name="Color1" type="System.Drawing.Color, System.Drawing">Blue</data>
<data name="Bitmap1" mimetype="application/x-microsoft.net.object.binary.base64">
  <value>[base64 mime encoded serialized .NET Framework object]</value>
</data>
<data name="Icon1" type="System.Drawing.Icon, System.Drawing"
mimetype="application/x-microsoft.net.object.bytearray.base64">
  <value>[base64 mime encoded string representing a byte array form of the .NET
Framework object]</value>
  <comment>This is a comment</comment>
</data>
```

There are any number of "resheader" rows that contain simple name/value pairs.

Each data row contains a name, and value. The row also contains a type or mimetype. Type corresponds to a .NET class that support text/value conversion through the `TypeConverter` architecture. Classes that don't support this are serialized and stored with the mimetype set.

The mimetype is used for serialized objects, and tells the `ResXResourceReader` how to depersist the object. This is currently not extensible. For a given mimetype the value must be set accordingly:

Note - `application/x-microsoft.net.object.binary.base64` is the format that the `ResXResourceWriter` will generate, however the reader can read any of the formats listed below.

```
mimetype: application/x-microsoft.net.object.binary.base64
value    : The object must be serialized with
          : System.Serialization.Formatter.Binary.BinaryFormatter
          : and then encoded with base64 encoding.
```

```
mimetype: application/x-microsoft.net.object.soap.base64
value    : The object must be serialized with
          : System.Runtime.Serialization.Formatter.Soap.SoapFormatter
          : and then encoded with base64 encoding.
```

```
mimetype: application/x-microsoft.net.object.bytearray.base64
value    : The object must be serialized into a byte array
          : using a System.ComponentModel.TypeConverter
          : and then encoded with base64 encoding.
```

```
-->
```

```
<xsd:schema id="root" xmlns="" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
  <xsd:element name="root" msdata:IsDataSet="true">
    <xsd:complexType>
      <xsd:choice maxOccurs="unbounded">
        <xsd:element name="metadata">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="value" type="xsd:string" minOccurs="0" />
            </xsd:sequence>
            <xsd:attribute name="name" type="xsd:string" />
            <xsd:attribute name="type" type="xsd:string" />
            <xsd:attribute name="mimetype" type="xsd:string" />
          </xsd:complexType>
        </xsd:element>
        <xsd:element name="assembly">
          <xsd:complexType>
            <xsd:attribute name="alias" type="xsd:string" />
            <xsd:attribute name="name" type="xsd:string" />
          </xsd:complexType>
        </xsd:element>
        <xsd:element name="data">
          <xsd:complexType>
            <xsd:sequence>
```

```

        <xsd:element name="value" type="xsd:string" minOccurs="0"
msdata:Ordinal="1" />
        <xsd:element name="comment" type="xsd:string" minOccurs="0"
msdata:Ordinal="2" />
    </xsd:sequence>
    <xsd:attribute name="name" type="xsd:string" msdata:Ordinal="1" />
    <xsd:attribute name="type" type="xsd:string" msdata:Ordinal="3" />
    <xsd:attribute name="mimetype" type="xsd:string" msdata:Ordinal="4" />
</xsd:complexType>
</xsd:element>
<xsd:element name="resheader">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="value" type="xsd:string" minOccurs="0"
msdata:Ordinal="1" />
        </xsd:sequence>
        <xsd:attribute name="name" type="xsd:string" use="required" />
    </xsd:complexType>
</xsd:element>
</xsd:choice>
</xsd:complexType>
</xsd:element>
</xsd:schema>
<resheader name="resmimetype">
    <value>text/microsoft-resx</value>
</resheader>
<resheader name="version">
    <value>2.0</value>
</resheader>
<resheader name="reader">
    <value>System.Resources.ResXResourceReader, System.Windows.Forms, Version=2.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089</value>
</resheader>
<resheader name="writer">
    <value>System.Resources.ResXResourceWriter, System.Windows.Forms, Version=2.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089</value>
</resheader>
</root>

```

Settings.Designer.vb (in *(root directory name)*\VBHelloWorld\My Project):

```

'-----
' <auto-generated>

```

```
' This code was generated by a tool.
' Runtime Version:4.0.30319.42000
'
' Changes to this file may cause incorrect behavior and will be lost if
' the code is regenerated.
' </auto-generated>
'-----

Option Strict On
Option Explicit On

Namespace My

    <Global.System.Runtime.CompilerServices.CompilerGeneratedAttribute(), _
Global.System.CodeDom.Compiler.GeneratedCodeAttribute("Microsoft.VisualStudio.Editors.Settings
"11.0.0.0"), _
Global.System.ComponentModel.EditorBrowsableAttribute(Global.System.ComponentModel.EditorBrowsable
_
    Partial Friend NotInheritable Class MySettings
        Inherits Global.System.Configuration.ApplicationSettingsBase

        Private Shared defaultInstance As MySettings =
CType(Global.System.Configuration.ApplicationSettingsBase.Synchronized(New
MySettings), MySettings)

        #Region "My.Settings Auto-Save Functionality"
            #If _MyType = "WindowsForms" Then
                Private Shared addedHandler As Boolean

                Private Shared addedHandlerLockObject As New Object

                <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),
Global.System.ComponentModel.EditorBrowsableAttribute(Global.System.ComponentModel.EditorBrowsable
_
            Private Shared Sub AutoSaveSettings(ByVal sender As Global.System.Object, ByVal
e As Global.System.EventArgs)
                If My.Application.SaveMySettingsOnExit Then
                    My.Settings.Save()
                End If
            End Sub
        #End If
    #End Region
```

```

Public Shared ReadOnly Property [Default]() As MySettings
    Get

        #If _MyType = "WindowsForms" Then
            If Not addedHandler Then
                SyncLock addedHandlerLockObject
                    If Not addedHandler Then
                        AddHandler My.Application.Shutdown, AddressOf AutoSaveSettings
                        addedHandler = True
                    End If
                End SyncLock
            End If
        #End If
        Return defaultInstance
    End Get
End Property
End Class
End Namespace

Namespace My

    <Global.Microsoft.VisualBasic.HideModuleNameAttribute(), _
    Global.System.Diagnostics.DebuggerNonUserCodeAttribute(), _
    Global.System.Runtime.CompilerServices.CompilerGeneratedAttribute()> _
    Friend Module MySettingsProperty

        <Global.System.ComponentModel.Design.HelpKeywordAttribute("My.Settings")> _
        Friend ReadOnly Property Settings() As Global.VBHelloWorld.My.MySettings
            Get
                Return Global.VBHelloWorld.My.MySettings.Default
            End Get
        End Property
    End Module
End Namespace

```

Settings.settings (in *(root directory name)*\VBHelloWorld\My Project):

```

<?xml version='1.0' encoding='utf-8'?>
<SettingsFile xmlns="http://schemas.microsoft.com/VisualStudio/2004/01/settings"
CurrentProfile="(Default)" UseMySettingsClassName="true">
  <Profiles>
    <Profile Name="(Default)" />

```

```
</Profiles>  
<Settings />  
</SettingsFile>
```

Planificar una compilación en AWS CodeBuild

Antes de usar AWS CodeBuild, debe responder estas preguntas:

1. ¿Dónde se guarda el código fuente? CodeBuild actualmente admite la creación a partir de los siguientes proveedores de repositorios de código fuente. El código fuente debe contener un archivo de especificación de compilación (buildspec). Una especificación de compilación es un conjunto de comandos de compilación y configuraciones relacionadas, en formato YAML, que se CodeBuild utiliza para ejecutar una compilación. Es posible declarar un archivo de especificación de compilación en una definición del proyecto de compilación.

Proveedor de repositorio	Obligatoria	Documentación
CodeCommit	<p>Nombre del repositorio.</p> <p>(Opcional) ID de confirmación asociado con el código fuente.</p>	<p>Consulte estos temas en la Guía del usuario de AWS CodeCommit:</p> <p>Crea un repositorio CodeCommit</p> <p>Crea una confirmación en CodeCommit</p>
Amazon S3	<p>Nombre del bucket de entrada.</p> <p>Nombre de objeto correspondiente al archivo ZIP de entrada de la compilación que contiene el código fuente.</p> <p>(Opcional) ID de versión asociado al</p>	<p>Consulte estos temas en la Guía de introducción de Amazon S3:</p> <p>Crear un bucket</p> <p>Añadir un objeto a un bucket.</p>

Proveedor de repositorio	Obligatoria	Documentación
	archivo ZIP de entrada de la compilación.	
GitHub	Nombre del repositorio. (Opcional) ID de confirmación asociado con el código fuente.	Consulte este tema en el sitio web de GitHub ayuda: Crear un repositorio
Bitbucket	Nombre del repositorio. (Opcional) ID de confirmación asociado con el código fuente.	Consulte este tema en el sitio web de documentación de Bitbucket Cloud: Creación de un repositorio

- ¿Qué comandos de compilación necesita ejecutar y en qué orden? De forma predeterminada, CodeBuild descarga la entrada de compilación del proveedor que especifique y carga la salida de la compilación en el depósito que especifique. Debe usar la especificación de compilación para indicar cómo convertir la entrada de compilación descargada en la salida de compilación prevista. Para obtener más información, consulte [Referencia de la especificación de compilación](#).
- ¿Qué tiempos de ejecución y herramientas necesita para ejecutar la compilación? Por ejemplo, ¿va a compilar para Java, Ruby, Python o Node.js? ¿Necesita la compilación Maven o Ant, o un compilador para Java, Ruby o Python? ¿Necesita la compilación Git, la AWS CLI u otras herramientas?

CodeBuild ejecuta compilaciones en entornos de compilación que usan imágenes de Docker. Estas imágenes de Docker deberán almacenarse en un tipo de repositorio compatible con CodeBuild. Estos incluyen el repositorio de imágenes de CodeBuild Docker, Docker Hub y Amazon Elastic Container Registry (Amazon ECR). Para obtener más información sobre el repositorio de imágenes de CodeBuild Docker, consulte [Imágenes de Docker proporcionadas por CodeBuild](#)

4. ¿Necesita AWS recursos que no le proporcione automáticamente? CodeBuild En tal caso, ¿qué políticas de seguridad necesitan estos recursos? Por ejemplo, puede que tengas que modificar la función de CodeBuild servicio CodeBuild para poder trabajar con esos recursos.
5. ¿Quieres CodeBuild trabajar con tu VPC? En tal caso, necesita el ID de la VPC, los identificadores de subred y los identificadores de grupo de seguridad de la configuración de su VPC. Para obtener más información, consulte [Úselo AWS CodeBuild con Amazon Virtual Private Cloud](#).

Después de responder a estas preguntas, debe disponer de la configuración y los recursos que necesita para ejecutar la compilación correctamente. Para ejecutar la compilación, puede:

- Usar la consola de AWS CodeBuild, AWS CLI o los SDK de AWS. Para obtener más información, consulte [Ejecutar una compilación](#).
- Cree o identifique una canalización yAWS CodePipeline, a continuación, añada una acción de compilación o prueba que indique que CodeBuild debe probar automáticamente el código, ejecutar la compilación o ambas opciones. Para obtener más información, consulte [Úselo CodePipeline con CodeBuild](#).

Referencia de especificación de compilación para CodeBuild

En este tema, se proporciona información de referencia importante sobre los archivos de especificación de compilación (buildspec). Una especificación de compilación es un conjunto de comandos de compilación y configuraciones relacionadas, en formato YAML, que se CodeBuild utiliza para ejecutar una compilación. Puede incluir una especificación de compilación como parte del código fuente o puede incluir una especificación de compilación cuando cree un proyecto de compilación. Para obtener información sobre cómo funciona una especificación de compilación, consulte [Cómo funciona CodeBuild](#).

Temas

- [Nombre de archivo y ubicación de almacenamiento de buildspec](#)
- [Sintaxis de buildspec](#)
- [Ejemplo de un archivo buildspec](#)
- [Versiones de buildspec](#)
- [Referencia de especificaciones de compilación para compilación por lotes](#)

Nombre de archivo y ubicación de almacenamiento de buildspec

Si incluye una especificación de compilación como parte del código fuente, de forma predeterminada, el archivo de especificación de compilación debe llamarse `buildspec.yml` y debe encontrarse en la raíz del directorio de código fuente.

Puede invalidar el nombre y la ubicación predeterminados del archivo de especificación de compilación. Por ejemplo, puede hacer lo siguiente:

- Usar un archivo de especificación de compilación diferente para las distintas compilaciones del mismo repositorio, como `buildspec_debug.yml` y `buildspec_release.yml`.
- Almacenar un archivo de especificación de compilación en otro lugar que no sea la raíz de su directorio de origen, como en `config/buildspec.yml` o en un bucket de S3. El bucket de S3 debe estar en la misma AWS región que tu proyecto de compilación. Especifique el archivo `buildspec` utilizando su ARN (por ejemplo, `arn:aws:s3:::<my-codebuild-sample2>/buildspec.yml`).

Solo puede especificar una especificación de compilación para un proyecto de compilación, independientemente del nombre del archivo de especificación de compilación.

Para invalidar el nombre y/o la ubicación del archivo de especificación de compilación, realice alguna de las siguientes operaciones:

- Ejecuta el `update-project` comando AWS CLI `create-project` o y establece el `buildspec` valor de la ruta al archivo `buildspec` alternativo en relación con el valor de la variable de entorno integrada. `CODEBUILD_SRC_DIR` También puedes hacer lo mismo con la `create project` operación en los SDK. AWS Para obtener más información, consulte [Creación de un proyecto de compilación](#) o [Cambiar la configuración de un proyecto de compilación](#).
- Ejecute el AWS CLI `start-build` comando y establezca el `buildspecOverride` valor de la ruta al archivo `buildspec` alternativo en relación con el valor de la variable de entorno integrada. `CODEBUILD_SRC_DIR` También puedes hacer lo mismo con la `start build` operación en los SDK. AWS Para obtener más información, consulte [Ejecutar una compilación](#).
- En una AWS CloudFormation plantilla, establece la `BuildSpec` propiedad de `Source` en un recurso de tipo en la ruta `AWS::CodeBuild::Project` al archivo `buildspec` alternativo en relación con el valor de la variable de entorno integrada. `CODEBUILD_SRC_DIR` Para obtener más información, consulte la `BuildSpec` propiedad en la [fuente del AWS CodeBuild proyecto](#) en la Guía del AWS CloudFormation usuario.

Sintaxis de buildspec

Los archivos de especificación de compilación deben expresarse en formato [YAML](#).

Si un comando contiene un carácter, o una cadena de caracteres, que no es compatible con YAML, debe encerrar el comando entre comillas (""). El siguiente comando se incluye entre comillas porque no se permiten dos puntos (:) seguidos de un espacio en YAML. La comilla en el comando utiliza la secuencia de escape (\").

```
"export PACKAGE_NAME=$(cat package.json | grep name | head -1 | awk -F: '{ print $2 }' | sed 's/[\\\",]//g')"
```

La especificación de compilación tiene la siguiente sintaxis:

```
version: 0.2

run-as: Linux-user-name

env:
  shell: shell-tag
  variables:
    key: "value"
    key: "value"
  parameter-store:
    key: "value"
    key: "value"
  exported-variables:
    - variable
    - variable
  secrets-manager:
    key: secret-id:json-key:version-stage:version-id
  git-credential-helper: no | yes

proxy:
  upload-artifacts: no | yes
  logs: no | yes

batch:
  fast-fail: false | true
  # build-list:
  # build-matrix:
  # build-graph:
```

```
phases:
  install:
    run-as: Linux-user-name
    on-failure: ABORT | CONTINUE
    runtime-versions:
      runtime: version
      runtime: version
    commands:
      - command
      - command
    finally:
      - command
      - command
    # steps:
  pre_build:
    run-as: Linux-user-name
    on-failure: ABORT | CONTINUE
    commands:
      - command
      - command
    finally:
      - command
      - command
    # steps:
  build:
    run-as: Linux-user-name
    on-failure: ABORT | CONTINUE
    commands:
      - command
      - command
    finally:
      - command
      - command
    # steps:
  post_build:
    run-as: Linux-user-name
    on-failure: ABORT | CONTINUE
    commands:
      - command
      - command
    finally:
      - command
      - command
```

```
# steps:
reports:
  report-group-name-or-arn:
    files:
      - location
      - location
    base-directory: location
    discard-paths: no | yes
    file-format: report-format
artifacts:
  files:
    - location
    - location
  name: artifact-name
  discard-paths: no | yes
  base-directory: location
  exclude-paths: excluded paths
  enable-symlinks: no | yes
  s3-prefix: prefix
  secondary-artifacts:
    artifactIdentifier:
      files:
        - location
        - location
      name: secondary-artifact-name
      discard-paths: no | yes
      base-directory: location
    artifactIdentifier:
      files:
        - location
        - location
      discard-paths: no | yes
      base-directory: location
cache:
  paths:
    - path
    - path
```

La especificación de compilación contiene lo siguiente:

versión

Mapeo obligatorio. Representa la versión de la especificación de compilación. Le recomendamos que utilice 0.2.

Note

Aunque la versión 0.1 sigue siendo compatible, le recomendamos que utilice la versión 0.2 siempre que sea posible. Para obtener más información, consulte [Versiones de buildspec](#).

run-as

Secuencia opcional. Disponible solo para usuarios de Linux. Especifica un usuario de Linux que ejecuta comandos en este archivo de especificación de compilación. `run-as` concede al usuario especificado permisos de lectura y ejecución. Cuando se especifica `run-as` en la parte superior del archivo `buildspec`, se aplica globalmente a todos los comandos. Si no desea especificar un usuario para todos los comandos de archivo `buildspec`, puede especificar uno para comandos en una fase utilizando `run-as` en uno de los bloques `phases`. Si no se especifica `run-as`, todos los comandos se ejecutan como usuario raíz.

env

Secuencia opcional. Representa información para una o más variables de entorno personalizadas.

Note

Para proteger la información confidencial, los CodeBuild registros ocultan lo siguiente:

- AWS identificadores de claves de acceso. Para obtener más información, consulte [Administración de claves de acceso para usuarios de IAM](#) en la Guía del usuario de AWS Identity and Access Management .
- Cadenas especificadas mediante el almacén de parámetros. Para obtener más información, consulte [Almacén de parámetros de Systems Manager](#) y [Tutorial de la consola del almacén de parámetros de Systems Manager](#) en la Guía del usuario de Amazon EC2 Systems Manager.
- Cadenas especificadas mediante AWS Secrets Manager. Para obtener más información, consulte [Administración de claves](#).

env/shell

Secuencia opcional. Especifica el intérprete de comandos compatible con los sistemas operativos Linux o Windows.

En los sistemas operativos Linux, las etiquetas de intérprete de comandos compatibles son:

- `bash`
- `/bin/sh`

En los sistemas operativos Windows, las etiquetas de intérprete de comandos compatibles son:

- `powershell.exe`
- `cmd.exe`

env/variables

Obligatorio si se especifica `env` y desea definir variables de entorno personalizadas en texto sin formato. Contiene una asignación de pares *clave/valor* escalares, donde cada asignación representa una única variable de entorno personalizada en texto sin formato. La *clave* es el nombre de la variable de entorno personalizada, mientras que el *valor* es el valor de la variable.

Important

Se desaconseja encarecidamente almacenar valores confidenciales en variables de entorno. Las variables de entorno se pueden mostrar en texto plano mediante herramientas como la CodeBuild consola y el AWS CLI. Para valores confidenciales, le recomendamos utilizar el mapeo `parameter-store` o `secrets-manager` en su lugar, tal y como se describe más adelante en esta sección.

Las variables de entorno que defina reemplazan las variables de entorno existentes. Por ejemplo, si la imagen de Docker ya contiene una variable de entorno denominada `MY_VAR` con un valor de `my_value` y establece una variable de entorno denominada `MY_VAR` con un valor de `other_value`, `my_value` se reemplaza por `other_value`. Asimismo, si la imagen de Docker ya contiene una variable de entorno denominada `PATH` con un valor de `/usr/local/sbin:/usr/local/bin` y establece una variable de entorno denominada `PATH` con un valor de `$PATH:/usr/share/ant/bin`, `/usr/local/sbin:/usr/local/bin` se reemplaza por el valor literal `$PATH:/usr/share/ant/bin`.

No establezca variables de entorno con un nombre que empiece por `CODEBUILD_`. Este prefijo se reserva para uso interno de .

Si se define una variable de entorno con el mismo nombre en varios lugares, el valor se determina de la siguiente manera:

- El valor de la llamada a la operación de inicio de la compilación tiene la máxima prioridad. Al crear una compilación, puede añadir o anular las variables de entorno. Para obtener más información, consulte [Ejecutar una compilación en AWS CodeBuild](#).
- El valor de la definición del proyecto de compilación es el siguiente en orden de prioridad. Al crear o editar un proyecto, puede añadir las variables de entorno en el nivel del proyecto. Para obtener más información, consulte [Creación de un proyecto de compilación en AWS CodeBuild](#) y [Cambiar la configuración de un proyecto de compilación en AWS CodeBuild](#).
- El valor en la declaración de especificación de compilación es el que menos prioridad tiene.

env/parameter-store

Obligatorio si se ha especificado `env` y desea recuperar variables de entorno personalizadas almacenadas en el almacén de parámetros de Amazon EC2 Systems Manager. Contiene una asignación de pares *clave/valor* escalares, donde cada asignación representa una única variable de entorno personalizada almacenada en el almacén de parámetros de Amazon EC2 Systems Manager. La *clave* es el nombre que utilizará más adelante en los comandos de compilación para hacer referencia a esta variable de entorno personalizada, y el *valor* es el nombre de la variable de entorno personalizada almacenada en el almacén de parámetros de Amazon EC2 Systems Manager. Para almacenar valores confidenciales, consulte [Almacén de parámetros de Systems Manager](#) y [Tutorial: Crear y probar un parámetro de cadena de caracteres \(consola\)](#) en la Guía del usuario de Amazon EC2 Systems Manager.

Important

CodeBuild Para permitir la recuperación de variables de entorno personalizadas almacenadas en el almacén de parámetros de Amazon EC2 Systems Manager, debe añadir `ssm:GetParameters` la acción a CodeBuild su rol de servicio. Para obtener más información, consulte [Crear un rol de servicio de CodeBuild](#).

Todas las variables de entorno que recupere del almacén de parámetros de Amazon EC2 Systems Manager reemplazan las variables de entorno existentes. Por ejemplo, si la imagen de Docker ya contiene una variable de entorno denominada `MY_VAR` con un valor de `my_value` y recupera una variable de entorno denominada `MY_VAR` con un valor

de `other_value`, `my_value` se reemplaza por `other_value`. Asimismo, si la imagen de Docker ya contiene una variable de entorno denominada `PATH` con un valor de `/usr/local/sbin:/usr/local/bin` y recupera una variable de entorno denominada `PATH` con un valor de `$PATH:/usr/share/ant/bin,/usr/local/sbin:/usr/local/bin` se reemplaza por el valor literal `$PATH:/usr/share/ant/bin`.

No almacene variables de entorno con un nombre que empiece por `CODEBUILD_`. Este prefijo se reserva para uso interno de .

Si se define una variable de entorno con el mismo nombre en varios lugares, el valor se determina de la siguiente manera:

- El valor de la llamada a la operación de inicio de la compilación tiene la máxima prioridad. Al crear una compilación, puede añadir o anular las variables de entorno. Para obtener más información, consulte [Ejecutar una compilación en AWS CodeBuild](#).
- El valor de la definición del proyecto de compilación es el siguiente en orden de prioridad. Al crear o editar un proyecto, puede añadir las variables de entorno en el nivel del proyecto. Para obtener más información, consulte [Creación de un proyecto de compilación en AWS CodeBuild](#) y [Cambiar la configuración de un proyecto de compilación en AWS CodeBuild](#).
- El valor en la declaración de especificación de compilación es el que menos prioridad tiene.

env/secrets-manager

Es obligatorio si desea recuperar las variables de entorno personalizadas almacenadas en AWS Secrets Manager. Especifique una `reference-key` de Secrets Manager utilizando el patrón siguiente:

```
<key>: <secret-id>:<json-key>:<version-stage>:<version-id>
```

```
<key>
```

(Obligatorio) Nombre de la variable de entorno local. Utilice este nombre para acceder a la variable durante la compilación.

```
<secret-id>
```

(Obligatorio) Nombre o nombre de recurso de Amazon (ARN) que sirve como identificador único del secreto. Para acceder a un secreto en su cuenta de AWS , basta con especificar

el nombre del secreto. Para acceder a un secreto de otra AWS cuenta, especifique el ARN secreto.

<json-key>

(Opcional) Especifica el nombre de la clave del par clave-valor de Secrets Manager cuyo valor desea recuperar. Si no especifica un `json-key`, CodeBuild recupera todo el texto secreto.

<version-stage>

(Opcional) Especifica la versión del secreto que desea recuperar mediante la etiqueta de fase asociada a la versión. Las etiquetas de fase se utilizan para realizar un seguimiento de las diferentes versiones durante el proceso de rotación. Si usa `version-stage`, no especifique `version-id`. Si no especifica una fase o un ID de versión, el valor predeterminado será recuperar la versión con el valor de la fase de versión `AWSCURRENT`.

<version-id>

(Opcional) Especifica el identificador único de la versión del secreto que desea utilizar. Si especifica `version-id`, no especifique `version-stage`. Si no especifica una fase o un ID de versión, el valor predeterminado será recuperar la versión con el valor de la fase de versión `AWSCURRENT`.

En el ejemplo siguiente, `TestSecret` es el nombre del par clave-valor almacenado en Secrets Manager. La clave de `TestSecret` es `MY_SECRET_VAR`. Se accede a la variable durante la compilación utilizando el nombre de `LOCAL_SECRET_VAR`.

```
env:  
  secrets-manager:  
    LOCAL_SECRET_VAR: "TestSecret:MY_SECRET_VAR"
```

Para obtener más información, consulte [¿Qué es AWS Secrets Manager?](#) en la Guía del usuario de AWS Secrets Manager .

`env/exported-variables`

Mapeo opcional. Se utiliza para enumerar las variables de entorno que desea exportar. Especifique el nombre de cada variable en la que desee exportar en una línea independiente `exported-variables`. La variable que desea exportar debe estar disponible en su contenedor durante la compilación. La variable exportada puede ser una variable de entorno.

Las variables de entorno exportadas se utilizan junto con ellas AWS CodePipeline para exportar variables de entorno desde la fase de creación actual a las etapas siguientes de la fase de

procesamiento. Para obtener más información, consulte [Trabajar con variables](#) en la Guía del usuario de AWS CodePipeline .

Durante una compilación, el valor de una variable está disponible a partir de la fase `install`. Se puede actualizar entre el inicio de la fase `install` y el final de la fase `post_build`. Una vez finalizada la fase `post_build`, el valor de las variables exportadas no puede cambiar.

 Note

No se pueden exportar los siguientes elementos:

- Secretos del almacén de parámetros de Amazon EC2 Systems Manager especificados en el proyecto de compilación.
- Secretos de Secrets Manager especificados en el proyecto de compilación
- Variables de entorno que empiezan por `AWS_`.

`env/ git-credential-helper`

Mapeo opcional. Se usa para indicar si CodeBuild usa su asistente de credenciales de Git para proporcionar las credenciales de Git. `yessi` se usa. De lo contrario, seleccione `no` o `sin especificar`. Para obtener más información, consulte [gitcredentials](#) en el sitio web de Git.

 Note

`git-credential-helper` no es compatible con compilaciones desencadenadas por un webhook para un repositorio de Git público.

`proxy`

Secuencia opcional. Se utiliza para representar configuraciones si ejecuta la compilación en un servidor de proxy explícito. Para obtener más información, consulte [Ejecución de CodeBuild en un servidor proxy explícito](#).

`proxy/upload-artifacts`

Mapeo opcional. Establezca en `yes` si desea que la compilación de un servidor de proxy explícito cargue artefactos. El valor predeterminado es `no`.

proxy/logs

Mapeo opcional. `yes` Configúrelo para crear CloudWatch registros en un servidor proxy explícito. El valor predeterminado es `no`.

phases

Secuencia obligatoria. Representa los comandos que CodeBuild se ejecutan durante cada fase de la compilación.

Note

En la versión 0.1 de `buildspec`, CodeBuild ejecuta cada comando en una instancia independiente del shell predeterminado del entorno de compilación. Esto significa que cada comando se ejecuta con independencia de los demás. Por lo tanto, de forma predeterminada, no puede ejecutar un comando que se base en el estado de comandos anteriores (por ejemplo, cambiar directorios o configurar variables de entorno). Para solventar esta limitación, le recomendamos utilizar la versión 0.2, que soluciona este problema. Si debe utilizar la versión de especificación de compilación 0.1, se recomiendan los enfoques que se describen en [Intérpretes de comandos y comandos de los entornos de compilación](#).

phases/*/run-as

Secuencia opcional. Utilice una fase de compilación para especificar un usuario de Linux que ejecuta sus comandos. Si `run-as` también se especifica globalmente para todos los comandos en la parte superior del archivo `buildspec`, entonces el usuario de nivel de fase tiene prioridad. Por ejemplo, si `run-as` especifica globalmente `User-1` y para la fase `install` solo una instrucción `run-as` especifica `User-2`, todos los comandos del archivo `buildspec` se ejecutan como `User-1` excepto los comandos de la fase `install`, que se ejecutan como `User-2`.

phases/*/on-failure

Secuencia opcional. Especifica la acción que se debe realizar si se produce un error durante la fase. Puede ser uno de los valores siguientes:

- `ABORT`: anular la compilación.
- `CONTINUE`: continuar con el paso siguiente.

Si no se especifica esta propiedad, el proceso de fallo sigue las fases de transición, como se muestra en [Transiciones de fases de compilación](#).

`phases/*/finally`

Bloque opcional. Los comandos especificados en un bloque `finally` se ejecutan después de los del bloque `commands`. Los comandos de un bloque `finally` se ejecutan aunque falle un comando del bloque `commands`. Por ejemplo, si el `commands` bloque contiene tres comandos y el primero falla, CodeBuild omite los dos comandos restantes y ejecuta todos los comandos del bloque. `finally` Se considera que la fase es satisfactoria cuando todos los comandos de los bloques `commands` y `finally` se ejecutan sin problemas. Si un comando de una fase falla, se considera que la fase falla.

Los nombres de las fases de compilación permitidos son:

`phases/install`

Secuencia opcional. Representa los comandos, si los hay, que CodeBuild se ejecutan durante la instalación. Le recomendamos que utilice la fase `install` únicamente para instalar paquetes en el entorno de compilación. Por ejemplo, puede utilizar esta fase para instalar una plataforma de comprobación de código como Mocha o RSpec.

`phases/install/runtime-versions`

Secuencia opcional. Una versión del entorno en tiempo de ejecución es compatible con la imagen estándar de Ubuntu 5.0 o una versión posterior y la imagen estándar de Amazon Linux 2 4.0 o una versión posterior. Si se especifica, al menos debe haber un entorno de tiempo de ejecución incluido en esta sección. Especifique un tiempo de ejecución mediante una versión específica, una versión principal seguida de una versión principal `.x` para especificar si CodeBuild utiliza esa versión principal con su versión secundaria más reciente o `latest` si utiliza la versión principal y secundaria más reciente (por ejemplo, `ruby: 3.2`, `nodejs: 18.x`, `ojava: latest`). Puede especificar el entorno de tiempo de ejecución mediante un número o una variable de entorno. Por ejemplo, si utiliza la imagen de Amazon Linux 2 estándar 4.0, lo siguiente especifica que la versión 17 de Java, la versión secundaria más reciente de python versión 3 y una versión contenida en una variable de entorno de Ruby están instaladas. Para obtener más información, consulte [Imágenes de Docker proporcionadas por CodeBuild](#).

```
phases:
```

```
install:
  runtime-versions:
    java: corretto8
    python: 3.x
    ruby: "$MY_RUBY_VAR"
```

Puede especificar uno o más tiempos de ejecución en la sección `runtime-versions` del archivo de especificación de compilación. Si el tiempo de ejecución depende de otro tiempo de ejecución, también puede especificar el tiempo de ejecución dependiente en el archivo de especificación de compilación. Si no especificas ningún tiempo de ejecución en el archivo `buildspec`, CodeBuild selecciona los tiempos de ejecución predeterminados que estén disponibles en la imagen que utilices. Si especificas uno o más tiempos de ejecución, utiliza solo esos tiempos de ejecución. CodeBuild Si no se especifica un tiempo de ejecución dependiente, CodeBuild intentará elegir el tiempo de ejecución dependiente por usted.

Si dos runtimes especificados están en conflicto, la compilación produce un error. Por ejemplo, `android: 29` y `java: openjdk11` están en conflicto, por lo que si se especifican ambos, la compilación produce un error.

Para obtener más información sobre los entornos de tiempo de ejecución disponibles, consulte [Tiempos de ejecución disponibles](#).

Note

Si se especifica una sección de `runtime-versions` y se utiliza una imagen distinta de Ubuntu Standard Image 2.0 o posterior, o la imagen estándar de Amazon Linux 2 (AL2) 1.0 o posterior, la compilación mostrará la advertencia "Skipping install of runtimes. Runtime version selection is not supported by this build image".

phases/install/commands

Secuencia opcional. Contiene una secuencia de escalares, donde cada escalar representa un único comando que CodeBuild se ejecuta durante la instalación. CodeBuild ejecuta cada comando, uno a la vez, en el orden indicado, de principio a fin.

phases/pre_build

Secuencia opcional. Representa los comandos, si los hay, que CodeBuild se ejecutan antes de la compilación. Por ejemplo, puede utilizar esta fase para iniciar sesión en Amazon ECR o puede instalar dependencias npm.

phases/pre_build/commands

Secuencia obligatoria si se especifica `pre_build`. Contiene una secuencia de escalares, donde cada escalar representa un único comando que CodeBuild se ejecuta antes de la compilación. CodeBuild ejecuta cada comando, uno a la vez, en el orden indicado, de principio a fin.

phases/build

Secuencia opcional. Representa los comandos, si los hay, que CodeBuild se ejecutan durante la compilación. Por ejemplo, puede utilizar esta fase para ejecutar Mocha, RSpec o sbt.

phases/build/commands

Es obligatorio si se ha especificado `build`. Contiene una secuencia de escalares, donde cada escalar representa un único comando que CodeBuild se ejecuta durante la compilación. CodeBuild ejecuta cada comando, uno a la vez, en el orden indicado, de principio a fin.

phases/post_build

Secuencia opcional. Representa los comandos, si los hay, que CodeBuild se ejecutan después de la compilación. Por ejemplo, puede utilizar Maven para empaquetar los artefactos de la compilación en un archivo JAR o WAR, o puede remitir una imagen de Docker hacia Amazon ECR. A continuación, puede enviar una notificación de compilación a través de Amazon SNS.

phases/post_build/commands

Es obligatorio si se ha especificado `post_build`. Contiene una secuencia de escalares, donde cada escalar representa un único comando que CodeBuild se ejecuta después de la compilación. CodeBuild ejecuta cada comando, uno a la vez, en el orden indicado, de principio a fin.

informes

report-group-name-or-arn

Secuencia opcional. Especifica el grupo de informes al que se envían los informes. Un proyecto puede tener un máximo de cinco grupos de informes. Especifique el ARN de un grupo de informes existente o el nombre de un nuevo grupo de informes. Si especifica un nombre, CodeBuild crea un grupo de informes con el nombre del proyecto y el nombre que especifique en el formato. `<project-name>-<report-group-name>` El nombre del grupo de informes también se puede establecer mediante una variable de entorno en la especificación de compilación, como. `$REPORT_GROUP_NAME` Para obtener más información, consulte [Nomenclatura de grupos de informes](#).

reports/<grupo-informes>/files

Secuencia obligatoria. Representa las ubicaciones que contienen los datos sin procesar de los resultados de las pruebas generados por el informe. Contiene una secuencia de escalares, en la que cada escalar representa una ubicación independiente donde se CodeBuild pueden encontrar los archivos de prueba, en relación con la ubicación de compilación original o, si está establecida, con la. `base-directory` Las ubicaciones pueden ser las siguientes:

- Un archivo único (por ejemplo, `my-test-report-file.json`).
- Un único archivo de un subdirectorio (por ejemplo, `my-subdirectory/my-test-report-file.json` o `my-parent-subdirectory/my-subdirectory/my-test-report-file.json`).
- `'**/* '` representa todos los archivos recursivamente.
- `my-subdirectory/*` representa todos los archivos de un subdirectorio denominado `my-subdirectory` .
- `my-subdirectory/**/*` representa todos los archivos recursivamente a partir de un subdirectorio denominado `my-subdirectory` .

reports/<grupo-informes>/file-format

Mapeo opcional. Representa el formato del archivo de informe. Si no se ha especificado, se utiliza JUNITXML. Este valor no distingue entre mayúsculas y minúsculas. Los valores posibles son los siguientes:

Informes de pruebas

CUCUMBERJSON

Cucumber JSON

JUNITXML

JUnit XML

NUNITXML

NUnit XML

NUNIT3XML

NUnit 3 XML

TESTNGXML

TestNG XML

VISUALSTUDIOTRX

Visual Studio TRX

Informes de cobertura de código

CLOVERXML

Clover XML

COBERTURAXML

Cobertura XML

JACOCOXML

JaCoCo XML

SIMPLECOV

SimpleCov JSON

Note

CodeBuild [acepta los informes de cobertura de código JSON generados por simplecov, no por simplecov-json.](#)

reports/<grupo-informes>/base-directory

Mapeo opcional. Representa uno o más directorios de nivel superior, en relación con la ubicación de compilación original, que se CodeBuild utilizan para determinar dónde encontrar los archivos de prueba sin procesar.

reports/<grupo-informes>/discard-paths

Opcional. Especifica si los directorios del archivo de informes se aplanan en la salida. Si esto no se especifica o contiene no, los archivos de informes se generan con su estructura de directorios intacta. Si esto contiene yes, todos los archivos de prueba se colocan en el mismo directorio de salida. Por ejemplo, si una ruta a un resultado de prueba es `com/myapp/mytests/TestResult.xml`, especificar `yes` colocará este archivo en `/TestResult.xml`.

artefactos

Secuencia opcional. Representa información sobre dónde se CodeBuild puede encontrar el resultado de la compilación y cómo CodeBuild se prepara para cargarlo en el depósito de salida de S3. Esta secuencia no es necesaria si, por ejemplo, va a crear e insertar una imagen de Docker en o si va a ejecutar pruebas unitarias en el código fuente pero no lo va a compilar.

Note

Los metadatos de Amazon S3 tienen un nombre de CodeBuild encabezado `x-amz-meta-codebuild-buildarn` que contiene el nombre `buildArn` de la CodeBuild compilación que publica los artefactos en Amazon S3. Se añade `buildArn` para permitir el seguimiento de las notificaciones en la fuente y para hacer referencia a la compilación de donde procede el artefacto.

artifacts/files

Secuencia obligatoria. Representa las ubicaciones que contienen los artefactos de salida de la compilación en el entorno de compilación. Contiene una secuencia de escalares, cada uno de los cuales representa una ubicación independiente en la que se CodeBuild pueden encontrar los artefactos de salida de la compilación, en relación con la ubicación de compilación original o, si está establecida, con el directorio base. Las ubicaciones pueden ser las siguientes:

- Un archivo único (por ejemplo, `my-file.jar`).

- Un único archivo de un subdirectorio (por ejemplo, *my-subdirectory*/my-file.jar o *my-parent-subdirectory/my-subdirectory/my-file.jar*).
- `'**/*'` representa todos los archivos recursivamente.
- *my-subdirectory*/* representa todos los archivos de un subdirectorio denominado *my-subdirectory*.
- *my-subdirectory*/**/* representa todos los archivos recursivamente a partir de un subdirectorio denominado *my-subdirectory*.

Al especificar las ubicaciones de los artefactos de salida de la construcción, CodeBuild puede localizar la ubicación de construcción original en el entorno de construcción. No tiene que anexar las ubicaciones de los artefactos de salida de la compilación a la ruta de acceso de la ubicación de la compilación original ni especificar `./` o similar. Si desea conocer la ruta a esta ubicación, puede ejecutar un comando como `echo $CODEBUILD_SRC_DIR` durante una compilación. La ubicación de cada entorno de compilación puede ser ligeramente diferente.

artifacts/name

Nombre opcional. Especifica un nombre para su artefacto de compilación. Este nombre se utiliza cuando se cumple alguna de las condiciones siguientes.

- Usas la CodeBuild API para crear tus compilaciones y el `overrideArtifactName` indicador se establece en el `ProjectArtifacts` objeto cuando se actualiza un proyecto, se crea un proyecto o se inicia una compilación.
- Usas la CodeBuild consola para crear tus compilaciones, se especifica un nombre en el archivo `buildspec` y seleccionas `Habilitar el control de versiones semántico` al crear o actualizar un proyecto. Para obtener más información, consulte [Creación de un proyecto de compilación \(consola\)](#).

Puede especificar un nombre en el archivo de especificación de compilación que se calcula en el momento de la compilación. El nombre especificado en un archivo de especificación utiliza el lenguaje de comandos Shell. Por ejemplo, puede adjuntar una fecha y una hora al nombre del artefacto para que siempre sea único. Los nombres de artefactos únicos impiden que los artefactos se sobrescriban. Para obtener más información, consulte [Lenguaje de comandos Shell](#).

- Este es un ejemplo de una nombre de artefacto asociado con la fecha de creación del artefacto.

```
version: 0.2
phases:
  build:
    commands:
```

```

    - rspec HelloWorld_spec.rb
artifacts:
  files:
    - '**/*'
  name: myname-$(date +%Y-%m-%d)

```

- Este es un ejemplo de un nombre de artefacto que usa una variable de entorno. CodeBuild Para obtener más información, consulte [Variables de entorno en los entornos de compilación](#).

```

version: 0.2
phases:
  build:
    commands:
      - rspec HelloWorld_spec.rb
artifacts:
  files:
    - '**/*'
  name: myname-$AWS_REGION

```

- Este es un ejemplo de un nombre de artefacto que usa una variable de CodeBuild entorno con la fecha de creación del artefacto adjunta.

```

version: 0.2
phases:
  build:
    commands:
      - rspec HelloWorld_spec.rb
artifacts:
  files:
    - '**/*'
  name: $AWS_REGION-$(date +%Y-%m-%d)

```

Puede añadir información sobre la ruta al nombre para que los artefactos nombrados se coloquen en directorios según la ruta que figure en el nombre. En este ejemplo, los artefactos de compilación se colocan en la salida dentro de `builds/<build number>/my-artifacts`.

```

version: 0.2
phases:
  build:
    commands:
      - rspec HelloWorld_spec.rb
artifacts:

```

```
files:
  - '**/*'
name: builds/$CODEBUILD_BUILD_NUMBER/my-artifacts
```

artifacts/discard-paths

Opcional. Especifica si los directorios de artefactos de compilación se aplanan en la salida. Si esto no se especifica o contiene no, los artefactos de compilación se generan con su estructura de directorios intacta. Si esto contiene yes, todos los artefactos de compilación se colocan en el mismo directorio de salida. Por ejemplo, si una ruta a un archivo en el artefacto de salida de compilación es `com/mycompany/app/HelloWorld.java`, especificar `yes` colocará este archivo en `/HelloWorld.java`.

artifacts/base-directory

Mapeo opcional. Representa uno o más directorios de nivel superior, en relación con la ubicación de compilación original, que se CodeBuild utilizan para determinar qué archivos y subdirectorios incluir en el artefacto de salida de la compilación. Los valores válidos son:

- Un único directorio de nivel superior (por ejemplo, `my-directory`).
- `'my-directory*'` representa todos los directorios de nivel superior con nombres que empiezan por `my-directory`.

Los directorios de nivel superior coincidentes no se incluyen en el artefacto de salida de la compilación, solo sus archivos y subdirectorios.

Puede utilizar `files` y `discard-paths` para restringir aún más los archivos y subdirectorios que se incluyen. Por ejemplo, para la siguiente estructura de directorios:

```
.
### my-build-1
#   ### my-file-1.txt
### my-build-2
    ### my-file-2.txt
    ### my-subdirectory
        ### my-file-3.txt
```

Y para la siguiente secuencia `artifacts`:

```
artifacts:
  files:
    - '*/my-file-3.txt'
```

```
base-directory: my-build-2
```

Se incluiría el siguiente subdirectorio y archivo en el artefacto de salida de la compilación:

```
.  
### my-subdirectory  
    ### my-file-3.txt
```

Sin embargo, para la siguiente secuencia `artifacts`:

```
artifacts:  
  files:  
    - '**/*'  
  base-directory: 'my-build*'  
  discard-paths: yes
```

Se incluirían los siguientes archivos en el artefacto de salida de la compilación:

```
.  
### my-file-1.txt  
### my-file-2.txt  
### my-file-3.txt
```

artifacts/exclude-paths

Mapeo opcional. Representa una o más rutas, relativas a `base-directory`, que CodeBuild se excluirán de los artefactos de construcción. El carácter asterisco (*) coincide con cero o varios caracteres de un componente de nombre sin superar límites de carpeta. Un asterisco doble (**) coincide con cero o más caracteres de un componente de nombre en todos los directorios.

A continuación, se muestran ejemplos de `exclude-path`:

- Para excluir un archivo de todos los directorios: `**/file-name/**/*`
- Para excluir todas las carpetas de punto: `**/.*/**/*`
- Para excluir todos los archivos de punto: `**/*.*`

artifacts/enable-symlinks

Opcional. Si el tipo de salida es ZIP, esto especifica si los enlaces simbólicos internos se conservan en el archivo ZIP. Si esto contiene `yes`, todos los enlaces simbólicos internos de la fuente se conservarán en el archivo ZIP de artefactos.

artifacts/s3-prefix

Opcional. Especifica un prefijo que se utiliza cuando los artefactos se envían a un bucket de Amazon S3 y el tipo de espacio de nombres es BUILD_ID. Cuando se usa, la ruta de salida del bucket es `<s3-prefix>/<build-id>/<name>.zip`.

artifacts/secondary-artifacts

Secuencia opcional. Representa una o varias definiciones de artefacto como mapeo entre un identificador de artefacto y una definición de este. Cada uno de los identificadores de artefacto de este bloque debe coincidir con un artefacto definido en el atributo `secondaryArtifacts` del proyecto. Todas las definiciones tienen la misma sintaxis que el bloque `artifacts` anterior.

Note

La secuencia de [artifacts/files](#) siempre es obligatoria, incluso si solo se han definido artefactos secundarios.

Por ejemplo, si el proyecto tiene la estructura siguiente:

```
{
  "name": "sample-project",
  "secondaryArtifacts": [
    {
      "type": "S3",
      "location": "<output-bucket1>",
      "artifactIdentifier": "artifact1",
      "name": "secondary-artifact-name-1"
    },
    {
      "type": "S3",
      "location": "<output-bucket2>",
      "artifactIdentifier": "artifact2",
      "name": "secondary-artifact-name-2"
    }
  ]
}
```

El archivo `buildspec` tiene este aspecto:

```
version: 0.2
```

```
phases:
build:
  commands:
    - echo Building...
artifacts:
  files:
    - '**/*'
secondary-artifacts:
  artifact1:
    files:
      - directory/file1
    name: secondary-artifact-name-1
  artifact2:
    files:
      - directory/file2
    name: secondary-artifact-name-2
```

cache

Secuencia opcional. Representa información sobre dónde CodeBuild se pueden preparar los archivos para cargar la caché en un depósito de caché de S3. Esta secuencia no es necesaria si el tipo de caché del proyecto es No Cache.

cache/paths

Secuencia obligatoria. Representa las ubicaciones de la caché. Contiene una secuencia de escalares, cada uno de los cuales representa una ubicación independiente en la que se CodeBuild pueden encontrar los artefactos de salida de la compilación, en relación con la ubicación de compilación original o, si está establecida, con el directorio base. Las ubicaciones pueden ser las siguientes:

- Un archivo único (por ejemplo, `my-file.jar`).
- Un único archivo de un subdirectorio (por ejemplo, `my-subdirectory/my-file.jar` o `my-parent-subdirectory/my-subdirectory/my-file.jar`).
- `'**/*'` representa todos los archivos recursivamente.
- `my-subdirectory/*` representa todos los archivos de un subdirectorio denominado `my-subdirectory`.
- `my-subdirectory/**/*` representa todos los archivos recursivamente a partir de un subdirectorio denominado `my-subdirectory`.

⚠ Important

Como una declaración de especificación de compilación debe ser una declaración YAML válida, los espacios de la declaración son importantes. Si el número de espacios en la declaración de la especificación de compilación no es válido, es posible que las compilaciones produzcan un error inmediatamente. Puede utilizar un validador YAML para comprobar si sus declaraciones de especificación de compilación son declaraciones YAML válidas.

Si utilizas el o los AWS CLI AWS SDK para declarar una especificación de construcción al crear o actualizar un proyecto de compilación, la especificación de compilación debe ser una cadena única expresada en formato YAML, junto con los espacios en blanco y los caracteres de escape de nueva línea necesarios. Encontrará un ejemplo en la siguiente sección.

Si utilizas las AWS CodePipeline consolas CodeBuild o en lugar del archivo `buildspec.yml`, solo puedes insertar comandos para la fase. `build` En lugar de utilizar la sintaxis anterior, incluirá en una sola línea todos los comandos que desea ejecutar durante la fase de compilación. En caso de que haya varios comandos, separe cada comando con `&&` (por ejemplo, `mvn test && mvn package`).

Puede usar las CodePipeline consolas CodeBuild o en lugar del archivo `buildspec.yml` para especificar las ubicaciones de los artefactos de salida de la compilación en el entorno de compilación. En lugar de utilizar la sintaxis anterior, incluirá en una sola línea todas las ubicaciones. Si hay varias ubicaciones, separe cada una de las ubicaciones con una coma (por ejemplo, `buildspec.yml, target/my-app.jar`).

Ejemplo de un archivo buildspec

A continuación se muestra un ejemplo de un archivo `buildspec.yml`.

```
version: 0.2

env:
  variables:
    JAVA_HOME: "/usr/lib/jvm/java-8-openjdk-amd64"
  parameter-store:
    LOGIN_PASSWORD: /CodeBuild/dockerLoginPassword

phases:
  install:
    commands:
```

```
- echo Entered the install phase...
- apt-get update -y
- apt-get install -y maven
finally:
- echo This always runs even if the update or install command fails
pre_build:
  commands:
    - echo Entered the pre_build phase...
    - docker login -u User -p $LOGIN_PASSWORD
  finally:
    - echo This always runs even if the login command fails
build:
  commands:
    - echo Entered the build phase...
    - echo Build started on `date`
    - mvn install
  finally:
    - echo This always runs even if the install command fails
post_build:
  commands:
    - echo Entered the post_build phase...
    - echo Build completed on `date`

reports:
arn:aws:codebuild:your-region:your-aws-account-id:report-group/report-group-name-1:
  files:
    - "**/*"
  base-directory: 'target/tests/reports'
  discard-paths: no
reportGroupCucumberJson:
  files:
    - 'cucumber/target/cucumber-tests.xml'
  discard-paths: yes
  file-format: CUCUMBERJSON # default is JUNITXML
artifacts:
  files:
    - target/messageUtil-1.0.jar
  discard-paths: yes
secondary-artifacts:
  artifact1:
    files:
      - target/artifact-1.0.jar
    discard-paths: yes
  artifact2:
```

```

files:
  - target/artifact-2.0.jar
discard-paths: yes
cache:
  paths:
    - '/root/.m2/**/*'

```

Este es un ejemplo de la especificación de compilación anterior, expresada como una sola cadena, para usarla con el o los SDK. AWS CLI AWS

```

"version: 0.2\n\nenv:\n  variables:\n    JAVA_HOME: \"/usr/lib/jvm/java-8-openjdk-
amd64\"
parameter-store:\n  LOGIN_PASSWORD: /CodeBuild/dockerLoginPassword\n
phases:\n\n  install:\n    commands:\n      - echo Entered the install phase...\n
- apt-get update -y\n      - apt-get install -y maven\n    finally:\n      - echo This
always runs even if the update or install command fails\n  pre_build:\n    commands:
\n      - echo Entered the pre_build phase...\n      - docker login -u User -p
$LOGIN_PASSWORD\n    finally:\n      - echo This always runs even if the login command
fails\n  build:\n    commands:\n      - echo Entered the build phase...\n      - echo
Build started on `date`\n      - mvn install\n    finally:\n      - echo This always
runs even if the install command fails\n  post_build:\n    commands:\n      - echo
Entered the post_build phase...\n      - echo Build completed on `date`\n\n  reports:
\n  reportGroupJUnitXml:\n    files:\n      - \"**/*\"\n    base-directory: 'target/
tests/reports'\n    discard-paths: false\n  reportGroupCucumberJson:\n    files:\n
- 'cucumber/target/cucumber-tests.xml'\n    file-format: CUCUMBERJSON\n\n  artifacts:\n
files:\n  - target/messageUtil-1.0.jar\n  discard-paths: yes\n  secondary-artifacts:
\n  artifact1:\n    files:\n      - target/messageUtil-1.0.jar\n    discard-
paths: yes\n  artifact2:\n    files:\n      - target/messageUtil-1.0.jar\n
discard-paths: yes\n  cache:\n  paths:\n    - '/root/.m2/**/*'"

```

Este es un ejemplo de los comandos de la build fase para usarlos con las consolas o. CodeBuild CodePipeline

```
echo Build started on `date` && mvn install
```

En estos ejemplos:

- Se establece una variable de entorno personalizada, en texto sin formato, con la clave `JAVA_HOME` y el valor `/usr/lib/jvm/java-8-openjdk-amd64`.
- Se hace referencia a una variable de entorno llamada `dockerLoginPassword` y almacenada en el almacén de parámetros de Amazon EC2 Systems Manager en comandos posteriores en la compilación mediante la clave `LOGIN_PASSWORD`.

- No puede cambiar estos nombres de fases de compilación. Los comandos que se ejecutan en este ejemplo son `apt-get update -y` y `apt-get install -y maven` (para instalar Apache Maven), `mvn install` (para compilar, probar y empaquetar el código fuente en un artefacto de salida de la compilación y para instalar el artefacto de salida de la compilación en su repositorio interno), `docker login` (para iniciar sesión en Docker con la contraseña correspondiente al valor de la variable de entorno personalizada `dockerLoginPassword` definida en el almacén de parámetros de Amazon EC2 Systems Manager) y varios comandos `echo`. Los `echo` comandos se incluyen aquí para mostrar cómo CodeBuild se ejecutan los comandos y el orden en que se ejecutan.
- `files` representa los archivos que se cargan en la ubicación de salida de la compilación. En este ejemplo, CodeBuild carga el único archivo `messageUtil-1.0.jar`. El archivo `messageUtil-1.0.jar` se encuentra en el directorio relativo denominado `target` en el entorno de compilación. Como se ha especificado `discard-paths: yes`, `messageUtil-1.0.jar` se carga directamente (y no en un directorio `target` intermedio). El nombre de archivo `messageUtil-1.0.jar` y el nombre del directorio relativo `target` se basan en la forma en la que Apache Maven crea y almacena los artefactos de salida de la compilación solo para este ejemplo. En sus propios escenarios, estos nombres de archivos y directorios serán diferentes.
- `reports` representa dos grupos de informes que generan informes durante la compilación:
 - `arn:aws:codebuild:your-region:your-aws-account-id:report-group/report-group-name-1` especifica el ARN de un grupo de informes. Los resultados de la prueba generados por el marco de prueba están en el directorio `target/tests/reports`. El formato del archivo es `JUnitXml` y la ruta no se elimina de los archivos que contienen los resultados de prueba.
 - `reportGroupCucumberJson` especifica un nuevo grupo de informes. Si el nombre del proyecto es `my-project`, se crea un grupo de informes con el nombre `my-project-reportGroupCucumberJson` cuando se ejecuta una compilación. Los resultados de prueba generados por el marco de pruebas están en `cucumber/target/cucumber-tests.xml`. El formato del archivo de prueba es `CucumberJson` y la ruta se elimina de los archivos que contienen los resultados de prueba.

Versión de buildspec

En la siguiente tabla se muestran las versiones de especificaciones de compilación y los cambios entre versiones.

Versión	Cambios
0.2	<ul style="list-style-type: none"> • <code>environment_variables</code> ahora se llama <code>env</code>. • <code>plaintext</code> ahora se llama <code>variables</code>. • La propiedad <code>type</code> de <code>artifacts</code> ya no se utiliza. • En la versión 0.1, AWS CodeBuild ejecuta cada comando de compilación en una instancia independiente del shell predeterminado del entorno de compilación. En la versión 0.2, CodeBuild ejecuta todos los comandos de compilación en la misma instancia del shell predeterminado del entorno de compilación.
0.1	Esta es la primera definición del formato de especificación de compilación.

Referencia de especificaciones de compilación para compilación por lotes

Este tema contiene la referencia de las especificaciones de compilación para las propiedades de compilación por lotes.

batch

Mapeo opcional. Configuración de compilación por lotes para el proyecto.

batch/fast-fail

Opcional. Especifica el comportamiento de la compilación por lotes cuando una o más tareas de compilación fallan.

`false`

Valor predeterminado. Se completarán todas las compilaciones en ejecución.

`true`

Todas las compilaciones en ejecución se detendrán cuando falle una de las tareas de compilación.

De forma predeterminada, todas las tareas de compilación por lotes se ejecutan con la configuración de compilación, como `env` y `phases`, especificada en el archivo de especificaciones de compilación. Es posible anular la configuración de compilación predeterminada especificando valores distintos de `env` o un archivo de especificaciones de compilación distinto en el parámetro `batch/<batch-type>/buildspec`.

El contenido de la propiedad `batch` varía en función del tipo de compilación por lotes que se especifique. Los tipos posibles de compilación por lotes son:

- [batch/build-graph](#)
- [batch/build-list](#)
- [batch/build-matrix](#)

batch/build-graph

Define un grafo de compilación. Un grafo de compilación define un conjunto de tareas que dependen de otras tareas del lote. Para obtener más información, consulte [Grafo de compilación](#).

Este elemento contiene una matriz de tareas de compilación. Cada tarea de compilación contiene las propiedades siguientes.

`identifier`

Obligatorio. Identificador de la tarea.

`buildspec`

Opcional. Ruta y nombre de archivo del archivo de especificaciones de compilación que se debe utilizar para esta tarea. Si no se especifica este parámetro, se utiliza el archivo de especificación de compilación actual.

`debug-session`

Opcional. Valor booleano que indica si la depuración de sesión está habilitada para esta compilación por lotes. Para obtener más información acerca de la depuración de sesión, consulte [Ver una compilación en ejecución en Session Manager](#).

`false`

La depuración de sesión está desactivada.

`true`

La depuración de sesión está activada.

`depend-on`

Opcional. Conjunto de identificadores de tareas de los que depende esta tarea. No se ejecutará esta tarea hasta que se hayan completado.

`env`

Opcional. Anulaciones del entorno de compilación para la tarea. Esto puede contener las propiedades siguientes:

`compute-type`

Identificador del tipo de computación que se va a utilizar para la tarea. Consulte `computeType` en [the section called “Modos y tipos de computación del entorno de compilación”](#) para ver los valores posibles.

`image`

Identificador de la imagen que se va a utilizar para la tarea. Consulte el identificador de imagen en [the section called “Imágenes de Docker proporcionadas por CodeBuild”](#) para ver los valores posibles.

`privileged-mode`

Valor booleano que indica si se debe ejecutar el daemon de Docker dentro de un contenedor de Docker. Se establece en `true` solo si el proyecto de compilación se utiliza para compilar imágenes de Docker. De lo contrario, se produce un error las compilaciones que intentan interactuar con el daemon de Docker. El ajuste predeterminado es `false`.

`type`

Identificador del tipo de entorno que se va a utilizar para la tarea. Consulte Tipo de entorno en [the section called “Modos y tipos de computación del entorno de compilación”](#) para ver los valores posibles.

`variables`

Variables de entorno que estarán presentes en el entorno de compilación. Para obtener más información, consulte [env/variables](#).

ignore-failure

Opcional. Valor booleano que indica si se puede ignorar un error en la tarea de compilación.

`false`

Valor predeterminado. Si esta tarea de compilación falla, se considerará fallida la compilación por lotes.

`true`

Si esta tarea de compilación falla, todavía es posible terminar la compilación por lotes de forma satisfactoria.

A continuación se muestra un ejemplo de una entrada de especificaciones de compilación en un grafo de compilación:

```
batch:
  fast-fail: false
  build-graph:
    - identifier: build1
      env:
        variables:
          BUILD_ID: build1
      ignore-failure: false
    - identifier: build2
      buildspec: build2.yml
      env:
        variables:
          BUILD_ID: build2
      depend-on:
        - build1
    - identifier: build3
      env:
        variables:
          BUILD_ID: build3
      depend-on:
        - build2
```

batch/build-list

Define una lista de compilación. Una lista de compilación sirve para definir una serie de tareas que se ejecutan en paralelo. Para obtener más información, consulte [Lista de compilación](#).

Este elemento contiene una matriz de tareas de compilación. Cada tarea de compilación contiene las propiedades siguientes.

identifier

Obligatorio. Identificador de la tarea.

buildspec

Opcional. Ruta y nombre de archivo del archivo de especificaciones de compilación que se debe utilizar para esta tarea. Si no se especifica este parámetro, se utiliza el archivo de especificación de compilación actual.

debug-session

Opcional. Valor booleano que indica si la depuración de sesión está habilitada para esta compilación por lotes. Para obtener más información acerca de la depuración de sesión, consulte [Ver una compilación en ejecución en Session Manager](#).

false

La depuración de sesión está desactivada.

true

La depuración de sesión está activada.

env

Opcional. Anulaciones del entorno de compilación para la tarea. Esto puede contener las propiedades siguientes:

compute-type

Identificador del tipo de computación que se va a utilizar para la tarea. Consulte computeType en [the section called “Modos y tipos de computación del entorno de compilación”](#) para ver los valores posibles.

image

Identificador de la imagen que se va a utilizar para la tarea. Consulte el identificador de imagen en [the section called “Imágenes de Docker proporcionadas por CodeBuild”](#) para ver los valores posibles.

privileged-mode

Valor booleano que indica si se debe ejecutar el daemon de Docker dentro de un contenedor de Docker. Se establece en `true` solo si el proyecto de compilación se utiliza para compilar imágenes de Docker. De lo contrario, se produce un error las compilaciones que intentan interactuar con el daemon de Docker. El ajuste predeterminado es `false`.

type

Identificador del tipo de entorno que se va a utilizar para la tarea. Consulte Tipo de entorno en [the section called “Modos y tipos de computación del entorno de compilación”](#) para ver los valores posibles.

variables

Variables de entorno que estarán presentes en el entorno de compilación. Para obtener más información, consulte [env/variables](#).

ignore-failure

Opcional. Valor booleano que indica si se puede ignorar un error en la tarea de compilación.

false

Valor predeterminado. Si esta tarea de compilación falla, se considerará fallida la compilación por lotes.

true

Si esta tarea de compilación falla, todavía es posible terminar la compilación por lotes de forma satisfactoria.

A continuación, se muestra una entrada de especificaciones de compilación de una lista de compilación:

```
batch:
  fast-fail: false
  build-list:
    - identifier: build1
      env:
        variables:
          BUILD_ID: build1
      ignore-failure: false
    - identifier: build2
      buildspec: build2.yml
```

```
env:  
  variables:  
    BUILD_ID: build2  
  ignore-failure: true
```

batch/build-matrix

Define una matriz de compilación. Una matriz de compilación define tareas con distintas configuraciones que se ejecutan en paralelo. CodeBuild crea una compilación aparte para cada combinación de configuración posible. Para obtener más información, consulte [Matriz de compilación](#).

static

Las propiedades estáticas se aplican a todas las tareas de compilación.

ignore-failure

Opcional. Valor booleano que indica si se puede ignorar un error en la tarea de compilación.

false

Valor predeterminado. Si esta tarea de compilación falla, se considerará fallida la compilación por lotes.

true

Si esta tarea de compilación falla, todavía es posible terminar la compilación por lotes de forma satisfactoria.

env

Opcional. El entorno de compilación se anula en todas las tareas.

privileged-mode

Valor booleano que indica si se debe ejecutar el daemon de Docker dentro de un contenedor de Docker. Se establece en `true` solo si el proyecto de compilación se utiliza para compilar imágenes de Docker. De lo contrario, se produce un error las compilaciones que intentan interactuar con el daemon de Docker. El ajuste predeterminado es `false`.

type

Identificador del tipo de entorno que se va a utilizar para la tarea. Consulte Tipo de entorno en [the section called “Modos y tipos de computación del entorno de compilación”](#) para ver los valores posibles.

dynamic

Las propiedades dinámicas definen la matriz de compilación.

buildspec

Opcional. Matriz que contiene la ruta y los nombres de los archivos de especificaciones de compilación que se van a utilizar en estas tareas. Si no se especifica este parámetro, se utiliza el archivo de especificación de compilación actual.

env

Opcional. El entorno de compilación anula estas tareas.

compute-type

Matriz que contiene los identificadores de los tipos de computación que se van a utilizar en estas tareas. Consulte computeType en [the section called “Modos y tipos de computación del entorno de compilación”](#) para ver los valores posibles.

image

Matriz que contiene los identificadores de las imágenes que se utilizarán en estas tareas. Consulte el identificador de imagen en [the section called “Imágenes de Docker proporcionadas por CodeBuild”](#) para ver los valores posibles.

variables

Matriz que contiene las variables de entorno que estarán presentes en los entornos de compilación para estas tareas. Para obtener más información, consulte [env/variables](#).

A continuación, se muestra un ejemplo de entrada de especificaciones de compilación de una matriz de compilación:

```
batch:
  build-matrix:
    static:
      ignore-failure: false
    dynamic:
      buildspec:
        - matrix1.yml
        - matrix2.yml
      env:
        variables:
          MY_VAR:
```

- VALUE1
- VALUE2
- VALUE3

Para obtener más información, consulte [Matriz de compilación](#).

Referencia de entornos de compilación de AWS CodeBuild

Si llama a AWS CodeBuild para ejecutar una compilación, deberá proporcionar información sobre el entorno que va a usar. Un entorno de compilación representa una combinación de sistema operativo, tiempo de ejecución del lenguaje de programación y herramientas que CodeBuild utiliza para ejecutar una compilación. Para obtener información sobre cómo funciona un entorno de compilación, consulte [Cómo funciona CodeBuild](#).

Un entorno de compilación contiene una imagen de Docker. Para obtener información, consulte el [glosario de Docker](#) en el sitio web de Docker Docs.

Cuando se proporciona información a CodeBuild sobre el entorno de compilación, es necesario especificar el identificador de una imagen de Docker en un tipo de repositorio compatible. Entre estos se incluyen el repositorio de imágenes de Docker de CodeBuild, imágenes disponibles públicamente en Docker Hub y repositorios de Amazon Elastic Container Registry (Amazon ECR) a los que su cuenta de AWS tiene permiso de acceso.

- Le recomendamos que utilice imágenes de Docker almacenadas en el repositorio de imágenes de Docker de CodeBuild, ya que están optimizadas para su uso con el servicio. Para obtener más información, consulte [Imágenes de Docker proporcionadas por CodeBuild](#).
- Para obtener el identificador de una imagen de Docker disponible públicamente almacenada en Docker Hub, consulte [Búsqueda de repositorios](#) en el sitio web de Docker Docs.
- Para obtener información sobre cómo trabajar con imágenes de Docker almacenadas en repositorios de Amazon ECR en su cuenta de AWS, consulte [Ejemplo de Amazon ECR](#).

Además de un identificador de imágenes de Docker, deberá especificar también el conjunto de recursos informáticos que utiliza el entorno. Para obtener más información, consulte [Modos y tipos de computación del entorno de compilación](#).

Temas

- [Imágenes de Docker proporcionadas por CodeBuild](#)
- [Modos y tipos de computación del entorno de compilación](#)

- [Intérpretes de comandos y comandos de los entornos de compilación](#)
- [Variables de entorno en los entornos de compilación](#)
- [Tareas en segundo plano en entornos de compilación](#)

Imágenes de Docker proporcionadas por CodeBuild

Una imagen compatible es la última versión principal de una imagen disponible CodeBuild y se actualiza con actualizaciones de versiones secundarias y de parches. CodeBuild optimiza la duración del aprovisionamiento de las compilaciones con imágenes compatibles almacenándolas en caché en [Amazon Machine Images \(AMI\) de la máquina](#). Si quiere aprovechar las ventajas del almacenamiento en caché y minimizar la duración del aprovisionamiento de la compilación, seleccione Usar siempre la imagen más reciente para esta versión en tiempo de ejecución en la sección Versión de imagen de la CodeBuild consola, en lugar de utilizar una versión más detallada, como `aws/codebuild/amazonlinux2-x86_64-standard:4.0-1.0.0`

CodeBuild actualiza con frecuencia la lista de imágenes de Docker para añadir las imágenes más recientes y eliminar las antiguas. Para obtener la lista más actualizada, realice alguna de las siguientes operaciones:

- En la CodeBuild consola, en el asistente de creación de proyectos de compilación o en la página Editar proyecto de compilación, en Imagen de entorno, selecciona Imagen gestionada. Elija valores en las listas desplegadas Operating system (Sistema operativo), Runtime (Tiempo de ejecución) y Runtime version (Versión de tiempo de ejecución). Para obtener más información, consulte [Creación de un proyecto de compilación \(consola\)](#) o [Cambiar la configuración de un proyecto de compilación \(consola\)](#).
- Para el AWS CLI, ejecute el `list-curated-environment-images` comando:

```
aws codebuild list-curated-environment-images
```

- Para los AWS SDK, llama a la `ListCuratedEnvironmentImages` operación correspondiente al lenguaje de programación de destino. Para obtener más información, consulte [Referencia de los SDK y las herramientas de AWS](#).

La imagen base de la plataforma Windows Server Core 2019 solo está disponible en las regiones siguientes:

- Este de EE. UU. (Norte de Virginia)

- Este de EE. UU. (Ohio)
- Oeste de EE. UU. (Oregón)
- Europa (Irlanda)

Imágenes de computación de EC2

AWS CodeBuild admite las siguientes imágenes de Docker que están disponibles para la computación de EC2. CodeBuild

Plataforma	Identificador de imagen	Definición
Amazon Linux 2	aws/codebuild/amazonlinux2-x86_64-standard:4.0	al2/standard/4.0
Amazon Linux 2023	aws/codebuild/amazonlinux2-x86_64-standard:5.0	al2/standard/5.0
Amazon Linux 2	aws/codebuild/amazonlinux2-x86_64-standard:corretto8	al2/standard/corretto8
Amazon Linux 2	aws/codebuild/amazonlinux2-x86_64-standard:corretto11	al2/standard/corretto11
Amazon Linux 2	aws/codebuild/amazonlinux2-aarch64-standard:2.0	al2/aarch64/standard/2.0
Amazon Linux 2023	aws/codebuild/amazonlinux2-aarch64-standard:3.0	al2/aarch64/standard/3.0
Ubuntu 20.04	aws/codebuild/standard:5.0	ubuntu/standard/5.0

Plataforma	Identificador de imagen	Definición
Ubuntu 22.04	aws/codebuild/standard:6.0	ubuntu/standard/6.0
Ubuntu 22.04	aws/codebuild/standard:7.0	ubuntu/standard/7.0
Windows Server Core 2019	aws/codebuild/windows-base:2019-1.0	N/A
Windows Server Core 2019	aws/codebuild/windows-base:2019-2.0	N/A
Windows Server Core 2019	aws/codebuild/windows-base:2019-3.0	N/A
Windows Server Core 2022	aws/codebuild/windows-base:2022-1.0	N/A

Imágenes de computación de Lambda

AWS CodeBuild admite las siguientes imágenes de Docker que están disponibles para su AWS Lambda procesamiento. CodeBuild

aarch64Arquitectura de

Plataforma	Identificador de imagen	Definición
Amazon Linux 2	aws/codebuild/amazonlinux-aarch64-lambda-standard:dotnet6	al-lambda/aarch64/dotnet6
Amazon Linux 2023	aws/codebuild/amazonlinux-aarch64-lambda-standard:dotnet8	al-lambda/aarch64/dotnet8

Plataforma	Identificador de imagen	Definición
Amazon Linux 2	<code>aws/codebuild/amazonlinux-aarch64-lambda-standard:go1.21</code>	al-lambda/aarch64/go1.21
Amazon Linux 2	<code>aws/codebuild/amazonlinux-aarch64-lambda-standard:corretto11</code>	al-lambda/aarch64/corretto11
Amazon Linux 2	<code>aws/codebuild/amazonlinux-aarch64-lambda-standard:corretto17</code>	al-lambda/aarch64/corretto17
Amazon Linux 2023	<code>aws/codebuild/amazonlinux-aarch64-lambda-standard:corretto21</code>	al-lambda/aarch64/corretto21
Amazon Linux 2	<code>aws/codebuild/amazonlinux-aarch64-lambda-standard:nodejs18</code>	al-lambda/aarch64/nodejs18
Amazon Linux 2023	<code>aws/codebuild/amazonlinux-aarch64-lambda-standard:nodejs20</code>	al-lambda/aarch64/nodejs20
Amazon Linux 2	<code>aws/codebuild/amazonlinux-aarch64-lambda-standard:python3.11</code>	al-lambda/aarch64/python3.11

Plataforma	Identificador de imagen	Definición
Amazon Linux 2023	aws/codebuild/amazonlinux-aarch64-lambda-standard:python3.12	al-lambda/aarch64/python3.12
Amazon Linux 2	aws/codebuild/amazonlinux-aarch64-lambda-standard:ruby3.2	al-lambda/aarch64/ruby3.2

x86_64Arquitectura de

Plataforma	Identificador de imagen	Definición
Amazon Linux 2	aws/codebuild/amazonlinux-x86_64-lambda-standard:dotnet6	al-lambda/x86_64/dotnet6
Amazon Linux 2023	aws/codebuild/amazonlinux-x86_64-lambda-standard:dotnet8	al-lambda/x86_64/dotnet8
Amazon Linux 2	aws/codebuild/amazonlinux-x86_64-lambda-standard:go1.21	al-lambda/x86_64/go1.21
Amazon Linux 2	aws/codebuild/amazonlinux-x86_64-lambda-standard:corretto11	al-lambda/x86_64/corretto11
Amazon Linux 2	aws/codebuild/amazonlinux-x86_64-lambda-standard:corretto17	al-lambda/x86_64/corretto17

Plataforma	Identificador de imagen	Definición
	<code>bda-standard:correcto17</code>	
Amazon Linux 2023	<code>aws/codebuild/amazonlinux-x86_64-lambda-standard:correcto21</code>	al-lambda/x86_64/corretto21
Amazon Linux 2	<code>aws/codebuild/amazonlinux-x86_64-lambda-standard:nodejs18</code>	al-lambda/x86_64/nodejs18
Amazon Linux 2023	<code>aws/codebuild/amazonlinux-x86_64-lambda-standard:nodejs20</code>	al-lambda/x86_64/nodejs20
Amazon Linux 2	<code>aws/codebuild/amazonlinux-x86_64-lambda-standard:python3.11</code>	al-lambda/x86_64/python3.11
Amazon Linux 2023	<code>aws/codebuild/amazonlinux-x86_64-lambda-standard:python3.12</code>	al-lambda/x86_64/python3.12
Amazon Linux 2	<code>aws/codebuild/amazonlinux-x86_64-lambda-standard:ruby3.2</code>	al-lambda/x86_64/ruby3.2

Imágenes obsoletas

Una imagen obsoleta es una imagen que ya no está almacenada en caché ni actualizada por CodeBuild. Una imagen obsoleta ya no recibe actualizaciones de versiones menores ni de parches y, dado que ya no se actualiza, es posible que su uso no sea seguro. Si tu CodeBuild proyecto está configurado para usar una versión de imagen anterior, el proceso de aprovisionamiento descargará esta imagen de Docker y la usará para crear el entorno de ejecución en contenedores, lo que puede aumentar la duración del aprovisionamiento y la duración total de la compilación.

CodeBuild ha dejado en desuso las siguientes imágenes de Docker. Puede seguir utilizando estas imágenes, pero no estarán almacenadas en caché en el host de compilación y, por lo tanto, los tiempos de aprovisionamiento serán mayores.

Plataforma	Identificador de imagen	Definición	Fecha de baja
Amazon Linux 2	aws/codebuild/ amazonlinux2- x86_64-st andard:3.0	al2/standard/3.0	9 de mayo de 2023
Ubuntu 18.04	aws/codebuild/ standard:4.0	ubuntu/standard/4.0	31 de marzo de 2023
Amazon Linux 2	aws/codebuild/ amazonlinux2- aarch64-s tandard:1.0	al2/aarch64/standa rd/1.0	31 de marzo de 2023
Ubuntu 18.04	aws/codebuild/ standard:3.0	ubuntu/standard/3.0	30 de junio de 2022
Amazon Linux 2	aws/codebuild/ amazonlinux2- x86_64-st andard:2.0	al2/standard/2.0	30 de junio de 2022

Temas

- [Tiempos de ejecución disponibles](#)
- [Versiones de tiempo de ejecución](#)

Tiempos de ejecución disponibles

Puede especificar uno o más tiempos de ejecución en la sección `runtime-versions` del archivo de especificación de compilación. Si el tiempo de ejecución depende de otro tiempo de ejecución, también puede especificar el tiempo de ejecución dependiente en el archivo de especificación de compilación. Si no especificas ningún tiempo de ejecución en el archivo `buildspec`, CodeBuild selecciona los tiempos de ejecución predeterminados que están disponibles en la imagen que utilizas. Si especificas uno o más tiempos de ejecución, utiliza solo esos tiempos de ejecución. CodeBuild Si no se especifica un tiempo de ejecución dependiente, CodeBuild intentará elegir el tiempo de ejecución dependiente por usted. Para obtener más información, consulte [Specify runtime versions in the buildspec file](#).

Temas

- [Motores de ejecución de imágenes de Linux](#)
- [Motores de ejecución de imágenes de Windows](#)

Motores de ejecución de imágenes de Linux

La siguiente tabla contiene los motores de ejecución disponibles y las imágenes de Linux estándar que los admiten.

Motores de ejecución de las plataformas Ubuntu y Amazon Linux

Nombre del entorno de tiempo de ejecución	Versión	Imágenes
dotnet	3.1	Amazon Linux 2 AArch64 estándar: 2.0 Ubuntu estándar: 5.0
	5.0	Ubuntu estándar: 5.0
	6.0	Amazon Linux 2 x86_64 Lambda estándar: dotnet6

Nombre del entorno de tiempo de ejecución	Versión	Imágenes
		<p>Amazon Linux 2 AArch64 Lambda estándar: dotnet6</p> <p>Amazon Linux 2 x86_64 estándar: 4.0</p> <p>Amazon Linux 2023 x86_64 estándar: 5.0</p> <p>Amazon Linux 2023 AArch64 estándar: 3.0</p> <p>Ubuntu estándar: 6.0</p> <p>Ubuntu estándar: 7.0</p>
	8.0	<p>Amazon Linux 2023 x86_64 estándar: 5.0</p> <p>Amazon Linux 2023 AArch64 estándar: 3.0</p> <p>Ubuntu estándar: 7.0</p>
golang	1.12	Amazon Linux 2 AArch64 estándar: 2.0
	1.13	Amazon Linux 2 AArch64 estándar: 2.0
	1.14	Amazon Linux 2 AArch64 estándar: 2.0
	1.15	Ubuntu estándar: 5.0
	1.16	Ubuntu estándar: 5.0

Nombre del entorno de tiempo de ejecución	Versión	Imágenes
	1.18	Amazon Linux 2 x86_64 estándar: 4.0 Ubuntu estándar: 6.0
	1,20	Amazon Linux 2023 x86_64 estándar: 5.0 Amazon Linux 2023 AArch64 estándar: 3.0 Ubuntu estándar: 7.0
	1.21	Amazon Linux 2 x86_64 Lambda estándar: go1.21 Amazon Linux 2 AArch64 Lambda estándar: go1.21 Amazon Linux 2023 x86_64 estándar: 5.0 Amazon Linux 2023 AArch64 estándar: 3.0 Ubuntu estándar: 7.0
	1.22	Amazon Linux 2023 x86_64 estándar: 5.0 Ubuntu estándar: 7.0

Nombre del entorno de tiempo de ejecución	Versión	Imágenes
java	corretto8	<p>Amazon Linux 2 x86_64 estándar: corretto8</p> <p>Amazon Linux 2023 x86_64 estándar: 5.0</p> <p>Amazon Linux 2 AArch64 estándar: 2.0</p> <p>Ubuntu estándar: 5.0</p> <p>Ubuntu estándar: 7.0</p>
	corretto11	<p>Amazon Linux 2 x86_64 estándar: corretto11</p> <p>Amazon Linux 2 x86_64 Lambda estándar: corretto11</p> <p>Amazon Linux 2023 x86_64 estándar: 5.0</p> <p>Amazon Linux 2 AArch64 Lambda estándar: corretto11</p> <p>Amazon Linux 2 AArch64 estándar: 2.0</p> <p>Ubuntu estándar: 5.0</p> <p>Ubuntu estándar: 7.0</p>

Nombre del entorno de tiempo de ejecución	Versión	Imágenes
	corretto17	<p>Amazon Linux 2 x86_64 Lambda estándar: corretto17</p> <p>Amazon Linux 2 AArch64 Lambda estándar: corretto17</p> <p>Amazon Linux 2 x86_64 estándar: 4.0</p> <p>Amazon Linux 2023 x86_64 estándar: 5.0</p> <p>Amazon Linux 2023 AArch64 estándar: 3.0</p> <p>Ubuntu estándar: 6.0</p> <p>Ubuntu estándar: 7.0</p>
	corretto21	<p>Amazon Linux 2 x86_64 Lambda estándar: corretto21</p> <p>Amazon Linux 2 AArch64 Lambda estándar: corretto21</p> <p>Amazon Linux 2023 x86_64 estándar: 5.0</p> <p>Amazon Linux 2023 AArch64 estándar: 3.0</p> <p>Ubuntu estándar: 7.0</p>
nodejs	10	Amazon Linux 2 AArch64 estándar: 2.0

Nombre del entorno de tiempo de ejecución	Versión	Imágenes
	12	Amazon Linux 2 AArch64 estándar: 2.0 Ubuntu estándar: 5.0
	14	Ubuntu estándar: 5.0
	16	Amazon Linux 2 x86_64 estándar: 4.0 Ubuntu estándar: 6.0
	18	Amazon Linux 2 x86_64 Lambda estándar: nodejs18 Amazon Linux 2 AArch64 Lambda estándar: nodejs18 Amazon Linux 2023 x86_64 estándar: 5.0 Amazon Linux 2023 AArch64 estándar: 3.0 Ubuntu estándar: 7.0

Nombre del entorno de tiempo de ejecución	Versión	Imágenes
	20	<p>Amazon Linux 2 x86_64 Lambda estándar: nodejs20</p> <p>Amazon Linux 2 AArch64 Lambda estándar: nodejs20</p> <p>Amazon Linux 2023 x86_64 estándar: 5.0</p> <p>Amazon Linux 2023 AArch64 estándar: 3.0</p> <p>Ubuntu estándar: 7.0</p>
php	7.3	<p>Amazon Linux 2 AArch64 estándar: 2.0</p> <p>Ubuntu estándar: 5.0</p>
	7.4	<p>Amazon Linux 2 AArch64 estándar: 2.0</p> <p>Ubuntu estándar: 5.0</p>
	8.0	<p>Ubuntu estándar: 5.0</p>
	8.1	<p>Amazon Linux 2 x86_64 estándar: 4.0</p> <p>Amazon Linux 2023 AArch64 estándar: 3.0</p> <p>Ubuntu estándar: 6.0</p>

Nombre del entorno de tiempo de ejecución	Versión	Imágenes
	8.2	Amazon Linux 2023 x86_64 estándar: 5.0 Amazon Linux 2023 AArch64 estándar: 3.0 Ubuntu estándar: 7.0
	8.3	Amazon Linux 2023 x86_64 estándar: 5.0 Amazon Linux 2023 AArch64 estándar: 3.0 Ubuntu estándar: 7.0
python	3.7	Amazon Linux 2 AArch64 estándar: 2.0 Ubuntu estándar: 5.0
	3.8	Amazon Linux 2 AArch64 estándar: 2.0 Ubuntu estándar: 5.0
	3.9	Amazon Linux 2 x86_64 estándar: 4.0 Amazon Linux 2023 x86_64 estándar: 5.0 Amazon Linux 2 AArch64 estándar: 2.0 Ubuntu estándar: 5.0 Ubuntu estándar: 7.0

Nombre del entorno de tiempo de ejecución	Versión	Imágenes
	3.10	<p>Amazon Linux 2023 x86_64 estándar: 5.0</p> <p>Ubuntu estándar: 6.0</p> <p>Ubuntu estándar: 7.0</p>
	3.11	<p>Amazon Linux 2 x86_64 Lambda estándar: python3.11</p> <p>Amazon Linux 2 AArch64 Lambda estándar: python3.11</p> <p>Amazon Linux 2023 x86_64 estándar: 5.0</p> <p>Amazon Linux 2023 AArch64 estándar: 3.0</p> <p>Ubuntu estándar: 7.0</p>
	3.12	<p>Amazon Linux 2 x86_64 Lambda estándar: python3.12</p> <p>Amazon Linux 2 AArch64 Lambda estándar: python3.12</p> <p>Amazon Linux 2023 x86_64 estándar: 5.0</p> <p>Amazon Linux 2023 AArch64 estándar: 3.0</p> <p>Ubuntu estándar: 7.0</p>

Nombre del entorno de tiempo de ejecución	Versión	Imágenes
ruby	2.6	Amazon Linux 2 AArch64 estándar: 2.0 Ubuntu estándar: 5.0
	2.7	Amazon Linux 2 AArch64 estándar: 2.0 Ubuntu estándar: 5.0
	3.1	Amazon Linux 2 x86_64 estándar: 4.0 Amazon Linux 2023 x86_64 estándar: 5.0 Ubuntu estándar: 6.0 Ubuntu estándar: 7.0
	3.2	Amazon Linux 2 x86_64 Lambda estándar: ruby3.2 Amazon Linux 2 AArch64 Lambda estándar: ruby3.2 Amazon Linux 2023 x86_64 estándar: 5.0 Amazon Linux 2023 AArch64 estándar: 3.0 Ubuntu estándar: 7.0
	3.3	Amazon Linux 2023 x86_64 estándar: 5.0 Ubuntu estándar: 7.0

Motores de ejecución de imágenes de Windows

La imagen base de Windows Server Core 2019 contiene los siguientes motores de ejecución siguientes.

Motores de ejecución de plataforma Windows

Nombre del entorno de tiempo de ejecución	Estándar de Windows Server Core 2019: versiones 1.0	Estándar de Windows Server Core 2019: versiones 2.0	Estándar de Windows Server Core 2019: versiones 3.0
dotnet	3.1	3.1	6.0
	5.0	6.0	7.0
		7.0	8.0
dotnet sdk	3.1	3.1	8.0
	5.0	6.0	
		7.0	
golang	1.14	1.18	1.21
gradle	6.7	7.6	8.5
java	Corretto 11	Corretto 11 Corretto 17	Corretto 21
maven	3.6	3.8	3.9
nodejs	14.15	16.19	20,11
php	7.4	8.1	8.3
powershell	7.1	7.2	7.4
python	3.8	3.10	3.12
ruby	2.7	3.1	3.3

Versiones de tiempo de ejecución

Cuando especifica un tiempo de ejecución en la sección [runtime-versions](#) del archivo buildspec, puede especificar una versión específica, una versión principal específica y la última versión secundaria o la última versión. En la tabla siguiente se muestran los tiempos de ejecución disponibles y cómo especificarlos. No todas las versiones en tiempo de ejecución están disponibles en todas las imágenes. La selección de la versión en tiempo de ejecución tampoco es compatible con las imágenes personalizadas. Para obtener más información, consulte [Tiempos de ejecución disponibles](#). Si desea instalar y usar una versión de tiempo de ejecución personalizada en lugar de las versiones de tiempo de ejecución preinstaladas, consulte [Versiones de tiempo de ejecución personalizadas](#).

Versiones de motores en tiempo de ejecución de las plataformas Ubuntu y Amazon Linux 2

Nombre del entorno de tiempo de ejecución	Versión	Versión específica	Versión principal específica y última versión secundaria	Última versión
android	28	android: 28	android: 28.x	android: latest
	29	android: 29	android: 29.x	
dotnet	3.1	dotnet: 3.1	dotnet: 3.x	dotnet: latest
	5.0	dotnet: 5.0	dotnet: 5.x	
	6.0	dotnet: 6.0	dotnet: 6.x	
	8.0	dotnet: 8.0	dotnet: 8.x	
golang	1.12	golang: 1.12	golang: 1.x	golang: latest
	1.13	golang: 1.13		
	1.14	golang: 1.14		
	1.15	golang: 1.15		
	1.16	golang: 1.16		

Nombre del entorno de tiempo de ejecución	Versión	Versión específica	Versión principal específica y última versión secundaria	Última versión
	1.18	golang: 1.18		
	1,20	golang: 1.20		
	1.21	golang: 1.21		
	1.22	golang: 1.22		
java	corretto8	java: corretto	java: corretto .x	java: latest
	corretto11	java: corretto 1	java: corretto 1.x	
	corretto17	java: corretto 7	java: corretto 7.x	
	corretto21	java: corretto 1	java: corretto 1.x	
nodejs	10	nodejs: 10	nodejs: 10.x	nodejs: latest
	12	nodejs: 12	nodejs: 12.x	
	14	nodejs: 14	nodejs: 14.x	
	16	nodejs: 16	nodejs: 16.x	
	18	nodejs: 18	nodejs: 18.x	
	20	nodejs: 20	nodejs: 20.x	
php	7.3	php: 7.3	php: 7.x	php: latest
	7.4	php: 7.4		

Nombre del entorno de tiempo de ejecución	Versión	Versión específica	Versión principal específica y última versión secundaria	Última versión
	8.0	php: 8.0	php: 8.x	
	8.1	php: 8.1		
	8.2	php: 8.2		
	8.3	php: 8.3		
python	3.7	python: 3.7	python: 3.x	python: latest
	3.8	python: 3.8		
	3.9	python: 3.9		
	3.10	python: 3.10		
	3.11	python: 3.11		
	3.12	python: 3.12		
ruby	2.6	ruby: 2.6	ruby: 2.x	ruby: latest
	2.7	ruby: 2.7		
	3.1	ruby: 3.1	ruby: 3.x	
	3.2	ruby: 3.2		
	3.3	ruby: 3.3		

Puede usar una especificación de compilación para instalar otros componentes (por ejemplo, Apache Maven AWS CLI, Apache Ant, Mocha, RSpec o similares) durante la fase de compilación. `install`

Para obtener más información, consulte [Ejemplo de un archivo buildspec](#).

Versiones de tiempo de ejecución personalizadas

En lugar de utilizar las versiones de tiempo de ejecución preinstaladas en imágenes CodeBuild gestionadas, puede instalar y utilizar las versiones personalizadas que prefiera. La siguiente tabla muestra los tiempos de ejecución personalizados disponibles y cómo especificarlos.

Note

La selección personalizada de versiones en tiempo de ejecución solo es compatible con las imágenes de Ubuntu y Amazon Linux.

Versiones de tiempo de ejecución personalizadas

Nombre del entorno de tiempo de ejecución	Sintaxis	Ejemplo
dotnet	<i><major>.<minor>.<patch></i>	5.0.408
golang	<i><major>.<minor></i>	1.19
	<i><major>.<minor>.<patch></i>	1.19.1
java	corretto <i><major></i>	corretto15
nodejs	<i><major></i>	14
	<i><major>.<minor></i>	14.21
	<i><major>.<minor>.<patch></i>	14.21.3
php	<i><major>.<minor>.<patch></i>	8.0.30
python	<i><major></i>	3
	<i><major>.<minor></i>	3.7
	<i><major>.<minor>.<patch></i>	3.7.16
ruby	<i><major>.<minor>.<patch></i>	3.0.6

Ejemplo de buildspec en tiempo de ejecución personalizado

Este es un ejemplo de una especificación de compilación que especifica versiones de tiempo de ejecución personalizadas.

```
version: 0.2
phases:
  install:
    runtime-versions:
      java: corretto15
      php: 8.0.30
      ruby: 3.0.6
      golang: 1.19
      python: 3.7
      nodejs: 14
      dotnet: 5.0.408
```

Modos y tipos de computación del entorno de compilación

En CodeBuild, puede especificar la imagen del entorno de procesamiento y tiempo de ejecución que se CodeBuild utiliza para ejecutar las compilaciones. El término cómputo se refiere al motor informático (la CPU, la memoria y el sistema operativo) que administra y mantiene CodeBuild. Una imagen de entorno de ejecución es una imagen de contenedor que se ejecuta sobre la plataforma de procesamiento seleccionada e incluye herramientas adicionales que la compilación podría necesitar, como la AWS CLI.

Temas

- [Acerca de los modos de computación](#)
- [Acerca de los tipos de entornos](#)

Acerca de los modos de computación

CodeBuild ofrece los siguientes modos de procesamiento:

- EC2
- AWS Lambda

EC2 ofrece una flexibilidad optimizada durante la construcción y AWS Lambda ofrece velocidades de arranque optimizadas. AWS Lambda admite compilaciones más rápidas debido a una latencia de

inicio más baja. AWS Lambda también se escala automáticamente, por lo que las compilaciones no tienen que hacer cola para ejecutarse. Para obtener más información, consulte [Trabajar con AWS Lambda computación en AWS CodeBuild](#).

Acerca de los tipos de entornos

AWS CodeBuild proporciona entornos de compilación con la siguiente memoria, vCPU y espacio en disco disponibles para el modo de cómputo EC2:

Tipo de computación	Valor de computeType del entorno	Valor de tipo del entorno	Memoria	vCPU	Espacio en disco
ARM Small	BUILD_GENERAL1_SMALL	ARM_CONTAINER	4 GB	2	50 GB
ARM Large	BUILD_GENERAL1_LARGE	ARM_CONTAINER	16 GB	8	50 GB
Linux Small ¹	BUILD_GENERAL1_SMALL	LINUX_CONTAINER	3 GB	2	64 GB
Linux Medium ¹	BUILD_GENERAL1_MEDIUM	LINUX_CONTAINER	7 GB	4	128 GB
Linux Large ¹	BUILD_GENERAL1_LARGE	LINUX_CONTAINER	15 GB	8	128 GB
Linux XLarge	BUILD_GENERAL1_XLARGE	LINUX_CONTAINER	70 GB	36	256 GB

Tipo de computación	Valor de computeType del entorno	Valor de tipo del entorno	Memoria	vCPU	Espacio en disco
Linux 2XLarge	BUILD_GENERAL1_2XLARGE	LINUX_CONTAINER	145 GB	72	824 GB (SSD)
Linux GPU Small	BUILD_GENERAL1_SMALL	LINUX_GPU_CONTAINER	16 GB	4	220 GB
Linux GPU Large	BUILD_GENERAL1_LARGE	LINUX_GPU_CONTAINER	255 GB	32	50 GB
Windows Medium	BUILD_GENERAL1_MEDIUM	WINDOWS_SERVER_2019_CONTAINER	7 GB	4	128 GB
Windows Large	BUILD_GENERAL1_LARGE	WINDOWS_SERVER_2019_CONTAINER	15 GB	8	128 GB

¹ La última versión de este tipo de imagen está almacenada en la caché. Si especificas una versión más específica, CodeBuild aprovisiona esa versión en lugar de la versión en caché. Esto puede dar lugar a tiempos de desarrollo más largos. Por ejemplo, para beneficiarse del almacenamiento en caché, especifique `aws/codebuild/amazonlinux2-x86_64-standard:5.0` en lugar de una versión más detallada, como `aws/codebuild/amazonlinux2-x86_64-standard:5.0-1.0.0`.

AWS CodeBuild proporciona entornos de compilación con la siguiente memoria y espacio en disco disponibles para el modo de AWS Lambda procesamiento:

Tipo de computación	Valor de computeType del entorno	Valor de tipo del entorno	Memoria	Espacio en disco
ARM Lambda 1GB	BUILD_LAMBDA_1GB	ARM_LAMBDA_CONTAINER	1 GB	10 GB
ARM Lambda 2GB	BUILD_LAMBDA_2GB	ARM_LAMBDA_CONTAINER	2 GB	10 GB
ARM Lambda 4GB	BUILD_LAMBDA_4GB	ARM_LAMBDA_CONTAINER	4 GB	10 GB
ARM Lambda 8GB	BUILD_LAMBDA_8GB	ARM_LAMBDA_CONTAINER	8 GB	10 GB
ARM Lambda de 10 GB	BUILD_LAMBDA_10GB	ARM_LAMBDA_CONTAINER	10 GB	10 GB
Linux Lambda 1GE	BUILD_LAMBDA_1GB	LINUX_LAMBDA_CONTAINER	1 GB	10 GB
Linux Lambda 2GE	BUILD_LAMBDA_2GB	LINUX_LAMBDA_CONTAINER	2 GB	10 GB
Linux Lambda 4GE	BUILD_LAMBDA_4GB	LINUX_LAMBDA_CONTAINER	4 GB	10 GB

Tipo de computación	Valor de computeType del entorno	Valor de tipo del entorno	Memoria	Espacio en disco
Linux Lambda 8GE	BUILD_LAMBDA_8GB	LINUX_LAMBDA_CONTAINER	8 GB	10 GB
Linux Lambda 10G	BUILD_LAMBDA_10GB	LINUX_LAMBDA_CONTAINER	10 GB	10 GB

Al utilizar otros tipos de entornos, se recomienda utilizar una imagen en caché para reducir los tiempos de compilación.

El espacio de disco indicado para cada entorno de compilación solo está disponible en el directorio especificado por la variable de entorno `CODEBUILD_SRC_DIR`.

Para elegir un tipo de computación:

- En la CodeBuild consola, en el asistente Crear proyecto de compilación o en la página Editar proyecto de compilación, en Entorno, expanda Configuración adicional y, a continuación, elija una de las opciones de Tipo de procesamiento. Para obtener más información, consulte [Creación de un proyecto de compilación \(consola\)](#) o [Cambiar la configuración de un proyecto de compilación \(consola\)](#).
- Para el AWS CLI, ejecute el `update-project` comando `create-project` o, especificando el `computeType` valor del `environment` objeto. Para obtener más información, consulte [Crear un proyecto de compilación \(AWS CLI\)](#) o [Cambiar la configuración de un proyecto de compilación \(AWS CLI\)](#).
- Para los AWS SDK, llama al equivalente de la `UpdateProject` operación `CreateProject` o del lenguaje de programación de destino y especifica el equivalente del `computeType` valor del `environment` objeto. Para obtener más información, consulte [Referencia de los SDK y las herramientas de AWS](#).

Algunos tipos de entorno y computación no están disponibles en todas las regiones:

- El tipo de procesamiento Linux GPU Small (LINUX_GPU_CONTAINER) solo está disponible en las regiones siguientes:
 - Este de EE. UU. (Norte de Virginia)
 - Oeste de EE. UU. (Oregón)
 - Asia-Pacífico (Tokio)
 - Canadá (centro)
 - Europa (Fráncfort)
 - Europa (Irlanda)
 - Europa (Londres)
- El tipo de procesamiento Linux GPU Large (LINUX_GPU_CONTAINER) solo está disponible en las regiones siguientes:
 - US East (Ohio)
 - Este de EE. UU. (Norte de Virginia)
 - Oeste de EE. UU. (Oregón)
 - Asia-Pacífico (Seúl)
 - Asia-Pacífico (Singapur)
 - Asia-Pacífico (Sídney)
 - Asia-Pacífico (Tokio)
 - Canadá (centro)
 - China (Pekín)
 - China (Ningxia)
 - Europa (Fráncfort)
 - Europa (Irlanda)
 - Europa (Londres)
- El tipo de entorno ARM_CONTAINER solo está disponible en las regiones siguientes:
 - US East (Ohio)
 - Este de EE. UU. (Norte de Virginia)
 - Oeste de EE. UU. (Norte de California)
 - Oeste de EE. UU. (Oregón)
 - Asia-Pacífico (Hong Kong)
 - Asia-Pacífico (Yakarta)

- Asia-Pacífico (Hyderabad)
 - Asia-Pacífico (Bombay)
 - Asia-Pacífico (Osaka)
 - Asia-Pacífico (Seúl)
 - Asia-Pacífico (Singapur)
 - Asia-Pacífico (Sídney)
 - Asia-Pacífico (Tokio)
 - Canadá (centro)
 - China (Pekín)
 - China (Ningxia)
 - Europa (Fráncfort)
 - Europa (Irlanda)
 - Europa (Londres)
 - Europa (Milán)
 - Europa (París)
 - Europa (España)
 - Europa (Estocolmo)
 - Israel (Tel Aviv)
 - Medio Oriente (Baréin)
 - Medio Oriente (EAU)
 - América del Sur (São Paulo)
- El tipo de computación BUILD_GENERAL1_2XLARGE solo está disponible en las regiones siguientes:
- US East (Ohio)
 - Este de EE. UU. (Norte de Virginia)
 - Oeste de EE. UU. (Norte de California)
 - Oeste de EE. UU. (Oregón)
 - Asia-Pacífico (Hyderabad)
 - Asia-Pacífico (Hong Kong)
-
- Modos y tipos de computación del entorno de compilación
- Asia-Pacífico (Yakarta)

- Asia-Pacífico (Melbourne)
- Asia-Pacífico (Bombay)
- Asia-Pacífico (Seúl)
- Asia-Pacífico (Singapur)
- Asia-Pacífico (Sídney)
- Asia-Pacífico (Tokio)
- Canadá (centro)
- China (Pekín)
- China (Ningxia)
- Europa (Fráncfort)
- Europa (Irlanda)
- Europa (Londres)
- Europa (París)
- Europa (España)
- Europa (Estocolmo)
- Europa (Zúrich)
- Israel (Tel Aviv)
- Medio Oriente (Baréin)
- Medio Oriente (EAU)
- América del Sur (São Paulo)
- El modo de cómputo AWS Lambda (ARM_LAMBDA_CONTAINERyLINUX_LAMBDA_CONTAINER) solo está disponible en las siguientes regiones:
 - Este de EE. UU. (Norte de Virginia)
 - Este de EE. UU. (Ohio)
 - Oeste de EE. UU. (Oregón)
 - Asia-Pacífico (Bombay)
 - Asia-Pacífico (Singapur)
 - Asia-Pacífico (Sídney)
 - Asia-Pacífico (Tokio)

- Europa (Irlanda)
- América del Sur (São Paulo)

Para el tipo de proceso `BUILD_GENERAL1_2XLARGE`, se admiten las imágenes de Docker de hasta 100 GB sin comprimir.

Note

Para imágenes de entornos de compilación personalizados, CodeBuild admite imágenes de Docker de hasta 50 GB sin comprimir en Linux y Windows, independientemente del tipo de procesamiento. Para comprobar el tamaño de la imagen de compilación, use Docker para ejecutar el comando `docker images REPOSITORY:TAG`.

Puede utilizar Amazon EFS para acceder a más espacio en el contenedor de compilación. Para obtener más información, consulte [Ejemplo de Amazon Elastic File System para AWS CodeBuild](#). Si desea manipular el espacio en disco del contenedor durante una compilación, la compilación debe ejecutarse en modo privilegiado.

Note

De forma predeterminada, el daemon de Docker está habilitado para compilaciones que no son de VPC. Si quieres usar contenedores Docker para compilaciones de VPC, [consulta Privilegios de tiempo de ejecución y capacidades de Linux](#) en el sitio web de Docker Docs y habilita el modo privilegiado. Además, Windows no admite el modo privilegiado.

Intérpretes de comandos y comandos de los entornos de compilación

Proporcione un conjunto de comandos de AWS CodeBuild que se ejecuten en un entorno de compilación durante el ciclo de vida de una compilación (por ejemplo, la instalación de dependencias de compilación, y la comprobación y compilación del código fuente). Hay varias formas de especificar estos comandos:

- Cree un archivo de especificación de compilación e inclúyalo con su código fuente. En este archivo, especifique los comandos que desea ejecutar en cada fase del ciclo de vida de

compilación. Para obtener más información, consulte [Referencia de especificación de compilación para CodeBuild](#).

- Utilice la consola de CodeBuild para crear un proyecto de compilación. En Insert build commands (Insertar comandos de compilación), para Build commands (Comandos de compilación), especifique los comandos que desea ejecutar en la fase build. Para obtener más información, consulte [Creación de un proyecto de compilación \(consola\)](#).
- Utilice la consola de CodeBuild para cambiar la configuración de un proyecto de compilación. En Insert build commands (Insertar comandos de compilación), para Build commands (Comandos de compilación), especifique los comandos que desea ejecutar en la fase build. Para obtener más información, consulte [Cambiar la configuración de un proyecto de compilación \(consola\)](#).
- Utilice la AWS CLI o los SDK de AWS para crear un proyecto de compilación o para cambiar la configuración de un proyecto de compilación. Haga referencia al código fuente que contiene un archivo buildspec con los comandos o especifique una única cadena que incluya el contenido de un archivo buildspec equivalente. Para obtener más información, consulte [Creación de un proyecto de compilación](#) o [Cambiar la configuración de un proyecto de compilación](#).
- Utilice la AWS CLI o los SDK de AWS para comenzar una compilación, especificando un archivo buildspec o una única cadena que incluya el contenido de un archivo buildspec equivalente. Para obtener más información, consulte la descripción del valor buildspecOverride en [Ejecutar una compilación](#).

Puede especificar cualquier comando de Lenguaje de comandos de intérprete de comandos (sh). En la versión de especificación de compilación 0.1, CodeBuild ejecuta cada comando del intérprete de comandos en una instancia independiente del entorno de compilación. Esto significa que cada comando se ejecuta con independencia de los demás. Por lo tanto, de forma predeterminada, no puede ejecutar un comando que se base en el estado de comandos anteriores (por ejemplo, cambiar directorios o configurar variables de entorno). Para solventar esta limitación, le recomendamos utilizar la versión 0.2, que soluciona este problema. Si debe utilizar la versión 0.1, se recomiendan los siguientes enfoques:

- Incluya un script del intérprete de comandos en el código fuente que contenga los comandos que desea ejecutar en una única instancia del intérprete de comandos predeterminado. Por ejemplo, podría incluir un archivo denominado `my-script.sh` en su código fuente que contenga comandos como `cd MyDir; mkdir -p mySubDir; cd mySubDir; pwd;`. A continuación, en el archivo buildspec, especifique el comando `./my-script.sh`.

- En el archivo `buildspec`, o en la opción Comandos de compilación para la fase `build` únicamente, especifique un solo comando que incluya todos los comandos que desee ejecutar en una sola instancia en el intérprete de comandos predeterminado (por ejemplo, `cd MyDir && mkdir -p mySubDir && cd mySubDir && pwd`).

Si CodeBuild encuentra un error, el error podría ser más difícil de solucionar que cuando se ejecuta un solo comando en su propia instancia del intérprete de comandos predeterminado.

Los comandos que se ejecutan en una imagen de Windows Server Core usan el intérprete de comandos Powershell.

Variables de entorno en los entornos de compilación

AWS CodeBuild proporciona varias variables de entorno que puede utilizar en los comandos de compilación:

AWS_DEFAULT_REGION

La región de AWS donde se ejecuta la compilación (por ejemplo, `us-east-1`). Esta variable de entorno se utiliza principalmente en la AWS CLI.

AWS_REGION

La región de AWS donde se ejecuta la compilación (por ejemplo, `us-east-1`). Esta variable de entorno se utiliza principalmente en los SDK de AWS.

CODEBUILD_BATCH_BUILD_IDENTIFIER

Identificador de la compilación en una compilación por lotes. Se indica en la especificación de compilación del lote. Para obtener más información, consulte [the section called “Referencia de especificaciones de compilación por lotes”](#).

CODEBUILD_BUILD_ARN

Nombre de recurso de Amazon (ARN) de la compilación (por ejemplo, `arn:aws:codebuild:region-ID:account-ID:build/codebuild-demo-project:b1e6661e-e4f2-4156-9ab9-82a19EXAMPLE`).

CODEBUILD_BUILD_ID

ID de CodeBuild de la compilación (por ejemplo, `codebuild-demo-project:b1e6661e-e4f2-4156-9ab9-82a19EXAMPLE`).

CODEBUILD_BUILD_IMAGE

Identificador de la imagen de compilación de CodeBuild (por ejemplo, `aws/codebuild/standard:2.0`).

CODEBUILD_BUILD_NUMBER

Número de compilación actual del proyecto.

CODEBUILD_BUILD_SUCCEEDING

Indica si la compilación actual se ha realizado correctamente. Se establece en 0 si la compilación ha producido un error o en 1 si la compilación se ha realizado correctamente.

CODEBUILD_INITIATOR

Entidad que inició la compilación. Si CodePipeline ha iniciado la compilación, este es el nombre de la canalización (por ejemplo `codepipeline/my-demo-pipeline`). Si un usuario ha iniciado la compilación, este es el nombre del usuario (por ejemplo, `MyUserName`). Si el complemento Jenkins para CodeBuild ha iniciado la compilación, esta es la cadena `CodeBuild-Jenkins-Plugin`.

CODEBUILD_KMS_KEY_ID

Identificador de la clave de AWS KMS que CodeBuild utiliza para cifrar el artefacto de salida de la compilación (por ejemplo, `arn:aws:kms:region-ID:account-ID:key/key-ID` o `alias/key-alias`).

CODEBUILD_LOG_PATH

Nombre del flujo de registro de los registros de CloudWatch para la compilación.

CODEBUILD_PUBLIC_BUILD_URL

La URL de los resultados de compilación de esta compilación se encuentra en el sitio web de compilaciones públicas. Esta variable solo se establece si el proyecto de compilación tiene activadas las compilaciones públicas. Para obtener más información, consulte [Proyectos de compilación pública en AWS CodeBuild](#).

CODEBUILD_RESOLVED_SOURCE_VERSION

Identificador de versión del código fuente de una compilación. El contenido depende del repositorio de código fuente:

CodeCommit, GitHub, GitHub Enterprise Server y Bitbucket

Esta variable contiene el ID de confirmación.

CodePipeline

Esta variable contiene la revisión de la fuente proporcionada por CodePipeline.

Si CodePipeline no puede resolver la revisión de la fuente, por ejemplo, cuando la fuente es un bucket de Amazon S3 que no tiene el control de versiones activado, esta variable de entorno no está establecida.

Amazon S3

Esta variable no está establecida.

Cuando proceda, la variable `CODEBUILD_RESOLVED_SOURCE_VERSION` solo está disponible después de la fase `DOWNLOAD_SOURCE`.

CODEBUILD_SOURCE_REPO_URL

Dirección URL del artefacto de entrada o el repositorio de código fuente. En el caso de Amazon S3, este valor es `s3://` seguido del nombre del bucket y la ruta del artefacto de entrada. Para CodeCommit y GitHub, es la dirección URL del clon del repositorio. Si una compilación se origina en CodePipeline, esta variable de entorno puede estar vacía.

En el caso de las fuentes secundarias, la variable de entorno de la URL del repositorio de fuentes secundarias es `CODEBUILD_SOURCE_REPO_URL_<sourceIdentifier>`, donde *<sourceIdentifier>* es el identificador de fuente que se crea.

CODEBUILD_SOURCE_VERSION

El formato del valor depende del repositorio del código fuente.

- Para Amazon S3, es el ID de versión asociado con el artefacto de entrada.
- Para CodeCommit, es el ID de confirmación o el nombre de ramificación asociado con la versión del código fuente que se va a compilar.
- Para GitHub, GitHub Enterprise Server y Bitbucket es el ID de confirmación, el nombre de ramificación o el nombre de etiqueta asociado con la versión del código fuente que se va a compilar.

Note

Para una compilación de GitHub o GitHub Enterprise Server activada por un evento de solicitud de extracción de webhook, es `pr/pull-request-number`.

En el caso de las fuentes secundarias, la variable de entorno de la versión de la fuente secundaria es `CODEBUILD_SOURCE_VERSION_<sourceIdentifier>`, donde *<sourceIdentifier>* es el identificador de fuente que se crea. Para obtener más información, consulte [Ejemplo de varios orígenes de entrada y varios artefactos de salida](#).

CODEBUILD_SRC_DIR

Ruta del directorio que CodeBuild utiliza para la compilación (por ejemplo, `/tmp/src123456789/src`).

En el caso de las fuentes secundarias, la variable de entorno de la versión de la fuente secundaria es `CODEBUILD_SRC_DIR_<sourceIdentifier>`, donde *<sourceIdentifier>* es el identificador de fuente que se crea. Para obtener más información, consulte [Ejemplo de varios orígenes de entrada y varios artefactos de salida](#).

CODEBUILD_START_TIME

Hora de inicio de la compilación especificada como marca temporal de Unix en milisegundos.

CODEBUILD_WEBHOOK_ACTOR_ACCOUNT_ID

ID de la cuenta del usuario que ha desencadenado el evento de webhook.

CODEBUILD_WEBHOOK_BASE_REF

Nombre de referencia de base del evento de webhook que desencadena la compilación actual. En las solicitudes de extracción, se trata de la referencia de ramificación.

CODEBUILD_WEBHOOK_EVENT

Evento de webhook que desencadena la compilación actual.

CODEBUILD_WEBHOOK_MERGE_COMMIT

El identificador de la confirmación de fusión utilizada para la compilación. Esta variable se establece cuando se fusiona una solicitud de extracción de Bitbucket con la estrategia de agrupación y se cierra la rama de la solicitud de extracción. En este caso, la confirmación original de la solicitud de extracción deja de existir, por lo que esta variable de entorno contiene el identificador de la confirmación de fusión anulada.

CODEBUILD_WEBHOOK_PREV_COMMIT

ID de la confirmación más reciente antes del evento push del webhook que desencadena la compilación actual.

CODEBUILD_WEBHOOK_HEAD_REF

Nombre de referencia de encabezado del evento de webhook que desencadena la compilación actual. Puede ser una referencia de ramificación o de etiqueta.

CODEBUILD_WEBHOOK_TRIGGER

Muestra el evento webhook que activó la compilación. Esta variable solo está disponible para compilaciones activadas por un webhook. El valor se analiza desde la carga de contenido enviada por GitHub, GitHub Enterprise Server o Bitbucket a CodeBuild. El formato del valor depende del tipo de evento que desencadenó la compilación.

- Para las compilaciones desencadenadas por una solicitud de extracción, es `pr/pull-request-number`.
- Para las compilaciones desencadenadas por la creación de una nueva rama o la inserción de una confirmación en una rama, es `branch/branch-name`.
- Para compilaciones activadas por el envío de una etiqueta a un repositorio, es `tag/tag-name`.

HOME

Esta variable de entorno siempre está establecida en `/root`.

También puede proporcionar entornos de compilación con sus propias variables de entorno. Para obtener más información, consulte los siguientes temas:

- [Úselo CodePipeline con CodeBuild](#)
- [Creación de un proyecto de compilación](#)
- [Cambiar la configuración de un proyecto de compilación](#)
- [Ejecutar una compilación](#)
- [Referencia de la especificación de compilación](#)

Para enumerar todas las variables de entorno disponibles en un entorno de compilación, puede ejecutar el comando `printenv` (para un entorno de compilación basado en Linux) o `"Get-ChildItem Env:"` (para entornos de compilación basados en Windows) durante una compilación. A excepción de las indicadas anteriormente, las variables de entorno que comienzan por `CODEBUILD_` son para uso interno de CodeBuild. No deben utilizarse en los comandos de compilación.

⚠ Important

Se desaconseja usar variables de entorno para almacenar valores confidenciales, especialmente los identificadores de clave de acceso de AWS. Las variables de entorno se pueden mostrar en texto sin formato con herramientas como la consola de CodeBuild y la AWS CLI.

Le recomendamos que almacene los valores confidenciales en el almacén de parámetros de Amazon EC2 Systems Manager y que después los recupere de la especificación de compilación. Para almacenar valores confidenciales, consulte [Almacén de parámetros de Systems Manager](#) y [Tutorial: Crear y probar un parámetro de cadena de caracteres \(consola\)](#) en la Guía del usuario de Amazon EC2 Systems Manager. Para recuperarlos, consulte la asignación de `parameter-store` en [Sintaxis de buildspec](#).

Tareas en segundo plano en entornos de compilación

Puede ejecutar tareas en segundo plano en entornos de compilación. Para ello, en la especificación de compilación, use el comando `nohup` para ejecutar un comando como una tarea en segundo plano, aunque el proceso de compilación cierre el shell. Utilice el comando `disown` para detener una tarea que se esté ejecutando en segundo plano.

Ejemplos:

- Iniciar un proceso en segundo plano y esperar a que se complete:

```
|  
nohup sleep 30 & echo $! > pidfile  
...  
wait $(cat pidfile)
```

- Iniciar un proceso en segundo plano y no esperar a que se complete:

```
|  
nohup sleep 30 & disown $!
```

- Iniciar un proceso en segundo plano y cancelarlo más adelante:

```
|  
nohup sleep 30 & echo $! > pidfile  
...
```

```
kill $(cat pidfile)
```

Ejecutar compilaciones localmente con el agente de AWS CodeBuild

Puede utilizar el agente de AWS CodeBuild para ejecutar compilaciones de CodeBuild en un equipo local. Hay agentes disponibles para las plataformas x86_64 y ARM.

Es posible suscribirse para recibir notificaciones cuando se publican versiones nuevas del agente.

Requisitos previos

Antes de comenzar, haga lo siguiente:

- Instale Git en su equipo local.
- Instale y configure [Docker](#) en su máquina local.

Configure la imagen de compilación

Solo es necesario configurar la imagen de compilación la primera vez que se va a ejecutar el agente o cuando la imagen haya cambiado.

Para configurar la imagen de compilación

1. Si desea utilizar una imagen seleccionada de Amazon Linux 2, puede extraerla del repositorio público de Amazon ECR de CodeBuild en https://gallery.ecr.aws/codebuild/amazonlinux2-x86_64-standard con este comando:

```
$ docker pull public.ecr.aws/codebuild/amazonlinux2-x86_64-standard:4.0
```

Como alternativa, si desea utilizar otra imagen de Linux, ejecute los pasos siguientes:

- a. Clone el repositorio de imágenes de CodeBuild:

```
$ git clone https://github.com/aws/aws-codebuild-docker-images.git
```

- b. Cambie al directorio de imágenes. En este ejemplo se utiliza la imagen de `aws/codebuild/standard:5.0`.

```
$ cd aws-codebuild-docker-images/ubuntu/standard/5.0
```

- c. Compile la imagen. Esto puede tardar varios minutos.

```
$ docker build -t aws/codebuild/standard:5.0 .
```

2. Descargue el agente de CodeBuild.

Para descargar la versión x86_64 del agente, ejecute el comando siguiente:

```
$ docker pull public.ecr.aws/codebuild/local-builds:latest
```

Para descargar la versión ARMA del agente, ejecute el comando siguiente:

```
$ docker pull public.ecr.aws/codebuild/local-builds:aarch64
```

3. El agente de CodeBuild está disponible en <https://gallery.ecr.aws/codebuild/local-builds>.

La firma del algoritmo de hash seguro (SHA) de la versión x86_64 del agente es:

```
sha256:fac17c6d6c3cb500f6e7975887de1e41d29a9e70a86d6f49f76a2beacfcf967e
```

La firma SHA de la versión ARM del agente es:

```
sha256:57a5dfda63be50edce13dea16dcd5e73e8d8559029658ba08b793c9a7adc68c7
```

Puede usar el SHA para identificar la versión del agente. Para ver la firma SHA del agente, ejecute el comando siguiente y busque el SHA en RepoDigests:

```
$ docker inspect public.ecr.aws/codebuild/local-builds:latest
```

Ejecutar el agente de CodeBuild

Para ejecutar el agente de CodeBuild

1. Cambie al directorio que contiene el código fuente del proyecto de compilación.
2. Descargue el script [codebuild_build.sh](#).

```
$ curl -O https://raw.githubusercontent.com/aws/aws-codebuild-docker-images/  
master/local_builds/codebuild_build.sh  
$ chmod +x codebuild_build.sh
```

3. Ejecute el script de `codebuild_build.sh` y especifique la imagen del contenedor y el directorio de salida:

Para ejecutar una compilación de `x86_64`, utilice el comando siguiente:

```
$ ./codebuild_build.sh -i <container-image> -a <output directory>
```

Para ejecutar una compilación de `ARM`, utilice el comando siguiente:

```
$ ./codebuild_build.sh -i <container-image> -a <output directory> -l  
public.ecr.aws/codebuild/local-builds:aarch64
```

Sustituya *<container-image>* por el nombre de la imagen del contenedor, como `aws/codebuild/standard:5.0` o `public.ecr.aws/codebuild/amazonlinux2-x86_64-standard:4.0`.

El script lanza la imagen de compilación y ejecuta la compilación en el proyecto en el directorio actual. Para especificar la ubicación del proyecto de compilación, añada la opción `-s <build project directory>` al comando del script.

Recepción de notificaciones de versiones nuevas del agente de CodeBuild

Puede suscribirse a las notificaciones de Amazon SNS para saber cuándo se lanzan nuevas versiones del agente de AWS CodeBuild.

Para suscribirse a las notificaciones del agente de CodeBuild

1. Abra la consola de Amazon SNS en <https://console.aws.amazon.com/sns/v3/home>.
2. En la barra de navegación, si aún no está seleccionada, cambie la región de AWS a Este de EE. UU. (Norte de Virginia). Debe seleccionar esta región de AWS porque las notificaciones de Amazon SNS a las que se va a suscribir se crean en esa región.
3. En el panel de navegación, seleccione Subscriptions.
4. Seleccione Create subscription.

5. En Crear suscripción, haga lo siguiente:

- a. En Topic ARN, use el siguiente nombre de recurso de Amazon (ARN):

```
arn:aws:sns:us-east-1:850632864840:AWS-CodeBuild-Local-Agent-Updates
```

- b. En Protocol (Protocolo), elija Email (Correo electrónico) o SMS.
- c. En Endpoint (Punto de conexión), elija dónde desea recibir las notificaciones (por correo electrónico o SMS). Escriba un correo electrónico, una dirección o un número de teléfono, incluido el código de área.
- d. Elija Crear una suscripción.
- e. Seleccione Correo electrónico para recibir un correo electrónico en el que se le pedirá que confirme la suscripción. Siga las instrucciones del mensaje para completar la suscripción.

Si ya no desea recibir estas notificaciones, utilice el siguiente procedimiento para cancelar la suscripción.

Para cancelar la suscripción a las notificaciones del agente de CodeBuild

1. Abra la consola de Amazon SNS en <https://console.aws.amazon.com/sns/v3/home>.
2. En el panel de navegación, seleccione Subscriptions.
3. Elija la suscripción y, en Actions (Acciones), elija Delete subscriptions (Eliminar suscripciones). Cuando se le pida confirmación, elija Delete.

Úselo AWS CodeBuild con Amazon Virtual Private Cloud

Por lo general, AWS CodeBuild no puede acceder a los recursos de una VPC. Para habilitar el acceso, debe proporcionar información de configuración adicional específica de VPC en la configuración de su CodeBuild proyecto. Esta información incluye el ID de la VPC, los identificadores de subred de la VPC y los identificadores de grupo de seguridad de la VPC. Las compilaciones habilitadas para VPC pueden acceder a los recursos que se encuentran dentro de la VPC. Para obtener más información acerca de cómo configurar una VPC en Amazon VPC, consulte la [Guía del usuario de Amazon VPC](#).

Temas

- [Casos de uso](#)
- [Permitir el acceso de Amazon VPC a sus proyectos CodeBuild](#)
- [Prácticas recomendadas para las VPC](#)
- [Solución de problemas con la configuración de la VPC](#)
- [Limitaciones de las VPC](#)
- [Uso de puntos de enlace de la VPC](#)
- [Plantilla de VPC de AWS CloudFormation](#)
- [Uso de AWS CodeBuild con un servidor proxy](#)

Casos de uso

La conectividad de VPC desde las AWS CodeBuild compilaciones permite:

- Ejecutar pruebas de integración desde la compilación con los datos de una base de datos de Amazon RDS aislada en una subred privada.
- Consulta datos en un ElastiCache clúster de Amazon directamente desde las pruebas.
- Interactúe con los servicios web internos alojados en Amazon EC2, Amazon ECS o servicios que utilizan el equilibrador de carga Elastic.
- Recuperar las dependencias de repositorios de artefactos internos y autoalojados, como PyPI para Python, Maven para Java y npm para Node.js.
- Acceda a los objetos de un bucket de S3 configurado para permitir el acceso únicamente a través de un punto de conexión de Amazon VPC.

- Consultar servicios web externos que requieren direcciones IP fijas a través de la dirección IP elástica de la gateway NAT o la instancia NAT asociada a la subred.

Sus compilaciones pueden tener acceso a cualquier recurso alojado en su VPC.

Permitir el acceso de Amazon VPC a sus proyectos CodeBuild

Incluya estos ajustes en la configuración de su VPC:

- Para el ID de VPC, elija el ID de VPC que utilice. CodeBuild
- En el caso de las subredes, elija una subred privada con traducción de NAT que incluya o tenga rutas hacia los recursos que utilizan. CodeBuild
- En el caso de los grupos de seguridad, elija los grupos de seguridad que se CodeBuild utilizan para permitir el acceso a los recursos de las VPC.

Si desea utilizar la consola para crear un proyecto de compilación, consulte [Creación de un proyecto de compilación \(consola\)](#). Al crear o cambiar el CodeBuild proyecto, en la VPC, elige el ID, las subredes y los grupos de seguridad de la VPC.

Para usar el AWS CLI para crear un proyecto de compilación, consulte. [Crear un proyecto de compilación \(AWS CLI\)](#) Si utiliza la función AWS CLI with CodeBuild, la función de servicio utilizada por el usuario de IAM CodeBuild para interactuar con los servicios en nombre del usuario de IAM debe tener una política adjunta. Para obtener más información, consulte [Permitir el CodeBuild acceso a AWS los servicios necesarios para crear una interfaz de red de VPC](#).

El objeto VPCConfig debe incluir el securityGroupIdsvPCid y las subredes.

- **vpcId**: obligatorio. El ID de VPC que CodeBuild utiliza. Ejecute este comando para obtener una lista de todos los ID de VPC de Amazon de su región:

```
aws ec2 describe-vpcs
```

- **subnets**: obligatorio. Los ID de subred que incluyen los recursos utilizados por. CodeBuild Ejecute este comando para obtener estos identificadores:

```
aws ec2 describe-subnets --filters "Name=vpc-id,Values=<vpc-id>" --region us-east-1
```

Note

Reemplace la `us-east-1` por su región.

- ***securityGroupIds***: Obligatorio. Los ID de los grupos de seguridad que utilizan CodeBuild para permitir el acceso a los recursos de las VPC. Ejecute este comando para obtener estos identificadores:

```
aws ec2 describe-security-groups --filters "Name=vpc-id,Values=<vpc-id>" --region us-east-1
```

Note

Reemplace la `us-east-1` por su región.

Prácticas recomendadas para las VPC

Utilice esta lista de comprobación cuando configure una VPC con la que trabajar. CodeBuild

- Configure su VPC con subredes públicas y privadas y una puerta de enlace NAT. La puerta de enlace NAT debe estar alojada en una subred pública. Para obtener más información, consulte [VPC son subredes públicas y privadas \(NAT\)](#) en la Guía del usuario de Amazon VPC.

Important

Necesita una puerta de enlace de NAT o una instancia de NAT para utilizarla CodeBuild con su VPC de forma que CodeBuild pueda llegar a los puntos finales públicos (por ejemplo, para ejecutar comandos de CLI al ejecutar compilaciones). No puede usar la puerta de enlace de Internet en lugar de una puerta de enlace NAT o una instancia de NAT porque CodeBuild no admite la asignación de direcciones IP elásticas a las interfaces de red que crea, y Amazon EC2 no admite la asignación automática de una dirección IP pública para ninguna interfaz de red creada fuera del lanzamiento de instancias de Amazon EC2.

- Incluya varias zonas de disponibilidad con su VPC.

- Asegúrese de que sus grupos de seguridad no permiten el acceso de tráfico entrante (de entrada) a sus compilaciones. CodeBuild no tiene requisitos específicos para el tráfico saliente, pero debe permitir el acceso a todos los recursos de Internet necesarios para su compilación, como GitHub Amazon S3.

Para obtener más información, consulte [Reglas del grupo de seguridad](#) en la Guía del usuario de Amazon VPC.

- Configure subredes independientes para sus compilaciones.
- Cuando configure sus CodeBuild proyectos para acceder a su VPC, elija solo subredes privadas.

Para obtener más información acerca de cómo configurar una VPC en Amazon VPC, consulte la [Guía del usuario de Amazon VPC](#).

Para obtener más información sobre AWS CloudFormation cómo configurar una VPC para usar la función CodeBuild VPC, consulte la [Plantilla de VPC de AWS CloudFormation](#)

Solución de problemas con la configuración de la VPC

Utilice la información que aparece en el mensaje de error para ayudarle a identificar, diagnosticar y resolver los problemas.

Las siguientes son algunas pautas que le ayudarán a solucionar un error común de CodeBuild VPC: `Build does not have internet connectivity. Please check subnet network configuration`

1. [Asegúrese de que la gateway de Internet está conectada a la VPC.](#)
2. [Asegúrese de que la tabla de enrutamiento de su subred pública apunta a la gateway de Internet.](#)
3. [Asegúrese de que las ACL de red permiten el tráfico.](#)
4. [Asegúrese de que los grupos de seguridad permiten el tráfico.](#)
5. [Solucione los problemas de la gateway NAT.](#)
6. [Asegúrese de que la tabla de enrutamiento de las subredes privadas apunta a la gateway NAT.](#)
7. Asegúrese de que la función de servicio utilizada por el usuario de IAM CodeBuild para interactuar con los servicios en nombre del usuario de IAM tenga los permisos de [esta](#) política. Para obtener más información, consulte [Crear un rol de servicio de CodeBuild](#).

Si faltan permisos, CodeBuild es posible que reciba un mensaje de error que diga: Unexpected EC2 error: UnauthorizedOperation Este error puede producirse si CodeBuild no tiene los permisos de Amazon EC2 necesarios para trabajar con una VPC.

Limitaciones de las VPC

- Windows no admite la conectividad de CodeBuild VPC desde.
- La conectividad de VPC desde no CodeBuild es compatible con las VPC compartidas.

Uso de puntos de enlace de la VPC

Puede mejorar la seguridad de sus compilaciones si configura AWS CodeBuild para usar un punto de conexión de VPC de interfaz. Los puntos de conexión de la interfaz funcionan con PrivateLink, una tecnología que le permite obtener acceso de forma privada a Amazon EC2 y CodeBuild utilizando direcciones IP privadas. PrivateLink restringe todo el tráfico de red entre las instancias administradas, CodeBuild y Amazon EC2 hacia la red de Amazon. Las instancias administradas no tienen acceso a Internet. Asimismo, no necesita una gateway de Internet, ni un dispositivo NAT, ni una gateway privada virtual. No es necesario configurar PrivateLink, pero es recomendable. Para obtener más información acerca de PrivateLink y los puntos de conexión de VPC, consulte [¿Qué es AWS PrivateLink?](#).

Antes de crear los puntos de enlace de la VPC

Antes de configurar los puntos de enlace de la VPC para AWS CodeBuild, debe conocer las siguientes restricciones y limitaciones.

Note

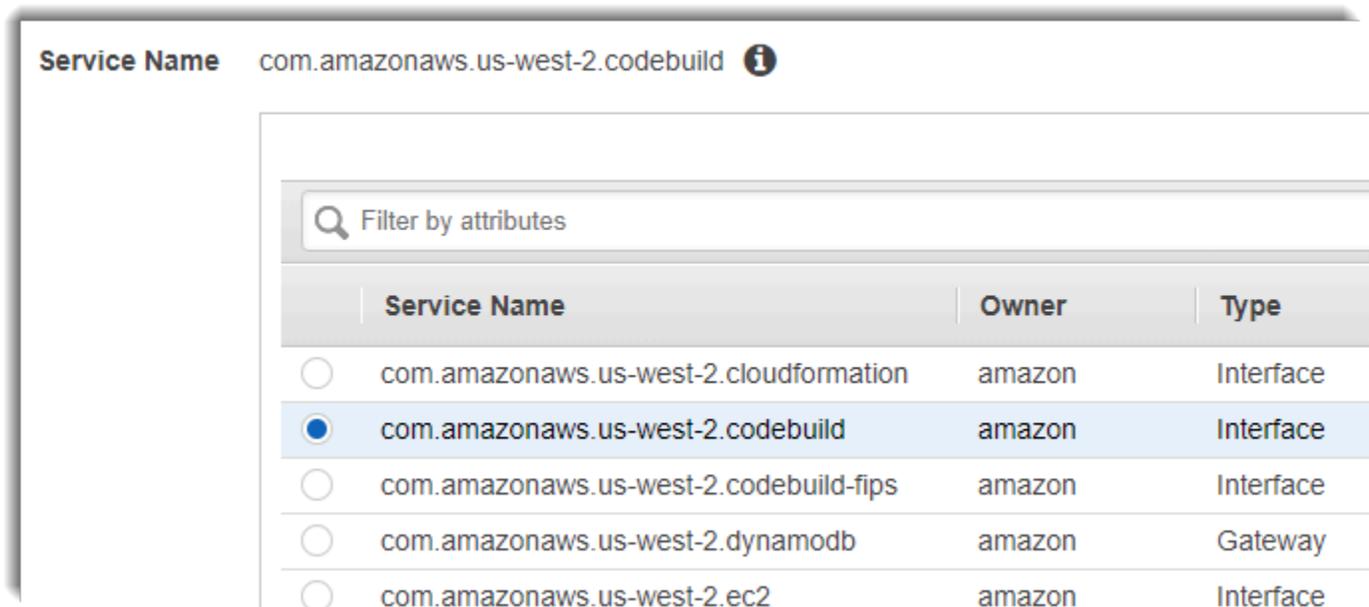
Utilice una [puerta de enlace NAT](#) si desea utilizar CodeBuild con servicios de AWS que no admitan conexiones de Amazon VPC PrivateLink.

- Los puntos de conexión de VPC solo admiten DNS proporcionadas por Amazon a través de Amazon Route 53. Si desea utilizar su propio DNS, puede utilizar el enrutamiento de DNS condicional. Para obtener más información, consulte [Conjuntos de opciones de DHCP](#) en la Guía del usuario de Amazon VPC.

- En la actualidad, los puntos de enlace de la VPC no admiten las solicitudes entre regiones. Asegúrese de crear su punto de conexión en la misma región de AWS que cualquier bucket de S3 que almacene la entrada y la salida de la compilación. Puede utilizar la consola de Amazon S3 o el comando [get-bucket-location](#) para buscar la ubicación del bucket. Utilice un punto de conexión de Amazon S3 específico de región para acceder al bucket (por ejemplo, `<bucket-name>.s3-us-west-2.amazonaws.com`). Para obtener más información sobre los puntos de conexión específicos de la región para Amazon S3, consulte [Amazon Simple Storage Service](#) en la Referencia general de Amazon Web Services. Si utiliza AWS CLI para enviar solicitudes a Amazon S3, establezca la región predeterminada en la región en la que se haya creado el bucket o utilice el parámetro `--region` en las solicitudes.

Creación de puntos de conexión de VPC para CodeBuild

Siga las instrucciones de [Creación de un punto de conexión de interfaz](#) para crear el punto de conexión `com.amazonaws.us-west-2.codebuild`. Este es un punto de conexión de la VPC para AWS CodeBuild.



region representa el identificador de región de una región de AWS compatible con CodeBuild, como `us-east-2` para la región Este de EE. UU. (Ohio). Para obtener una lista de las regiones de AWS compatibles, consulte [CodeBuild](#) en la Referencia general de AWS. El punto de conexión ya contiene la región que especificó al iniciar sesión en AWS. Si cambia la región, el punto de conexión de la VPC se actualizará en consecuencia.

Crear una política de punto de conexión de VPC para CodeBuild

Puede crear una política para los puntos de conexión VPC para AWS CodeBuild en la que puede especificar:

- La entidad de seguridad que puede realizar acciones.
- Las acciones que se pueden realizar.
- Los recursos en los que se pueden realizar acciones.

La siguiente política de ejemplo especifica que todas las entidades principales solo pueden iniciar y consultar compilaciones para el proyecto `project-name`.

```
{
  "Statement": [
    {
      "Action": [
        "codebuild:ListBuildsForProject",
        "codebuild:StartBuild",
        "codebuild:BatchGetBuilds"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:codebuild:region-ID:account-ID:project/project-name",
      "Principal": "*"
    }
  ]
}
```

Para obtener más información, consulte [Control del acceso a los servicios con puntos de conexión de VPC](#) en la guía del usuario de Amazon VPC.

Plantilla de VPC de AWS CloudFormation

AWS CloudFormation le permite crear y aprovisionar implementaciones de infraestructura de AWS de manera predecible y repetida utilizando archivos de plantilla para crear y eliminar una colección de recursos como una única unidad (una pila). Para obtener más información, consulte la [Guía del usuario de AWS CloudFormation](#).

A continuación, se muestra una plantilla de YAML de AWS CloudFormation para configurar una VPC que va a utilizar AWS CodeBuild. Este archivo también está disponible en [samples.zip](#).

Description: This template deploys a VPC, with a pair of public and private subnets spread

across two Availability Zones. It deploys an internet gateway, with a default route on the public subnets. It deploys a pair of NAT gateways (one in each AZ), and default routes for them in the private subnets.

Parameters:

EnvironmentName:

Description: An environment name that is prefixed to resource names

Type: String

VpcCIDR:

Description: Please enter the IP range (CIDR notation) for this VPC

Type: String

Default: 10.192.0.0/16

PublicSubnet1CIDR:

Description: Please enter the IP range (CIDR notation) for the public subnet in the first Availability Zone

Type: String

Default: 10.192.10.0/24

PublicSubnet2CIDR:

Description: Please enter the IP range (CIDR notation) for the public subnet in the second Availability Zone

Type: String

Default: 10.192.11.0/24

PrivateSubnet1CIDR:

Description: Please enter the IP range (CIDR notation) for the private subnet in the first Availability Zone

Type: String

Default: 10.192.20.0/24

PrivateSubnet2CIDR:

Description: Please enter the IP range (CIDR notation) for the private subnet in the second Availability Zone

Type: String

Default: 10.192.21.0/24

Resources:

VPC:

Type: AWS::EC2::VPC

Properties:

```
CidrBlock: !Ref VpcCIDR
EnableDnsSupport: true
EnableDnsHostnames: true
```

Tags:

- Key: Name
- Value: !Ref EnvironmentName

InternetGateway:

```
Type: AWS::EC2::InternetGateway
```

Properties:**Tags:**

- Key: Name
- Value: !Ref EnvironmentName

InternetGatewayAttachment:

```
Type: AWS::EC2::VPCGatewayAttachment
```

Properties:

```
InternetGatewayId: !Ref InternetGateway
VpcId: !Ref VPC
```

PublicSubnet1:

```
Type: AWS::EC2::Subnet
```

Properties:

```
VpcId: !Ref VPC
AvailabilityZone: !Select [ 0, !GetAZs '' ]
CidrBlock: !Ref PublicSubnet1CIDR
MapPublicIpOnLaunch: true
Tags:
  - Key: Name
    Value: !Sub ${EnvironmentName} Public Subnet (AZ1)
```

PublicSubnet2:

```
Type: AWS::EC2::Subnet
```

Properties:

```
VpcId: !Ref VPC
AvailabilityZone: !Select [ 1, !GetAZs '' ]
CidrBlock: !Ref PublicSubnet2CIDR
MapPublicIpOnLaunch: true
Tags:
  - Key: Name
    Value: !Sub ${EnvironmentName} Public Subnet (AZ2)
```

PrivateSubnet1:

```
Type: AWS::EC2::Subnet
Properties:
  VpcId: !Ref VPC
  AvailabilityZone: !Select [ 0, !GetAZs '' ]
  CidrBlock: !Ref PrivateSubnet1CIDR
  MapPublicIpOnLaunch: false
  Tags:
    - Key: Name
      Value: !Sub ${EnvironmentName} Private Subnet (AZ1)
```

```
PrivateSubnet2:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId: !Ref VPC
    AvailabilityZone: !Select [ 1, !GetAZs '' ]
    CidrBlock: !Ref PrivateSubnet2CIDR
    MapPublicIpOnLaunch: false
    Tags:
      - Key: Name
        Value: !Sub ${EnvironmentName} Private Subnet (AZ2)
```

```
NatGateway1EIP:
  Type: AWS::EC2::EIP
  DependsOn: InternetGatewayAttachment
  Properties:
    Domain: vpc
```

```
NatGateway2EIP:
  Type: AWS::EC2::EIP
  DependsOn: InternetGatewayAttachment
  Properties:
    Domain: vpc
```

```
NatGateway1:
  Type: AWS::EC2::NatGateway
  Properties:
    AllocationId: !GetAtt NatGateway1EIP.AllocationId
    SubnetId: !Ref PublicSubnet1
```

```
NatGateway2:
  Type: AWS::EC2::NatGateway
  Properties:
    AllocationId: !GetAtt NatGateway2EIP.AllocationId
    SubnetId: !Ref PublicSubnet2
```

```
PublicRouteTable:
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId: !Ref VPC
    Tags:
      - Key: Name
        Value: !Sub ${EnvironmentName} Public Routes

DefaultPublicRoute:
  Type: AWS::EC2::Route
  DependsOn: InternetGatewayAttachment
  Properties:
    RouteTableId: !Ref PublicRouteTable
    DestinationCidrBlock: 0.0.0.0/0
    GatewayId: !Ref InternetGateway

PublicSubnet1RouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    RouteTableId: !Ref PublicRouteTable
    SubnetId: !Ref PublicSubnet1

PublicSubnet2RouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    RouteTableId: !Ref PublicRouteTable
    SubnetId: !Ref PublicSubnet2

PrivateRouteTable1:
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId: !Ref VPC
    Tags:
      - Key: Name
        Value: !Sub ${EnvironmentName} Private Routes (AZ1)

DefaultPrivateRoute1:
  Type: AWS::EC2::Route
  Properties:
    RouteTableId: !Ref PrivateRouteTable1
    DestinationCidrBlock: 0.0.0.0/0
    NatGatewayId: !Ref NatGateway1
```

```
PrivateSubnet1RouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    RouteTableId: !Ref PrivateRouteTable1
    SubnetId: !Ref PrivateSubnet1

PrivateRouteTable2:
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId: !Ref VPC
  Tags:
    - Key: Name
      Value: !Sub ${EnvironmentName} Private Routes (AZ2)

DefaultPrivateRoute2:
  Type: AWS::EC2::Route
  Properties:
    RouteTableId: !Ref PrivateRouteTable2
    DestinationCidrBlock: 0.0.0.0/0
    NatGatewayId: !Ref NatGateway2

PrivateSubnet2RouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    RouteTableId: !Ref PrivateRouteTable2
    SubnetId: !Ref PrivateSubnet2

NoIngressSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupName: "no-ingress-sg"
    GroupDescription: "Security group with no ingress rule"
    VpcId: !Ref VPC

Outputs:
  VPC:
    Description: A reference to the created VPC
    Value: !Ref VPC

PublicSubnets:
  Description: A list of the public subnets
  Value: !Join [ ",", [ !Ref PublicSubnet1, !Ref PublicSubnet2 ] ]
```

```
PrivateSubnets:
  Description: A list of the private subnets
  Value: !Join [ ",", [ !Ref PrivateSubnet1, !Ref PrivateSubnet2 ]]

PublicSubnet1:
  Description: A reference to the public subnet in the 1st Availability Zone
  Value: !Ref PublicSubnet1

PublicSubnet2:
  Description: A reference to the public subnet in the 2nd Availability Zone
  Value: !Ref PublicSubnet2

PrivateSubnet1:
  Description: A reference to the private subnet in the 1st Availability Zone
  Value: !Ref PrivateSubnet1

PrivateSubnet2:
  Description: A reference to the private subnet in the 2nd Availability Zone
  Value: !Ref PrivateSubnet2

NoIngressSecurityGroup:
  Description: Security group with no ingress rule
  Value: !Ref NoIngressSecurityGroup
```

Uso de AWS CodeBuild con un servidor proxy

Puede utilizar AWS CodeBuild con un servidor proxy para regular el tráfico HTTP y HTTPS de entrada y salida de Internet. Para ejecutar CodeBuild con un servidor proxy, debe instalar un servidor proxy en una subred pública y CodeBuild en una subred privada de una VPC.

Existen dos casos de uso principales en los que CodeBuild se ejecuta en un servidor proxy:

- Elimina la necesidad de utilizar una gateway NAT o una instancia NAT en la VPC.
- Permite especificar las URL a las que las instancias del servidor proxy pueden obtener acceso y a las que el servidor proxy deniega el acceso.

Puede utilizar CodeBuild con dos tipos de servidores proxy. En ambos casos, el servidor proxy se ejecuta en una subred pública, mientras que CodeBuild lo hace en una subred privada.

- Proxy explícito: si utiliza un servidor proxy explícito, debe configurar las variables de entorno `NO_PROXY`, `HTTP_PROXY` y `HTTPS_PROXY` en CodeBuild en el nivel de proyecto. Para obtener

más información, consulte [Cambiar la configuración de un proyecto de compilación en AWS CodeBuild](#) y [Creación de un proyecto de compilación en AWS CodeBuild](#).

- Proxy transparente: si utiliza un servidor proxy transparente, no se necesita ninguna configuración especial.

Temas

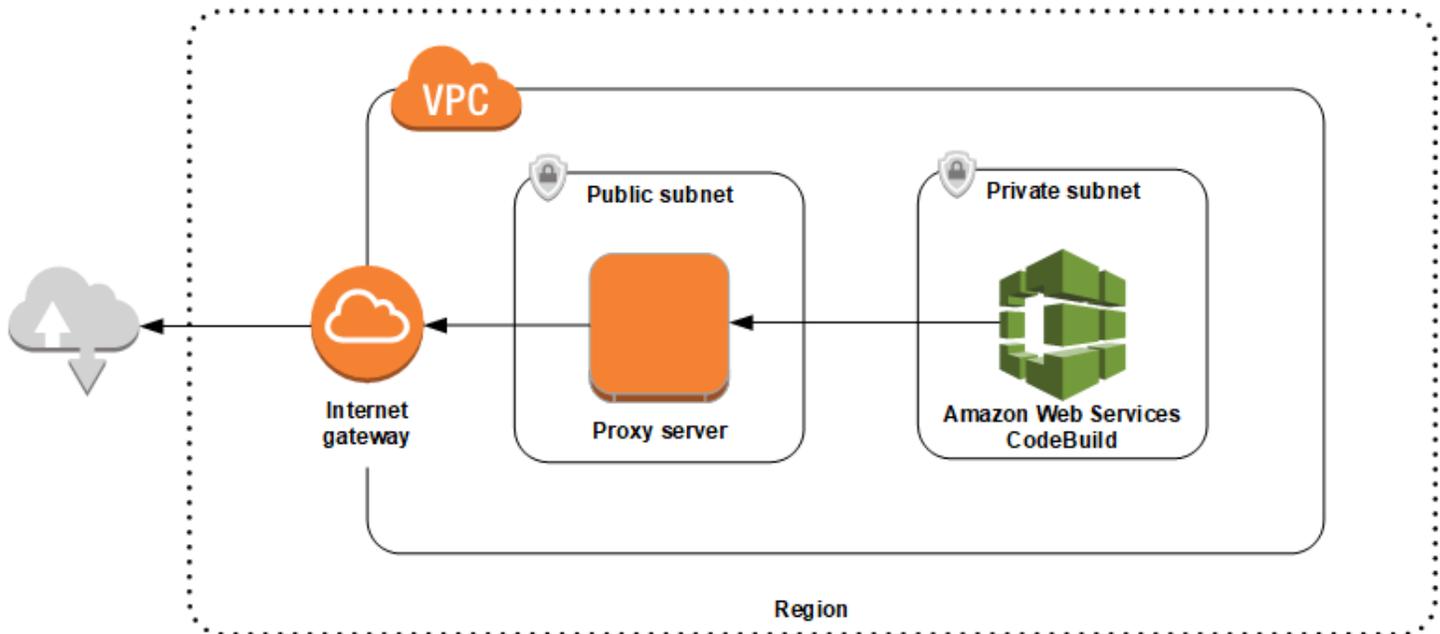
- [Componentes necesarios para ejecutar CodeBuild en un servidor proxy](#)
- [Ejecución de CodeBuild en un servidor proxy explícito](#)
- [Ejecución de CodeBuild en un servidor proxy transparente](#)
- [Ejecución de un administrador de paquetes y otras herramientas en un servidor proxy](#)

Componentes necesarios para ejecutar CodeBuild en un servidor proxy

Para ejecutar AWS CodeBuild en un servidor proxy transparente o explícito, necesita estos componentes:

- Una VPC.
- Una subred pública en la VPC para el servidor proxy.
- Una subred privada en la VPC para CodeBuild.
- Una gateway de Internet que permita la comunicación entre la VPC e Internet.

En el siguiente diagrama, se muestran cómo interactúan los componentes.



Configuración de una VPC, varias subredes y una gateway de red

A continuación, se presentan los pasos necesarios para ejecutar AWS CodeBuild en un servidor proxy transparente o explícito.

1. Cree una VPC. Para obtener más información, consulte [Creación de una VPC](#) en la Guía del usuario de Amazon VPC.
2. Cree dos subredes en la VPC. Una será una subred pública llamada `Public Subnet` en la que se ejecutará el servidor proxy. La otra será una subred privada llamada `Private Subnet` en la que se ejecutará CodeBuild.

Para obtener más información, consulte [Creación de una subred en la VPC](#).

3. Cree una gateway de Internet y vincúlela a su VPC. Para obtener más información, consulte [Creación y asociación de una gateway de Internet](#).
4. Agregue una regla a la tabla de enrutamiento predeterminada que dirija el tráfico saliente desde la VPC (0.0.0.0/0) a la gateway de Internet. Para obtener más información, consulte [Adición y eliminación de rutas de una tabla de enrutamiento](#).
5. Agregue una regla al grupo de seguridad predeterminado de la VPC que permita la entrada de tráfico SSH (TCP 22) procedente de la VPC (0.0.0.0/0).
6. Siga las instrucciones que se indican en [Lanzar una instancia con el asistente de inicialización de instancias](#) en la Guía del usuario de Amazon EC2 para lanzar una instancia de Amazon Linux. Cuando el asistente esté en ejecución, seleccione las siguientes opciones:

- En Elija un tipo de instancia, seleccione una Imagen de máquina de Amazon (AMI) de Amazon Linux.
- En Subnet (Subred), seleccione la subred pública que creó anteriormente en este tema. Si utilizó el nombre recomendado, se llamará Public Subnet (Subred pública).
- En Auto-assign Public IP (IP de asignación automática), elija Enable (Habilitar).
- En la página Configure Security Group (Configurar grupo de seguridad), en Assign a security group (Asignar un grupo de seguridad), seleccione Select an existing security group (Seleccionar un grupo de seguridad existente). A continuación, seleccione el grupo de seguridad predeterminado.
- Cuando haga clic en Launch (Lanzar), seleccione un par de claves existente o cree uno.

Seleccione la configuración predeterminada de todas las demás opciones.

7. Una vez que la instancia EC2 esté en ejecución, desactive las comprobaciones de origen/destino. Para obtener más información, consulte [Desactivación de comprobaciones de origen/destino](#) en la Guía del usuario de Amazon VPC.
8. Cree una tabla de enrutamiento en la VPC. Agregue una regla a la tabla de enrutamiento que direcciona el tráfico destinado a Internet al servidor proxy. Asocie esta tabla de enrutamiento con la subred privada. Esta operación es necesaria para que las solicitudes de salida de las instancias de la subred privada en la que se ejecuta CodeBuild se direccionen siempre a través del servidor proxy.

Instalación y configuración de un servidor proxy

Existen muchos servidores proxy entre los que elegir. Aquí, vamos a utilizar un servidor proxy de código abierto, Squid, para mostrar cómo se ejecuta AWS CodeBuild en un servidor proxy. Puede emplear estos mismos conceptos con otros servidores proxy.

Para instalar Squid, utilice un repositorio yum ejecutando los siguientes comandos:

```
sudo yum update -y
sudo yum install -y squid
```

Después de instalar Squid, edite el archivo `squid.conf` siguiendo las instrucciones que se indican más adelante en este tema.

Configuración de Squid para el tráfico HTTPS

En HTTPS, el tráfico HTTP se encapsula en una conexión Transport Layer Security (TLS). Squid utiliza una característica llamada [SslPeekAndSplice](#) para recuperar la indicación de nombre de servidor (SNI) del inicio de TLS que contiene el host de Internet solicitado. Esta operación es necesaria, ya que Squid no necesita descifrar el tráfico HTTPS. Para habilitar SslPeekAndSplice, Squid necesita un certificado. Cree este certificado con OpenSSL:

```
sudo mkdir /etc/squid/ssl
cd /etc/squid/ssl
sudo openssl genrsa -out squid.key 2048
sudo openssl req -new -key squid.key -out squid.csr -subj "/C=XX/ST=XX/L=squid/O=squid/CN=squid"
sudo openssl x509 -req -days 3650 -in squid.csr -signkey squid.key -out squid.crt
sudo cat squid.key squid.crt | sudo tee squid.pem
```

Note

En el caso de HTTP, Squid no requiere configuración. En todos los mensajes de solicitud HTTP/1.1, puede recuperar el campo del encabezado de host, donde se especifica el host de Internet que se está solicitando.

Ejecución de CodeBuild en un servidor proxy explícito

Temas

- [Configuración de Squid como un servidor proxy explícito](#)
- [Creación de un proyecto de CodeBuild.](#)
- [Archivo squid.conf de ejemplo de un servidor proxy explícito](#)

Para ejecutar AWS CodeBuild en un servidor proxy explícito, debe configurar el servidor proxy de forma que permita o deniegue el tráfico dirigido a sitios externos o el tráfico procedente de estos sitios y debe configurar las variables de entorno HTTPS_PROXY y HTTP_PROXY.

Configuración de Squid como un servidor proxy explícito

Si desea configurar el servidor proxy de Squid para que sea explícito, debe realizar las siguientes modificaciones en el archivo `/etc/squid/squid.conf`:

- Elimine las siguientes reglas de lista de control de acceso (ACL) predeterminadas.

```
acl localnet src 10.0.0.0/8
acl localnet src 172.16.0.0/12
acl localnet src 192.168.0.0/16
acl localnet src fc00::/7
acl localnet src fe80::/10
```

Agregue lo siguiente en lugar de las reglas de ACL predeterminadas que ha eliminado. La primera línea admite las solicitudes procedentes de la VPC. Las siguientes dos líneas conceden al servidor proxy acceso a una URL de destino que AWS CodeBuild podría estar utilizando. Modifique la expresión regular de la última línea para especificar buckets de S3 o un repositorio de CodeDommit en una región de AWS. Por ejemplo:

- Si la fuente es Amazon S3, use el comando `acl download_src dstdom_regex .*s3\.us-west-1\.amazonaws\.com` para conceder acceso a los buckets de S3 de la región `us-west-1`.
- Si la fuente es AWS CodeCommit, use `git-codecommit.<your-region>.amazonaws.com` para agregar una región de AWS a una lista de elementos permitidos.

```
acl localnet src 10.1.0.0/16 #Only allow requests from within the VPC
acl allowed_sites dstdomain .github.com #Allows to download source from GitHub
acl allowed_sites dstdomain .bitbucket.com #Allows to download source from Bitbucket
acl download_src dstdom_regex .*\.amazonaws\.com #Allows to download source from
Amazon S3 or CodeCommit
```

- Reemplace `http_access allow localnet` por lo siguiente:

```
http_access allow localnet allowed_sites
http_access allow localnet download_src
```

- Si desea que la compilación cargue registros y artefactos, realice una de las siguientes acciones:
 1. Antes de la instrucción `http_access deny all`, inserte las siguientes instrucciones. Permiten a CodeBuild acceder a CloudWatch y Amazon S3. Se requiere acceso a CloudWatch para que CodeBuild pueda crear registros de CloudWatch. El acceso a Amazon S3 es necesario para poder cargar artefactos y el almacenamiento en caché de Amazon S3.

```
https_port 3130 cert=/etc/squid/ssl/squid.pem ssl-bump intercept
acl SSL_port port 443
http_access allow SSL_port
acl allowed_https_sites ssl::server_name .amazonaws.com
acl step1 at_step SslBump1
```

```
acl step2 at_step SslBump2
acl step3 at_step SslBump3
ssl_bump peek step1 all
ssl_bump peek step2 allowed_https_sites
ssl_bump splice step3 allowed_https_sites
ssl_bump terminate step2 all
```

- Después de guardar `squid.conf`, ejecute el comando siguiente:

```
sudo iptables -t nat -A PREROUTING -p tcp --dport 443 -j REDIRECT --to-port 3130
sudo service squid restart
```

2. Agregue proxy al archivo `buildspec`. Para obtener más información, consulte [Sintaxis de buildspec](#).

```
version: 0.2
proxy:
  upload-artifacts: yes
  logs: yes
phases:
  build:
    commands:
      - command
```

Note

Si recibe un error de tiempo de espera agotado `RequestError`, consulte [RequestError error de tiempo de espera cuando se ejecuta CodeBuild en un servidor proxy](#).

Para obtener más información, consulte [Archivo `squid.conf` de ejemplo de un servidor proxy explícito](#) más adelante en este tema.

Creación de un proyecto de CodeBuild.

Para ejecutar AWS CodeBuild con el servidor proxy explícito, defina las variables de entorno `HTTPS_PROXY` y `HTTP_PROXY` con la dirección IP privada de la instancia EC2 que creó para el servidor proxy y el puerto 3128 en el nivel del proyecto. La dirección IP privada será parecida a la siguiente: `http://your-ec2-private-ip-address:3128`. Para obtener más información,

consulte [Creación de un proyecto de compilación en AWS CodeBuild](#) y [Cambiar la configuración de un proyecto de compilación en AWS CodeBuild](#).

Utilice el comando siguiente para ver el registro de acceso del proxy de Squid:

```
sudo tail -f /var/log/squid/access.log
```

Archivo **squid.conf** de ejemplo de un servidor proxy explícito

A continuación, se incluye un ejemplo de un archivo `squid.conf` configurado para un servidor proxy explícito.

```
acl localnet src 10.0.0.0/16 #Only allow requests from within the VPC
# add all URLs to be whitelisted for download source and commands to be run in build
environment
acl allowed_sites dstdomain .github.com #Allows to download source from github
acl allowed_sites dstdomain .bitbucket.com #Allows to download source from bitbucket
acl allowed_sites dstdomain ppa.launchpad.net #Allows to run apt-get in build
environment
acl download_src dstdom_regex .*\.amazonaws\.com #Allows to download source from S3
or CodeCommit
acl SSL_ports port 443
acl Safe_ports port 80 # http
acl Safe_ports port 21 # ftp
acl Safe_ports port 443 # https
acl Safe_ports port 70 # gopher
acl Safe_ports port 210 # wais
acl Safe_ports port 1025-65535 # unregistered ports
acl Safe_ports port 280 # http-mgmt
acl Safe_ports port 488 # gss-http
acl Safe_ports port 591 # filemaker
acl Safe_ports port 777 # multiling http
acl CONNECT method CONNECT
#
# Recommended minimum Access Permission configuration:
#
# Deny requests to certain unsafe ports
http_access deny !Safe_ports
# Deny CONNECT to other than secure SSL ports
http_access deny CONNECT !SSL_ports
# Only allow cachemgr access from localhost
http_access allow localhost manager
http_access deny manager
```

```
# We strongly recommend the following be uncommented to protect innocent
# web applications running on the proxy server who think the only
# one who can access services on "localhost" is a local user
#http_access deny to_localhost
#
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS
#
# Example rule allowing access from your local networks.
# Adapt localnet in the ACL section to list your (internal) IP networks
# from where browsing should be allowed
http_access allow localnet allowed_sites
http_access allow localnet download_src
http_access allow localhost
# Add this for CodeBuild to access CWL end point, caching and upload artifacts S3
bucket end point
https_port 3130 cert=/etc/squid/ssl/squid.pem ssl-bump intercept
acl SSL_port port 443
http_access allow SSL_port
acl allowed_https_sites ssl::server_name .amazonaws.com
acl step1 at_step SslBump1
acl step2 at_step SslBump2
acl step3 at_step SslBump3
ssl_bump peek step1 all
ssl_bump peek step2 allowed_https_sites
ssl_bump splice step3 allowed_https_sites
ssl_bump terminate step2 all
# And finally deny all other access to this proxy
http_access deny all
# Squid normally listens to port 3128
http_port 3128
# Uncomment and adjust the following to add a disk cache directory.
#cache_dir ufs /var/spool/squid 100 16 256
# Leave coredumps in the first cache dir
coredump_dir /var/spool/squid
#
# Add any of your own refresh_pattern entries above these.
#
refresh_pattern ^ftp: 1440 20% 10080
refresh_pattern ^gopher: 1440 0% 1440
refresh_pattern -i (/cgi-bin/|\?) 0 0% 0
refresh_pattern . 0 20% 4320
```

Ejecución de CodeBuild en un servidor proxy transparente

Para ejecutar AWS CodeBuild en un servidor proxy transparente, debe configurar el servidor proxy para que tenga acceso a los sitios web y los dominios con los que interactúa.

Configuración de Squid como un servidor proxy transparente

Si desea configurar un servidor proxy para que sea transparente, debe concederle acceso a los dominios y sitios web que desee. Para ejecutar AWS CodeBuild con un servidor proxy transparente, debe concederle acceso a `amazonaws.com`. También debe concederle acceso a otros sitios web que CodeBuild utilice. Los valores variarán en función de cómo cree los proyectos de CodeBuild. Existen sitios web de ejemplo con repositorios, como GitHub, Bitbucket, Yum y Maven. Para conceder a Squid acceso a determinados dominios y sitios web, utilice un comando similar al siguiente para actualizar el archivo `squid.conf`. Este comando de ejemplo concede acceso a `amazonaws.com`, `github.com` y `bitbucket.com`. Puede editar este ejemplo para conceder acceso a otros sitios web.

```
cat | sudo tee /etc/squid/squid.conf #EOF
visible_hostname squid
#Handling HTTP requests
http_port 3129 intercept
acl allowed_http_sites dstdomain .amazonaws.com
#acl allowed_http_sites dstdomain domain_name [uncomment this line to add another
  domain]
http_access allow allowed_http_sites
#Handling HTTPS requests
https_port 3130 cert=/etc/squid/ssl/squid.pem ssl-bump intercept
acl SSL_port port 443
http_access allow SSL_port
acl allowed_https_sites ssl::server_name .amazonaws.com
acl allowed_https_sites ssl::server_name .github.com
acl allowed_https_sites ssl::server_name .bitbucket.com
#acl allowed_https_sites ssl::server_name [uncomment this line to add another website]
acl step1 at_step SslBump1
acl step2 at_step SslBump2
acl step3 at_step SslBump3
ssl_bump peek step1 all
ssl_bump peek step2 allowed_https_sites
ssl_bump splice step3 allowed_https_sites
ssl_bump terminate step2 all
http_access deny all
```

```
EOF
```

Las solicitudes entrantes procedentes de las instancias de la subred privada deben redirigirse a los puertos de Squid. Squid escucha el tráfico HTTP en el puerto 3129 (en lugar de en el 80) y el tráfico HTTPS en el puerto 3130 (en lugar de en el 443). Utilice el comando iptables para direccionar el tráfico:

```
sudo iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to-port 3129
sudo iptables -t nat -A PREROUTING -p tcp --dport 443 -j REDIRECT --to-port 3130
sudo service iptables save
sudo service squid start
```

Creación de un proyecto de CodeBuild.

Una vez configurado el servidor proxy, podrá utilizarlo con AWS CodeBuild en una subred privada sin necesidad de configurar nada más. Todas las solicitudes HTTP y HTTPS pasan por el servidor proxy público. Utilice el comando siguiente para ver el registro de acceso del proxy de Squid:

```
sudo tail -f /var/log/squid/access.log
```

Ejecución de un administrador de paquetes y otras herramientas en un servidor proxy

Para ejecutar una herramienta, como un administrador de paquetes, en un servidor proxy

1. Agregue la herramienta a la lista de permitidos en su servidor proxy agregando instrucciones en el archivo `squid.conf`.
2. Agregue una línea al archivo `buildspec` que apunte al punto de conexión privado del servidor proxy.

En los siguientes ejemplos, se muestra cómo hacerlo con `apt-get`, `curl` y `maven`. Si utiliza una herramienta diferente, se aplicarán los mismos principios. Agréguela a una lista de permitidos en el archivo `squid.conf` y agregue un comando al archivo de especificación de compilación para que CodeBuild conozca el punto de conexión del servidor proxy.

Para ejecutar **apt-get** en un servidor proxy

1. Agregue las siguientes instrucciones en el archivo `squid.conf` para incluir `apt-get` en la lista de permitidos del servidor proxy. Las primeras tres líneas permiten que ejecutar `apt-get` en el entorno de compilación.

```
acl allowed_sites dstdomain ppa.launchpad.net # Required for apt-get to run in the
build environment
acl apt_get dstdom_regex .*\.launchpad.net # Required for CodeBuild to run apt-get
in the build environment
acl apt_get dstdom_regex .*\.ubuntu.com # Required for CodeBuild to run apt-get
in the build environment
http_access allow localnet allowed_sites
http_access allow localnet apt_get
```

2. Agregue la siguiente instrucción en el archivo `buildspec` para que los comandos `apt-get` busquen la configuración del proxy en `/etc/apt/apt.conf.d/00proxy`.

```
echo 'Acquire::http::Proxy "http://<private-ip-of-proxy-server>:3128";
Acquire::https::Proxy "http://<private-ip-of-proxy-server>:3128";
Acquire::ftp::Proxy "http://<private-ip-of-proxy-server>:3128";' > /etc/apt/
apt.conf.d/00proxy
```

Para ejecutar **curl** en un servidor proxy

1. Agregue lo siguiente al archivo `squid.conf` para incluir `curl` en la lista de permitidos del entorno de compilación.

```
acl allowed_sites dstdomain ppa.launchpad.net # Required to run apt-get in the
build environment
acl allowed_sites dstdomain google.com # Required for access to a webiste. This
example uses www.google.com.
http_access allow localnet allowed_sites
http_access allow localnet apt_get
```

2. Agregue la siguiente instrucción al archivo `buildspec`, de modo que `curl` utilice el servidor proxy privado para acceder al sitio web que agregó a `squid.conf`. En este ejemplo, el sitio web es `google.com`.

```
curl -x <private-ip-of-proxy-server>:3128 https://www.google.com
```

Para ejecutar **maven** en un servidor proxy

1. Agregue lo siguiente al archivo `squid.conf` para incluir maven en la lista de permitidos del entorno de compilación.

```
acl allowed_sites dstdomain ppa.launchpad.net # Required to run apt-get in the
build environment
acl maven dstdom_regex .*\.maven.org # Allows access to the maven repository in the
build environment
http_access allow localnet allowed_sites
http_access allow localnet maven
```

2. Agregue la siguiente instrucción al archivo `buildspec`.

```
maven clean install -DproxySet=true -DproxyHost=<private-ip-of-proxy-server> -
DproxyPort=3128
```

Trabajar con proyectos de compilación y compilaciones en AWS CodeBuild

Para empezar, siga las instrucciones de [Creación de un proyecto de compilación](#) y, a continuación, siga los pasos de [Ejecutar una compilación](#). Para obtener más información sobre los proyectos de compilación y las compilaciones, consulte los siguientes temas.

Temas

- [Trabajar con proyectos de compilación](#)
- [Trabajar con compilaciones en AWS CodeBuild](#)

Trabajar con proyectos de compilación

Un proyecto de compilación incluye información sobre cómo ejecutar una compilación, incluido dónde obtener el código fuente, qué entorno de compilación se debe usar, qué comandos de compilación se deben ejecutar y dónde se debe almacenar el resultado de la compilación.

Puede realizar estas tareas cuando trabaje con proyectos de compilación:

Temas

- [Creación de un proyecto de compilación en AWS CodeBuild](#)
- [Creación de una regla de notificación](#)
- [Ver una lista de nombres de proyectos de compilación en AWS CodeBuild](#)
- [Ver los detalles de un proyecto de compilación en AWS CodeBuild](#)
- [Almacenamiento en caché de compilaciones de AWS CodeBuild](#)
- [Incorpora activadores AWS CodeBuild](#)
- [GitLab conexiones](#)
- [Uso de webhooks con AWS CodeBuild](#)
- [Cambiar la configuración de un proyecto de compilación en AWS CodeBuild](#)
- [Eliminar un proyecto de compilación en AWS CodeBuild](#)
- [Uso de proyectos compartidos](#)
- [Etiquetado de proyectos en AWS CodeBuild](#)

- [Batch se integra AWS CodeBuild](#)
- [GitHub Action Runner en AWS CodeBuild](#)
- [Proyectos de compilación pública en AWS CodeBuild](#)

Creación de un proyecto de compilación en AWS CodeBuild

Puede usar la consola de AWS CodeBuild, AWS CLI o los SDK de AWS para crear un proyecto de compilación.

Requisitos previos

Antes de crear un proyecto de compilación, responda las preguntas de [Planificación de una compilación](#).

Temas

- [Creación de un proyecto de compilación \(consola\)](#)
- [Crear un proyecto de compilación \(AWS CLI\)](#)
- [Crear un proyecto de compilación \(SDK de AWS\)](#)
- [Crear un proyecto de compilación \(AWS CloudFormation\)](#)

Creación de un proyecto de compilación (consola)

Abra la AWS CodeBuild consola en <https://console.aws.amazon.com/codesuite/codebuild/home>.

Si aparece una página de CodeBuild información, elija Crear proyecto de compilación. De lo contrario, en el panel de navegación, expanda Compilar, elija Proyectos de compilación y, a continuación, elija Crear proyecto de compilación.

Elija Crear el proyecto de compilación.

Rellene las secciones siguientes: Cuando haya terminado, elija Crear proyecto de compilación en la parte inferior de la página.

Secciones:

- [Configuración del proyecto](#)
- [Origen](#)
- [Entorno](#)

- [Buildspec](#)
- [Configuración de lotes](#)
- [Artefactos](#)
- [Registros](#)

Configuración del proyecto

Nombre del proyecto

Introduzca un nombre para este proyecto de compilación. Los nombres de los proyectos de compilación deben ser únicos en cada AWS cuenta.

Descripción

Introduzca una descripción opcional del proyecto de compilación para ayudar a otros usuarios a entender para qué se utiliza el proyecto.

Insignia de compilación

(Opcional) Seleccione Habilitar insignia de compilación para convertir el estado de compilación del proyecto en visible e incrustable. Para obtener más información, consulte [Ejemplo de insignias de compilación](#).

Note

La insignia de compilación no se aplica si el proveedor de código fuente es Amazon S3.

Habilitar el límite de compilaciones simultáneas

(Opcional) Si desea limitar el número de compilaciones simultáneas del proyecto, siga los pasos siguientes:

1. Seleccione Restringir el número de compilaciones simultáneas que puede iniciar este proyecto.
2. En Límite de compilación simultánea, introduzca el número máximo de compilaciones simultáneas permitidas para este proyecto. Este límite no puede ser superior al límite de compilaciones simultáneas establecido para la cuenta. Si intenta introducir un número mayor que el límite de la cuenta, aparecerá un mensaje de error.

Solo se inician compilaciones nuevas si el número actual de compilaciones es menor o igual que este límite. Si el número de compilaciones actual alcanza este límite, se regulan las compilaciones nuevas y no se ejecutan.

Información adicional

(Opcional) En el caso de las etiquetas, introduzca el nombre y el valor de las etiquetas que desee que utilicen AWS los servicios auxiliares. Utilice Agregar fila para añadir una etiqueta. Puede añadir hasta 50 etiquetas.

Origen

Proveedor de fuentes

Seleccione el tipo de proveedor de código fuente. Utilice las listas siguientes para realizar las selecciones apropiadas para el proveedor de código fuente:

Note

CodeBuild no es compatible con Bitbucket Server.

Amazon S3

Bucket

Seleccione el nombre del bucket de entrada que contiene el código fuente.

Clave de objeto de S3 o carpeta de S3

Introduzca el nombre del archivo ZIP o la ruta a la carpeta que contiene el código fuente. Introduzca una barra diagonal (/) para descargarlo todo en el bucket de S3.

Versión de fuente

Introduzca el ID de versión del objeto que representa la compilación del archivo de entrada. Para obtener más información, consulte [Ejemplo de la versión fuente con AWS CodeBuild](#).

CodeCommit

Repositorio

Seleccione el repositorio que desea utilizar.

Tipo de referencia

Seleccione Ramificación, Etiqueta de Git o ID de confirmación para especificar la versión de su código fuente. Para obtener más información, consulte [Ejemplo de la versión fuente con AWS CodeBuild](#).

Note

Recomendamos seleccionar nombres de ramificaciones de Git que no parezcan identificadores de confirmación, como 811dd1ba1aba14473856cee38308caed7190c0d o 5392f7. Esto ayuda a evitar que la retirada desde Git colisione con confirmaciones reales.

Profundidad del clon de Git

Elija crear un clon superficial con un historial truncado en el número de confirmaciones especificado. Si desea un clon completo, elija Completo.

Submódulos de Git

Seleccione Usar submódulos de Git si desea incluir submódulos de Git en el repositorio.

Bitbucket

Repositorio

Elija Conectarse mediante OAuth o Conectarse con una contraseña de la aplicación Bitbucket y siga las instrucciones para conectarse o volver a conectarse a Bitbucket.

Seleccione un repositorio público o un repositorio de su cuenta.

Versión de fuente

Introduzca una ramificación, un ID de confirmación, una etiqueta o una referencia y un ID de confirmación. Para obtener más información, consulte [Ejemplo de la versión fuente con AWS CodeBuild](#).

Note

Recomendamos seleccionar nombres de ramificaciones de Git que no parezcan identificadores de confirmación, como 811dd1ba1aba14473856cee38308caed7190c0d o 5392f7. Esto ayuda a evitar que la retirada desde Git colisione con confirmaciones reales.

Profundidad del clon de Git

Seleccione Git clone depth (Profundidad del clon de Git) para crear un clon superficial con un historial truncado en el número de confirmaciones especificado. Si desea un clon completo, elija Completo.

Submódulos de Git

Seleccione Usar submódulos de Git si desea incluir submódulos de Git en el repositorio.

Estado de la compilación

Seleccione Notificar los estados de compilación al proveedor de fuentes al inicio y al final de las compilaciones si desea que se notifique el estado del inicio y finalización de la compilación al proveedor de fuentes.

Para poder informar del estado de la compilación al proveedor de fuentes, el usuario asociado al proveedor de fuentes debe tener acceso de escritura al repositorio. Si el usuario no tiene acceso de escritura, no es posible actualizar el estado de compilación. Para obtener más información, consulte [Acceso al proveedor de fuentes](#).

En Contexto de estado, introduzca el valor que se debe utilizar en el parámetro name en el estado de confirmación de Bitbucket. Para obtener más información, consulte [compilación](#) en la documentación de la API de Bitbucket.

En URL de destino, introduzca el valor que se debe utilizar en el parámetro url en el estado de confirmación de Bitbucket. Para obtener más información, consulte [compilación](#) en la documentación de la API de Bitbucket.

El estado de una compilación activada por un webhook siempre se notifica al proveedor de origen. Para que se notifique al proveedor de fuentes el estado de una compilación iniciada desde la consola o desde una llamada a la API, es necesario seleccionar esta opción.

Si las compilaciones de un proyecto se activan mediante un webhook, es necesario enviar una nueva confirmación al repositorio para que se aplique cualquier cambio en esta opción.

En los eventos de webhook de fuente principal, selecciona Reconstruir cada vez que se inserte un cambio de código en este repositorio si quieres CodeBuild compilar el código fuente cada vez que se inserte un cambio de código en este repositorio. Para obtener más información acerca de los webhooks y los grupos de filtros, consulte [Eventos de webhooks de Bitbucket](#).

GitHub

Repositorio

Elige Conectar mediante OAuth o Conectar con un token de acceso GitHub personal y sigue las instrucciones para conectarte (o volver a conectarte) GitHub y autorizar el acceso a AWS CodeBuild

Seleccione un repositorio público o un repositorio de su cuenta.

Versión de fuente

Introduzca una ramificación, un ID de confirmación, una etiqueta o una referencia y un ID de confirmación. Para obtener más información, consulte [Ejemplo de la versión fuente con AWS CodeBuild](#).

Note

Recomendamos seleccionar nombres de ramificaciones de Git que no parezcan identificadores de confirmación, como 811dd1ba1aba14473856cee38308caed7190c0d o 5392f7. Esto ayuda a evitar que la retirada desde Git colisione con confirmaciones reales.

Profundidad del clon de Git

Seleccione Git clone depth (Profundidad del clon de Git) para crear un clon superficial con un historial truncado en el número de confirmaciones especificado. Si desea un clon completo, elija Completo.

Submódulos de Git

Seleccione Usar submódulos de Git si desea incluir submódulos de Git en el repositorio.

Estado de la compilación

Seleccione Notificar los estados de compilación al proveedor de fuentes al inicio y al final de las compilaciones si desea que se notifique el estado del inicio y finalización de la compilación al proveedor de fuentes.

Para poder informar del estado de la compilación al proveedor de fuentes, el usuario asociado al proveedor de fuentes debe tener acceso de escritura al repositorio. Si el usuario no tiene acceso de escritura, no es posible actualizar el estado de compilación. Para obtener más información, consulte [Acceso al proveedor de fuentes](#).

En el contexto de estado, introduce el valor que se va a utilizar para el `context` parámetro en el GitHub estado de confirmación. Para obtener más información, consulta Cómo [crear un estado de confirmación](#) en la guía para GitHub desarrolladores.

En la URL de destino, introduce el valor que se va a utilizar para el `target_url` parámetro en el estado de GitHub confirmación. Para obtener más información, consulta Cómo [crear un estado de confirmación](#) en la guía para GitHub desarrolladores.

El estado de una compilación activada por un webhook siempre se notifica al proveedor de origen. Para que se notifique al proveedor de fuentes el estado de una compilación iniciada desde la consola o desde una llamada a la API, es necesario seleccionar esta opción.

Si las compilaciones de un proyecto se activan mediante un webhook, es necesario enviar una nueva confirmación al repositorio para que se aplique cualquier cambio en esta opción.

En los eventos de webhook de código fuente principal, selecciona Reconstruir cada vez que se envíe un cambio de código a este repositorio si quieres CodeBuild compilar el código fuente cada vez que se inserte un cambio de código en este repositorio. Para obtener más información acerca de los webhooks y los grupos de filtros, consulte [GitHub eventos de webhook](#).

GitHub Enterprise Server

GitHub Token de acceso personal empresarial

Consulte [GitHub Ejemplo de Enterprise Server](#) para obtener información sobre cómo copiar un token de acceso personal en el portapapeles. Pegue el token en el campo de texto y seleccione Guardar token.

Note

Solo es necesario introducir y guardar el token de acceso personal una vez. CodeBuild utiliza este token en todos los proyectos futuros.

Versión de fuente

Introduzca una solicitud de recuperación, ramificación, ID de confirmación, etiqueta o referencia y un ID de confirmación. Para obtener más información, consulte [Ejemplo de la versión fuente con AWS CodeBuild](#).

Note

Recomendamos seleccionar nombres de ramificaciones de Git que no parezcan identificadores de confirmación, como 811dd1ba1aba14473856cee38308caed7190c0d o 5392f7. Esto ayuda a evitar que la retirada desde Git colisione con confirmaciones reales.

Profundidad del clon de Git

Seleccione Git clone depth (Profundidad del clon de Git) para crear un clon superficial con un historial truncado en el número de confirmaciones especificado. Si desea un clon completo, elija Completo.

Submódulos de Git

Seleccione Usar submódulos de Git si desea incluir submódulos de Git en el repositorio.

Estado de la compilación

Seleccione Notificar los estados de compilación al proveedor de fuentes al inicio y al final de las compilaciones si desea que se notifique el estado del inicio y finalización de la compilación al proveedor de fuentes.

Para poder informar del estado de la compilación al proveedor de fuentes, el usuario asociado al proveedor de fuentes debe tener acceso de escritura al repositorio. Si el usuario no tiene acceso de escritura, no es posible actualizar el estado de compilación. Para obtener más información, consulte [Acceso al proveedor de fuentes](#).

En el contexto de estado, introduzca el valor que se utilizará para el `context` parámetro en el estado de GitHub confirmación. Para obtener más información, consulta Cómo [crear un estado de confirmación](#) en la guía para GitHub desarrolladores.

En la URL de destino, introduce el valor que se va a utilizar para el `target_url` parámetro en el estado de GitHub confirmación. Para obtener más información, consulta Cómo [crear un estado de confirmación](#) en la guía para GitHub desarrolladores.

El estado de una compilación activada por un webhook siempre se notifica al proveedor de origen. Para que se notifique al proveedor de fuentes el estado de una compilación iniciada desde la consola o desde una llamada a la API, es necesario seleccionar esta opción.

Si las compilaciones de un proyecto se activan mediante un webhook, es necesario enviar una nueva confirmación al repositorio para que se aplique cualquier cambio en esta opción.

SSL no segura

Selecciona Activar SSL inseguro para ignorar las advertencias de SSL al conectarte al repositorio de proyectos de tu GitHub empresa.

En Eventos de webhook de fuente principal, selecciona Reconstruir cada vez que se inserte un cambio de código en este repositorio si quieres CodeBuild compilar el código fuente cada vez que se inserte un cambio de código en este repositorio. Para obtener más información acerca de los webhooks y los grupos de filtros, consulte [GitHub eventos de webhook](#).

GitLab

Conexión

Conecta tu GitLab cuenta mediante AWS CodeConnections y usa la conexión para asociar tu repositorio de terceros como fuente para tu proyecto de compilación.

Elige Conexión predeterminada o Conexión personalizada.

La conexión predeterminada aplica una GitLab conexión predeterminada en todos los proyectos. La conexión personalizada aplica una GitLab conexión personalizada que anula la configuración predeterminada de tu cuenta.

Conexión predeterminada

El nombre de la conexión predeterminada asociada a tu cuenta.

Si aún no ha creado una conexión con su proveedor, consulte las [Crea una conexión a GitLab \(consola\)](#) instrucciones.

Conexión personalizada

Elija el nombre de la conexión personalizada que desee utilizar.

Si aún no ha creado una conexión con su proveedor, consulte las [Crea una conexión a GitLab \(consola\)](#) instrucciones.

Repositorio

Seleccione el repositorio que desea utilizar.

Versión de fuente

Introduce un ID de solicitud de extracción, una sucursal, un ID de confirmación, una etiqueta o una referencia y un ID de confirmación. Para obtener más información, consulte [Ejemplo de la versión fuente con AWS CodeBuild](#).

Note

Recomendamos seleccionar nombres de ramificaciones de Git que no parezcan identificadores de confirmación, como 811dd1ba1aba14473856cee38308caed7190c0d o 5392f7. Esto ayuda a evitar que la retirada desde Git colisione con confirmaciones reales.

Profundidad del clon de Git

Seleccione Git clone depth (Profundidad del clon de Git) para crear un clon superficial con un historial truncado en el número de confirmaciones especificado. Si desea un clon completo, elija Completo.

Estado de la compilación

Seleccione Notificar los estados de compilación al proveedor de fuentes al inicio y al final de las compilaciones si desea que se notifique el estado del inicio y finalización de la compilación al proveedor de fuentes.

Para poder informar del estado de la compilación al proveedor de fuentes, el usuario asociado al proveedor de fuentes debe tener acceso de escritura al repositorio. Si el usuario no tiene acceso de escritura, no es posible actualizar el estado de compilación. Para obtener más información, consulte [Acceso al proveedor de fuentes](#).

GitLab Self Managed

Conexión

Conecta tu GitLab cuenta mediante AWS CodeConnections y usa la conexión para asociar tu repositorio de terceros como fuente para tu proyecto de compilación.

Elige Conexión predeterminada o Conexión personalizada.

La conexión predeterminada aplica una conexión GitLab autogestionada predeterminada en todos los proyectos. La conexión personalizada aplica una conexión GitLab autogestionada personalizada que anula la configuración predeterminada de la cuenta.

Conexión predeterminada

El nombre de la conexión predeterminada asociada a tu cuenta.

Si aún no ha creado una conexión con su proveedor, consulte [Crear una conexión para GitLab autogestionarse](#) en la Guía del usuario de la consola Developer Tools para obtener instrucciones.

Conexión personalizada

Elija el nombre de la conexión personalizada que desee utilizar.

Si aún no ha creado una conexión con su proveedor, consulte [Crear una conexión GitLab autogestionada](#) en la Guía del usuario de la consola Developer Tools para obtener instrucciones.

Repositorio

Seleccione el repositorio que desea utilizar.

Versión de fuente

Introduce un ID de solicitud de extracción, una rama, un ID de confirmación, una etiqueta o una referencia y un ID de confirmación. Para obtener más información, consulte [Ejemplo de la versión fuente con AWS CodeBuild](#).

Note

Recomendamos seleccionar nombres de ramificaciones de Git que no parezcan identificadores de confirmación, como

811dd1ba1aba14473856cee38308caed7190c0d o 5392f7. Esto ayuda a evitar que la retirada desde Git colisione con confirmaciones reales.

Profundidad del clon de Git

Seleccione Git clone depth (Profundidad del clon de Git) para crear un clon superficial con un historial truncado en el número de confirmaciones especificado. Si desea un clon completo, elija Completo.

Estado de la compilación

Seleccione Notificar los estados de compilación al proveedor de fuentes al inicio y al final de las compilaciones si desea que se notifique el estado del inicio y finalización de la compilación al proveedor de fuentes.

Para poder informar del estado de la compilación al proveedor de fuentes, el usuario asociado al proveedor de fuentes debe tener acceso de escritura al repositorio. Si el usuario no tiene acceso de escritura, no es posible actualizar el estado de compilación. Para obtener más información, consulte [Acceso al proveedor de fuentes](#).

Entorno

Modelo de aprovisionamiento

Realice una de las siguientes acciones siguientes:

- Para usar flotas bajo demanda administradas por AWS CodeBuild, elija Bajo demanda. Con flotas bajo demanda, CodeBuild proporciona computación para sus construcciones. Las máquinas se destruyen cuando finaliza la compilación. Las flotas bajo demanda están completamente administradas e incluyen capacidades de escalado automático para gestionar los picos de demanda.
- Para usar las flotas de capacidad reservada gestionadas por AWS CodeBuild, elija Capacidad reservada y, a continuación, seleccione un nombre de flota. Con las flotas de capacidad reservada, configura un conjunto de instancias dedicadas para su entorno de compilación. Estas máquinas permanecen inactivas, listas para procesar compilaciones o pruebas de forma inmediata y reducen la duración de las compilaciones. Con flotas de capacidad reservada, sus máquinas estarán siempre en funcionamiento y seguirán incurriendo en costos mientras estén aprovisionadas.

Para obtener más información, consulte [Trabajando con capacidad reservada en AWS CodeBuild](#).

Imagen del entorno

Realice una de las siguientes acciones siguientes:

- Para usar una imagen de Docker gestionada por AWS CodeBuild, elija Imagen gestionada y, a continuación, seleccione Sistema operativo, Tiempo de ejecución, Imagen y Versión de la imagen. Realice una selección en Environment type (Tipo de entorno) si está disponible.
- Para usar otra imagen de Docker, elija Custom image (Imagen personalizada). En Tipo de entorno, elija ARM, Linux, GPU de Linux o Windows. Si elige Otro registro, en URL de registro externo, introduzca el nombre y etiqueta de la imagen de Docker en Docker Hub, usando el formato *docker repository/docker image name*. Si elige Amazon ECR, utilice el repositorio de Amazon ECR y la imagen de Amazon ECR para elegir la imagen de Docker en su cuenta. AWS
- Para usar una imagen de Docker privada, elija Imagen personalizada. En Tipo de entorno, elija ARM, Linux, GPU de Linux o Windows. En Image registry (Registro de imagen), elija Other registry (Otro registro) y, a continuación, introduzca el ARN de las credenciales para su imagen de Docker privada. Es necesario crear las credenciales con Secrets Manager. Para obtener más información, consulte [AWS Secrets Manager](#) en la Guía del usuario de AWS Secrets Manager .

Note

CodeBuild reemplaza las imágenes de Docker personalizadas `ENTRYPOINT`.

Computación

Realice una de las siguientes acciones siguientes:

- Para utilizar el procesamiento de EC2, elija EC2. La computación de EC2 ofrece una flexibilidad optimizada durante las ejecuciones de acción.
- Para usar la computación Lambda, elija Lambda. La computación Lambda ofrece velocidades de inicio optimizadas para sus compilaciones. Lambda admite compilaciones más rápidas debido a una latencia de inicio más baja. Lambda también escala automáticamente, por lo que las compilaciones no tienen que esperar en cola para ejecutarse. Para obtener más información, consulte [Trabajar con AWS Lambda computación en AWS CodeBuild](#).

Rol de servicio

Realice una de las siguientes acciones siguientes:

- Si no tiene un rol de CodeBuild servicio, elija Nuevo rol de servicio. Escriba un nombre para el nuevo rol en el campo Role Name.
- Si tiene un rol CodeBuild de servicio, elija el rol de servicio existente. En ARN de rol, seleccione el rol de servicio.

Note

Cuando usas la consola para crear un proyecto de compilación, puedes crear un rol de CodeBuild servicio al mismo tiempo. De forma predeterminada, el rol funciona solamente con ese proyecto de compilación. Si utiliza la consola para asociar este rol de servicio a otro proyecto de compilación, el rol se actualizará para que funcione con el otro proyecto de compilación. Un rol de servicio puede funcionar con hasta diez proyectos de compilación.

Configuración adicional

Timeout (Tiempo de espera)

Especifique un valor, entre 5 minutos y 8 horas, tras el cual CodeBuild se detenga la compilación si no se completa. Si se dejan los campos de horas y minutos en blanco, se usará el valor predeterminado de 60 minutos.

Privilegiado

(Opcional) Seleccione Habilitar esta marca si desea crear imágenes de Docker o si desea que sus compilaciones obtengan privilegios elevados solo si planea usar este proyecto de compilación para crear imágenes de Docker. De lo contrario, todas las compilaciones asociadas que intenten interactuar con el daemon de Docker producirán un error. También debe iniciar el daemon de Docker para que las compilaciones puedan interactuar con él. Una forma de hacer esto es inicializar el daemon de Docker en la fase `install` de la especificación de compilación ejecutando los siguientes comandos de compilación. No ejecute estos comandos si ha elegido una imagen de entorno de compilación proporcionada CodeBuild con el soporte de Docker.

Note

De forma predeterminada, el daemon de Docker está habilitado para compilaciones que no son de VPC. Si quieres usar contenedores Docker para compilaciones de VPC, [consulta Privilegios de tiempo de ejecución y capacidades de Linux](#) en el sitio web de Docker Docs y habilita el modo privilegiado. Además, Windows no admite el modo privilegiado.

```
- nohup /usr/local/bin/dockerd --host=unix:///var/run/docker.sock --  
host=tcp://127.0.0.1:2375 --storage-driver=overlay2 &  
- timeout 15 sh -c "until docker info; do echo .; sleep 1; done"
```

VPC

Si quieres CodeBuild trabajar con tu VPC:

- Para la VPC, elija el ID de VPC que utilice. CodeBuild
- Para las subredes de VPC, elija las subredes que incluyen los recursos que utiliza. CodeBuild
- Para los grupos de seguridad de VPC, elija los grupos de seguridad que se CodeBuild utilizan para permitir el acceso a los recursos de las VPC.

Para obtener más información, consulte [Úselo AWS CodeBuild con Amazon Virtual Private Cloud](#).

Computación

Seleccione una de las opciones disponibles.

Variables de entorno

Introduzca el nombre y el valor y, a continuación, seleccione el tipo de cada variable de entorno que se vaya a utilizar en las compilaciones.

Note

CodeBuild establece automáticamente la variable de entorno de su AWS región. Debe definir las siguientes variables de entorno, si no las ha agregado a su archivo `buildspec.yml`:

- `AWS_ACCOUNT_ID`

- `IMAGE_REPO_NAME`
- `IMAGE_TAG`

La consola y AWS CLI los usuarios pueden ver las variables de entorno. Si no le preocupa que la variable de entorno esté visible, establezca los campos Name y Value, y después establezca Type en Plaintext.

Se recomienda almacenar una variable de entorno con un valor confidencial, como un identificador de clave de AWS acceso, una clave de acceso AWS secreta o una contraseña, como parámetro en el almacén AWS Secrets Manager de parámetros de Amazon EC2 Systems Manager o.

Si utiliza el almacén de parámetros de Amazon EC2 Systems Manager, en Tipo, elija Parámetro. En Nombre, introduzca un identificador CodeBuild para la referencia. En Valor, escriba el nombre del parámetro tal como está almacenado en el almacén de parámetros de Amazon EC2 Systems Manager. Utilizando el parámetro denominado `/CodeBuild/dockerLoginPassword` como ejemplo, en Tipo, seleccione Parámetro. En Nombre, escriba `LOGIN_PASSWORD`. En Valor, introduzca `/CodeBuild/dockerLoginPassword`.

Important

Si utiliza el almacén de parámetros de Amazon EC2 Systems Manager, le recomendamos que almacene los parámetros con nombres de parámetros que comiencen por `/CodeBuild/` (por ejemplo, `/CodeBuild/dockerLoginPassword`). Puede utilizar la CodeBuild consola para crear un parámetro en Amazon EC2 Systems Manager. Seleccione Crear parámetro y siga las instrucciones del cuadro de diálogo. (En ese cuadro de diálogo, para la clave KMS, puede especificar el ARN de una AWS KMS clave de su cuenta. Amazon EC2 Systems Manager utiliza esta clave para cifrar el valor del parámetro durante el almacenamiento y descifrarlo durante la recuperación.) Si utiliza la CodeBuild consola para crear un parámetro, la consola comienza con el nombre del parámetro a `/CodeBuild/` medida que se almacena. Para obtener más información, consulte [Almacén de parámetros de Systems Manager](#) y [Tutorial de la consola del almacén de parámetros de Systems Manager](#) en la Guía del usuario de Amazon EC2 Systems Manager.

Si el proyecto de compilación hace referencia a parámetros almacenados en el almacén de parámetros de Amazon EC2 Systems Manager, el rol de servicio del proyecto de compilación debe permitir la acción `ssm:GetParameters`. Si

seleccionaste Nueva función de servicio anteriormente, CodeBuild incluye esta acción en la función de servicio predeterminada de tu proyecto de compilación. Sin embargo, si ha seleccionado Rol de servicio existente, deberá incluir esta acción en el rol de servicio por separado.

Si el proyecto de compilación hace referencia a parámetros que se encuentran en el almacén de parámetros de Amazon EC2 Systems Manager cuyos nombres no empiezan por `/CodeBuild/` y ha seleccionado Nuevo rol de servicio, debe actualizar ese rol de servicio para permitir el acceso a nombres de parámetro que no empiecen por `/CodeBuild/`. Esto es así porque el rol de servicio permite el acceso únicamente a los nombres de parámetro que empiezan por `/CodeBuild/`.

Si se selecciona Nuevo rol de servicio, el rol de servicio incluye permisos para descifrar todos los parámetros en el espacio de nombres de `/CodeBuild/` en el almacén de parámetros de Amazon EC2 Systems Manager.

Las variables de entorno que defina reemplazan las variables de entorno existentes. Por ejemplo, si la imagen de Docker ya contiene una variable de entorno denominada `MY_VAR` con un valor de `my_value` y establece una variable de entorno denominada `MY_VAR` con un valor de `other_value`, `my_value` se reemplaza por `other_value`. Asimismo, si la imagen de Docker ya contiene una variable de entorno denominada `PATH` con un valor de `/usr/local/sbin:/usr/local/bin` y establece una variable de entorno denominada `PATH` con un valor de `$PATH:/usr/share/ant/bin,/usr/local/sbin:/usr/local/bin` se reemplaza por el valor literal `$PATH:/usr/share/ant/bin`.

No establezca variables de entorno con un nombre que empiece por `CODEBUILD_`. Este prefijo se reserva para uso interno de .

Si se define una variable de entorno con el mismo nombre en varios lugares, el valor se determina de la siguiente manera:

- El valor de la llamada a la operación de inicio de la compilación tiene la máxima prioridad.
- El valor de la definición del proyecto de compilación es el siguiente en orden de prioridad.
- El valor en la declaración de especificación de compilación es el que menos prioridad tiene.

Si se utiliza Secrets Manager, seleccione Secrets Manager en Tipo. En Nombre, introduce un identificador para CodeBuild hacer referencia. En Valor, introduzca un `reference-key`

utilizando el patrón `secret-id:json-key:version-stage:version-id`. Para obtener más información, consulte [Secrets Manager reference-key in the buildspec file](#).

Important

Si se utiliza Secrets Manager, recomendamos almacenar los secretos con nombres que comiencen por `/CodeBuild/` (por ejemplo, `/CodeBuild/dockerLoginPassword`). Para obtener más información, consulte [AWS Secrets Manager](#) en la Guía del usuario de AWS Secrets Manager .

Si el proyecto de compilación hace referencia a secretos almacenados en Secrets Manager, el rol de servicio del proyecto de compilación debe permitir la acción `secretsmanager:GetSecretValue`. Si anteriormente seleccionaste Nueva función de servicio, CodeBuild incluye esta acción en la función de servicio predeterminada de tu proyecto de compilación. Sin embargo, si ha seleccionado Rol de servicio existente, deberá incluir esta acción en el rol de servicio por separado.

Si el proyecto de compilación hace referencia a secretos almacenados en Secrets Manager cuyos nombres no empiezan por `/CodeBuild/` y se selecciona Nuevo rol de servicio, es necesario actualizar ese rol de servicio para permitir el acceso a nombres de parámetro que no empiecen por `/CodeBuild/`. Esto es así porque el rol de servicio permite el acceso únicamente a los nombres de secreto que empiezan por `/CodeBuild/`.

Si elige Nuevo rol de servicio, el rol de servicio incluye permisos para descifrar todos los secretos en el espacio de nombres de `/CodeBuild/` en Secrets Manager.

Buildspec

Especificaciones de compilación

Realice una de las siguientes acciones siguientes:

- Si el código fuente incluye un archivo de especificación de compilación, elija Use a buildspec file (Usar un archivo buildspec). De forma predeterminada, CodeBuild busca un archivo con un nombre `buildspec.yml` en el directorio raíz del código fuente. Si el archivo de especificación de compilación usa un nombre o una ubicación diferentes, introduzca su ruta desde la raíz de la fuente en el nombre de la especificación de compilación (por ejemplo, `buildspec-two.yml` o `configuration/buildspec.yml`). Si el archivo de especificación de compilación está en un bucket de S3, debe estar en la misma región de AWS que el proyecto de compilación.

Especifique el archivo de especificación de compilación mediante su ARN (por ejemplo, `arn:aws:s3:::<my-codebuild-sample2>/buildspec.yml`).

- Si el código fuente no incluye un archivo de especificación de compilación o si desea ejecutar comandos de compilación diferentes de los especificados para la fase `build` en el archivo `buildspec.yml` del directorio raíz de código fuente, elija `Insert build commands` (Insertar comandos de compilación). En `Build commands` (Comandos de compilación), escriba los comandos que desea ejecutar en la fase `build`. En caso de que haya varios comandos, separe cada comando con `&&` (por ejemplo, `mvn test && mvn package`). Para ejecutar comandos en otras fases, o si tiene una lista especialmente larga de comandos para la fase `build`, añada un archivo `buildspec.yml` al directorio raíz de código fuente, añada los comandos al archivo y, a continuación, elija `Usar el archivo buildspec.yml del directorio raíz del código fuente`.

Para obtener más información, consulte [Referencia de la especificación de compilación](#).

Configuración de lotes

Es posible ejecutar un conjunto de compilaciones como una sola operación. Para obtener más información, consulte [Batch se integra AWS CodeBuild](#).

Definir la configuración por lotes

Seleccione esta opción para permitir las compilaciones por lotes en el proyecto.

Rol de servicio por lotes

Proporciona el rol de servicio para las compilaciones por lotes.

Elija una de las opciones siguientes:

- Si no tiene un rol de servicio, elija `Nuevo rol de servicio`. En `Rol de servicio`, introduzca un nombre nuevo para el rol nuevo.
- Si tiene un rol de servicio por lotes de CodeBuild, elija `Rol de servicio existente`. En `Rol de servicio`, elija el rol de servicio.

Las compilaciones por lotes introducen un nuevo rol de seguridad en la configuración por lotes. Este nuevo rol es obligatorio, ya que CodeBuild debe poder llamar a `RetryBuild` las acciones `StartBuild``StopBuild`, y en tu nombre para ejecutar compilaciones como parte de un lote. Los clientes deberían utilizar un nuevo rol, y no el mismo rol que utilizan para su compilación, por dos motivos:

- Al asignar los permisos de `StartBuild`, `StopBuild` y `RetryBuild` en el rol de compilación, una sola compilación podría iniciar más compilaciones a través de la especificación de compilación.
- CodeBuild Las compilaciones por lotes proporcionan restricciones que restringen la cantidad de compilaciones y tipos de procesamiento que se pueden usar para las compilaciones del lote. Si la función de compilación tiene estos permisos, es posible que las propias compilaciones puedan eludir estas restricciones.

Tipos de computación permitidos para lotes

Seleccione los tipos de computación permitidos para el lote. Seleccione todos los que correspondan.

Número máximo de compilaciones permitidas en el lote

Introduzca el número máximo de compilaciones permitidas en el lote. Si un lote supera este límite, fallará.

Tiempo de espera del lote

Introduzca la cantidad máxima de tiempo que puede durar la compilación del lote.

Combinar artefactos

Seleccione Combinar todos los artefactos del lote en una sola ubicación para tener todos los artefactos del lote combinados en una sola ubicación.

Modo de informe por lotes

Seleccione el modo de informe de estados de compilación deseado para las compilaciones por lotes.

Note

Este campo solo está disponible cuando la fuente del proyecto es Bitbucket o GitHub Enterprise GitHub, y en Fuente se selecciona Informar del estado de la compilación al proveedor de la fuente cuando las compilaciones comienzan y terminan.

Compilaciones agregadas

Seleccione esta opción para que los estados de todas las compilaciones del lote se combinen en un único informe de estado.

Compilaciones individuales

Seleccione esta opción para que se notifiquen los estados de compilación de cada compilación del lote por separado.

Artefactos

Tipo

Realice una de las siguientes acciones siguientes:

- Si no desea crear ningún artefacto de salida de la compilación, elija No artifacts. Por ejemplo, puede interesar esta opción si solo va a ejecutar pruebas de compilación o si desea insertar una imagen de Docker en un repositorio de Amazon ECR.
- Para almacenar la salida de la compilación en un bucket de S3, seleccione Amazon S3 y, a continuación, haga lo siguiente:
 - Si desea utilizar su nombre de proyecto para el archivo ZIP o carpeta de salida de la compilación, deje en blanco Nombre. De lo contrario, escriba el nombre. (Si desea producir un archivo ZIP y desea que el archivo ZIP tenga una extensión de archivo, asegúrese de incluirlo detrás del nombre del archivo ZIP).
 - Seleccione Habilitar control semántico de versiones si desea que el nombre especificado en el archivo buildspec invalide cualquier nombre especificado en la consola. El nombre en un archivo buildspec se calcula en tiempo de compilación y utiliza el lenguaje de comandos Shell. Por ejemplo, puede adjuntar una fecha y una hora al nombre del artefacto para que siempre sea único. Los nombres de artefactos únicos impiden que los artefactos se sobrescriban. Para obtener más información, consulte [Sintaxis de buildspec](#).
 - En Bucket name (Nombre del bucket), seleccione el nombre del bucket de salida.
 - Si eligió Insert build commands (Insertar comandos de compilación) anteriormente en este procedimiento, en Output files (Archivos de salida), escriba las ubicaciones de los archivos de la compilación que desea incluir en el archivo ZIP o carpeta de salida de la compilación. Si hay varias ubicaciones, separe cada una de las ubicaciones con una coma (por ejemplo, `appspec.yml, target/my-app.jar`). Para obtener más información, consulte la descripción de files en [Sintaxis de buildspec](#).
 - Si no desea que los artefactos de compilación se cifren, seleccione Remove artifacts encryption (Quitar cifrado de artefactos),

Para cada conjunto de artefactos secundarios que desee:

1. Para Source identifier (Identificador de artefacto), escriba un valor que tenga como máximo 128 caracteres, únicamente de tipo alfanumérico y guiones bajos.
2. Elija Add artifact (Añadir artefacto).
3. Siga los pasos anteriores para configurar los artefactos secundarios.
4. Elija Save artifact (Guardar artefacto).

Configuración adicional

Clave de cifrado

(opcional) Realice una de las operaciones siguientes:

- Para utilizar la Clave administrada de AWS para Amazon S3 en su cuenta para cifrar los artefactos de salida de la compilación, deje Clave de cifrado en blanco. Esta es la opción predeterminada.
- Para utilizar una clave administrada por el cliente para cifrar los artefactos de salida de la compilación, en Clave de cifrado, introduzca el ARN de la clave administrada por el cliente. Utilice el formato `arn:aws:kms:region-ID:account-ID:key/key-ID`.

Tipo de caché

En Tipo de caché, seleccione una de las opciones siguientes:

- Si no desea usar una memoria caché, seleccione Sin caché.
- Si prefiere utilizar una caché de Amazon S3, seleccione Amazon S3 y haga lo siguiente:
 - En Bucket, señecopme el nombre del bucket de S3 donde se almacena la caché.
 - (Opcional) Para Prefijo de ruta de caché, escriba un prefijo de ruta de Amazon S3. El valor Prefijo de ruta de caché es similar a un nombre de directorio. Le permite almacenar la caché en el mismo directorio en un bucket.

Important

No añada una barra (/) al final del prefijo de ruta.

- Si desea utilizar una caché local, seleccione Local y elija uno o varios modos de caché local.

Note

El modo de caché de capas de Docker solo está disponible para Linux. Si lo selecciona, el proyecto deberá ejecutarse en modo con privilegios.

El uso de memoria caché ahorra mucho tiempo de compilación, ya que algunas partes reutilizables del entorno de compilación se almacenan en ella y se usan en las distintas compilaciones. Para obtener información acerca de cómo especificar una caché en el archivo de especificación de compilación, consulte [Sintaxis de buildspec](#). Para obtener más información acerca del almacenamiento en caché, consulte [Almacenamiento en caché de compilaciones de AWS CodeBuild](#).

Registros

En Registros, seleccione los registros que desea crear. Puede crear CloudWatch registros de Amazon Logs, Amazon S3 o ambos.

CloudWatch

Si quieres los CloudWatch registros de Amazon Logs:

CloudWatch registros

Seleccione CloudWatch los registros.

Nombre del grupo

Introduce el nombre de tu grupo de CloudWatch registros de Amazon Logs.

Nombre de transmisión

Introduce el nombre de la secuencia de CloudWatch registro de Amazon Logs.

S3

Si desea registros de Amazon S3:

Registros de S3

Seleccione Registros de S3.

Bucket

Seleccione el nombre del bucket de S3 de los registros.

Prefijo de ruta

Introduzca el prefijo de sus registros.

Desactive el cifrado de registros de S3

Seleccione si no desea cifrar los registros de S3.

Crear un proyecto de compilación (AWS CLI)

Para obtener más información sobre el uso del AWS CLI with CodeBuild, consulte la [Referencia de la línea de comandos](#).

Para crear un proyecto de CodeBuild compilación mediante el AWS CLI, debe crear una estructura de [proyecto](#) con formato JSON, rellenar la estructura y [create-project](#) ejecutar el comando para crear el proyecto.

Creación del archivo JSON

Cree un archivo JSON mínimo con el comando [create-project](#), mediante la opción `--generate-cli-skeleton`:

```
aws codebuild create-project --generate-cli-skeleton > <json-file>
```

Esto crea un archivo JSON con la ruta y el nombre de archivo especificados por *<json-file>*.

Rellenar el archivo JSON

Modifique los datos de JSON como se indica a continuación y guarde los resultados.

```
{
  "name": "<project-name>",
  "description": "<description>",
  "source": {
    "type": "CODECOMMIT" | "CODEPIPELINE" | "GITHUB" | "GITHUB_ENTERPRISE" | "GITLAB" |
    "GITLAB_SELF_MANAGED" | "BITBUCKET" | "S3" | "NO_SOURCE",
    "location": "<source-location>",
    "gitCloneDepth": "<git-clone-depth>",
    "buildspec": "<buildspec>",
    "InsecureSsl": "<insecure-ssl>",
    "reportBuildStatus": "<report-build-status>",
    "buildStatusConfig": {
      "context": "<context>",
```

```

    "targetUrl": "<target-url>"
  },
  "gitSubmodulesConfig": {
    "fetchSubmodules": "<fetch-submodules>"
  },
  "auth": {
    "type": "<auth-type>",
    "resource": "<auth-resource>"
  },
  "sourceIdentifier": "<source-identifier>"
},
"secondarySources": [
  {
    "type": "CODECOMMIT" | "CODEPIPELINE" | "GITHUB" | "GITHUB_ENTERPRISE" |
"GITLAB" | "GITLAB_SELF_MANAGED" | "BITBUCKET" | "S3" | "NO_SOURCE",
    "location": "<source-location>",
    "gitCloneDepth": "<git-clone-depth>",
    "buildspec": "<buildspec>",
    "InsecureSsl": "<insecure-ssl>",
    "reportBuildStatus": "<report-build-status>",
    "auth": {
      "type": "<auth-type>",
      "resource": "<auth-resource>"
    },
    "sourceIdentifier": "<source-identifier>"
  }
],
"secondarySourceVersions": [
  {
    "sourceIdentifier": "<secondary-source-identifier>",
    "sourceVersion": "<secondary-source-version>"
  }
],
"sourceVersion": "<source-version>",
"artifacts": {
  "type": "CODEPIPELINE" | "S3" | "NO_ARTIFACTS",
  "location": "<artifacts-location>",
  "path": "<artifacts-path>",
  "namespaceType": "<artifacts-namespacetype>",
  "name": "<artifacts-name>",
  "overrideArtifactName": "<override-artifact-name>",
  "packaging": "<artifacts-packaging>"
},
"secondaryArtifacts": [

```

```

{
  "type": "CODEPIPELINE" | "S3" | "NO_ARTIFACTS",
  "location": "<secondary-artifact-location>",
  "path": "<secondary-artifact-path>",
  "namespaceType": "<secondary-artifact-namespaceType>",
  "name": "<secondary-artifact-name>",
  "packaging": "<secondary-artifact-packaging>",
  "artifactIdentifier": "<secondary-artifact-identifier>"
}
],
"cache": {
  "type": "<cache-type>",
  "location": "<cache-location>",
  "mode": [
    "<cache-mode>"
  ]
},
"environment": {
  "type": "LINUX_CONTAINER" | "LINUX_GPU_CONTAINER" | "ARM_CONTAINER" |
"WINDOWS_SERVER_2019_CONTAINER" | "WINDOWS_SERVER_2022_CONTAINER",
  "image": "<image>",
  "computeType": "BUILD_GENERAL1_SMALL" | "BUILD_GENERAL1_MEDIUM" |
"BUILD_GENERAL1_LARGE" | "BUILD_GENERAL1_2XLARGE",
  "certificate": "<certificate>",
  "environmentVariables": [
    {
      "name": "<environmentVariable-name>",
      "value": "<environmentVariable-value>",
      "type": "<environmentVariable-type>"
    }
  ],
  "registryCredential": [
    {
      "credential": "<credential-arn-or-name>",
      "credentialProvider": "<credential-provider>"
    }
  ],
  "imagePullCredentialsType": "CODEBUILD" | "SERVICE_ROLE",
  "privilegedMode": "<privileged-mode>"
},
"serviceRole": "<service-role>",
"timeoutInMinutes": <timeout>,
"queuedTimeoutInMinutes": <queued-timeout>,
"encryptionKey": "<encryption-key>",

```

```
"tags": [
  {
    "key": "<tag-key>",
    "value": "<tag-value>"
  }
],
"vpcConfig": {
  "securityGroupIds": [
    "<security-group-id>"
  ],
  "subnets": [
    "<subnet-id>"
  ],
  "vpcId": "<vpc-id>"
},
"badgeEnabled": "<badge-enabled>",
"logsConfig": {
  "cloudWatchLogs": {
    "status": "<cloudwatch-logs-status>",
    "groupName": "<group-name>",
    "streamName": "<stream-name>"
  },
  "s3Logs": {
    "status": "<s3-logs-status>",
    "location": "<s3-logs-location>",
    "encryptionDisabled": "<s3-logs-encryption-disabled>"
  }
},
"fileSystemLocations": [
  {
    "type": "EFS",
    "location": "<EFS-DNS-name-1>:/<directory-path>",
    "mountPoint": "<mount-point>",
    "identifier": "<efs-identifier>",
    "mountOptions": "<efs-mount-options>"
  }
],
"buildBatchConfig": {
  "serviceRole": "<batch-service-role>",
  "combineArtifacts": <combine-artifacts>,
  "restrictions": {
    "maximumBuildsAllowed": <max-builds>,
    "computeTypesAllowed": [
      "<compute-type>"
    ]
  }
}
```

```
    ]
  },
  "timeoutInMins": <batch-timeout>,
  "batchReportMode": "REPORT_AGGREGATED_BATCH" | "REPORT_INDIVIDUAL_BUILDS"
},
"concurrentBuildLimit": <concurrent-build-limit>
}
```

Sustituya lo siguiente:

name

Obligatorio. El nombre de este proyecto de compilación. Este nombre debe ser único en todos los proyectos de compilación de tu cuenta. AWS

description

Opcional. La descripción de este proyecto de compilación.

origen

Obligatorio. Un [ProjectSource](#) objeto que contiene información sobre la configuración del código fuente de este proyecto de compilación. Después de añadir un objeto source, puede añadir hasta 12 orígenes más mediante el . Esta configuración incluye lo siguiente:

source/type

Obligatorio. El tipo de repositorio que contiene el código fuente que se va a compilar. Los valores válidos son:

- CODECOMMIT
- CODEPIPELINE
- GITHUB
- GITHUB_ENTERPRISE
- GITLAB
- GITLAB_SELF_MANAGED
- BITBUCKET
- S3

- NO_SOURCE

Si usa NO_SOURCE, la especificación de compilación no puede ser un archivo porque el proyecto no tiene un origen. En su lugar, debe usar el atributo `buildspec` para especificar una cadena con formato YAML para su especificación de compilación. Para obtener más información, consulte [Ejemplo de proyecto sin un origen](#).

source/location

Obligatorio a menos que `<source-type>` sea CODEPIPELINE. La ubicación del código fuente para el tipo de repositorio especificado.

- Para CodeCommit, la URL del clon HTTPS del repositorio que contiene el código fuente y el archivo `buildspec` (por ejemplo,). `https://git-codecommit.<region-id>.amazonaws.com/v1/repos/<repo-name>`
- Para Amazon S3, el nombre del bucket de entrada de compilación, seguido de la ruta y el nombre del archivo ZIP que contiene el código fuente y la especificación de compilación. Por ejemplo:
 - Para un archivo ZIP ubicado en la raíz del bucket de entrada: `<bucket-name>/<object-name>.zip`.
 - Para un archivo ZIP ubicado en una subcarpeta del bucket de entrada: `<bucket-name>/<subfolder-path>/<object-name>.zip`.
- Para GitHub, la URL del clon HTTPS del repositorio que contiene el código fuente y el archivo `buildspec`. La dirección URL debe contener `github.com`. Debes conectar tu AWS cuenta a tu cuenta de GitHub. Para ello, utilice la CodeBuild consola para crear un proyecto de compilación.
 - Elija `Authorize application`. (Una vez que se haya conectado a su cuenta de GitHub, no necesitará terminar de crear el proyecto de compilación. Puede cerrar la CodeBuild consola.)
- En el caso de GitHub Enterprise Server, la URL del clon HTTP o HTTPS del repositorio que contiene el código fuente y el archivo `buildspec`. También debe conectar su AWS cuenta a su cuenta de GitHub Enterprise Server. Para ello, utilice la CodeBuild consola para crear un proyecto de compilación.
 1. Cree un token de acceso personal en GitHub Enterprise Server.
 2. Copie este token en su portapapeles para poder usarlo cuando cree su CodeBuild proyecto. Para obtener más información, consulte [Crear un token de acceso personal para la línea de comandos](#) en el sitio web de GitHub ayuda.
 3. Cuando utilice la consola para crear su CodeBuild proyecto, en `Source`, para `Source provider`, elija `GitHubEnterprise`.

4. En **Personal Access Token**, pegue el token que ha copiado en el portapapeles. Elija **Save Token**. Su CodeBuild cuenta ahora está conectada a su cuenta de GitHub Enterprise Server.
- Para la GitLab versión GitLab autogestionada, la URL del clon HTTPS del repositorio que contiene el código fuente y el archivo `buildspec`. Ten en cuenta que, si lo usas GitLab, la URL debe contener `gitlab.com`. Si utilizas la opción de GitLab autogestión, no es necesario que la URL contenga `gitlab.com`. Debes conectar tu AWS cuenta a tu cuenta GitLab o GitLab a tu cuenta autogestionada. Para ello, utilice la CodeBuild consola para crear un proyecto de compilación.
 - En el panel de navegación de las herramientas para desarrolladores, selecciona **Configuración**, **Conexiones** y, a continuación, **Crear conexión**. En esta página, cree una conexión autogestionada GitLab o una conexión GitLab autogestionada y, a continuación, seleccione **Conectar a GitLab**.
 - Para Bitbucket, la URL clon HTTPS al repositorio que contiene el código fuente y el archivo `buildspec`. La dirección URL debe contener `bitbucket.org`. También debes conectar tu AWS cuenta a tu cuenta de Bitbucket. Para ello, usa la CodeBuild consola para crear un proyecto de compilación.
 1. Cuando use la consola para conectarse (o volver a conectarse) a Bitbucket, en la página **Confirm access to your account** de Bitbucket, elija **Grant access**. (Una vez que te hayas conectado a tu cuenta de Bitbucket, no necesitas terminar de crear el proyecto de compilación. Puedes cerrar la CodeBuild consola.)
 - Para AWS CodePipeline, no especifique un `location` valor `source`. CodePipeline ignora este valor porque al crear una canalización en CodePipeline, se especifica la ubicación del código fuente en la etapa de origen de la canalización.

fuelle/ gitCloneDepth

Opcional. La profundidad del historial que se va a descargar. El valor mínimo es 0. Si esta valor es 0, superior a 25 o no se facilita, se descargará el historial completo con cada proyecto de compilación. Si el tipo de origen es Amazon S3, este valor no se admite.

source/buildspec

Opcional. La definición o archivo de especificación de compilación que va a usar. Si este valor no se proporciona o se establece en una cadena vacía, el código fuente debe contener un archivo `buildspec.yml` en su directorio raíz. Si se establece este valor, puede ser una definición de especificación de compilación en línea, la ruta a un archivo de especificación de compilación alternativo en relación con el directorio raíz de la fuente principal, o la ruta hasta un bucket de S3. El depósito debe estar en la misma AWS región que el proyecto de construcción. Especifique

el archivo `buildspec` utilizando su ARN (por ejemplo, `arn:aws:s3:::<my-codebuild-sample2>/buildspec.yml`). Para obtener más información, consulte [Nombre de archivo y ubicación de almacenamiento de buildspec](#).

source/auth

No utilizar. Este objeto solo lo usa la CodeBuild consola.

fuentes/ reportBuildStatus

Especifica si se debe enviar al proveedor de código fuente el estado de inicio y finalización de una compilación. Si lo configuras con un proveedor de fuentes que no sea GitHub GitHub Enterprise Server o Bitbucket, aparecerá `invalidInputException` un.

Para poder informar del estado de la compilación al proveedor de fuentes, el usuario asociado al proveedor de fuentes debe tener acceso de escritura al repositorio. Si el usuario no tiene acceso de escritura, no es posible actualizar el estado de compilación. Para obtener más información, consulte [Acceso al proveedor de fuentes](#).

fuentes/ buildStatusConfig

Contiene información que define la forma en que el proyecto de CodeBuild compilación informa del estado de la compilación al proveedor de origen. Esta opción se utiliza solo cuando el tipo de fuente es `GITHUB`, `GITHUB_ENTERPRISE` o `BITBUCKET`.

fuentes/buildStatusConfig/contexto

Con fuentes de Bitbucket, este parámetro se utiliza para el parámetro `name` en el estado de confirmación de Bitbucket. En el GitHub caso de las fuentes, este parámetro se utiliza para el `context` parámetro en el estado de GitHub confirmación.

Por ejemplo, puedes hacer que `context` contenga el número de compilación y el activador del webhook mediante las siguientes variables de CodeBuild entorno:

```
AWS CodeBuild sample-project Build #${CODEBUILD_BUILD_NUMBER} -  
${CODEBUILD_WEBHOOK_TRIGGER}
```

El resultado es que el contexto de la compilación n.º 24 se desencadena por un evento de solicitud de extracción de webhook:

```
AWS CodeBuild sample-project Build #24 - pr/8
```

fuente//targetURL buildStatusConfig

Con fuentes de Bitbucket, este parámetro se utiliza para el parámetro `url` en el estado de confirmación de Bitbucket. En el GitHub caso de las fuentes, este parámetro se utiliza para el `target_url` parámetro en el estado de GitHub confirmación.

Por ejemplo, es posible establecer la `targetUrl` y `https://aws.amazon.com/codebuild/<path to build>` y el estado de la confirmación se vinculará a esta URL.

También puedes incluir variables de CodeBuild entorno en el `targetUrl` para añadir información adicional a la URL. Por ejemplo, para añadir la región de compilación a la URL, establezca `targetUrl` en:

```
"targetUrl": "https://aws.amazon.com/codebuild/<path to build>?region=$AWS_REGION"
```

Si la región de compilación es `us-east-2`, esto se expandirá a:

```
https://aws.amazon.com/codebuild/<path to build>?region=us-east-2
```

fuente/ gitSubmodulesConfig

Opcional. Información acerca de la configuración de submódulos de Git. Se utiliza CodeCommit únicamente GitHub con GitHub Enterprise Server y Bitbucket.

source/ fetchSubmodules gitSubmodulesConfig

Establezca `fetchSubmodules` en `true` si desea incluir submódulos de Git en el repositorio. Los submódulos de Git que se incluyan deben estar configurados como HTTPS.

fuente/ InsecureSsl

Opcional. Se utiliza únicamente con GitHub Enterprise Server. Establezca este valor `true` para ignorar las advertencias de TLS al conectarse al repositorio de proyectos de GitHub Enterprise Server. El valor predeterminado es `false`. `InsecureSsl` debe utilizarse con fines de prueba únicamente. No debe utilizarse en un entorno de producción.

source/sourcIdentifier

Identificador definido por el usuario para la fuente del proyecto. Opcional para la fuente principal. Obligatorio para las fuentes secundarias.

secondarySources

Opcional. Conjunto de [ProjectSource](#) objetos que contienen información sobre las fuentes secundarias de un proyecto de compilación. Es posible añadir hasta 12 fuentes secundarias. Los objetos `secondarySources` utilizan las mismas propiedades que el objeto `ProjectSource`. En un objeto de fuente secundario, `sourceIdentifier` es obligatorio.

secondarySourceVersions

Opcional. Una matriz de objetos [ProjectSourceVersion](#). Si se especifica `secondarySourceVersions` en el nivel de compilación, prevalece sobre esto.

sourceVersion

Opcional. Versión de la entrada de la compilación que se creará para este proyecto. Si no se especifica, se utiliza la versión más reciente. Si se especifica, debe ser una de las siguientes opciones:

- Para CodeCommit, el ID de confirmación, la rama o la etiqueta de Git que se va a utilizar.
- Para GitHub, el ID de confirmación, el ID de la solicitud de extracción, el nombre de la rama o el nombre de la etiqueta que corresponda a la versión del código fuente que quieres compilar. Si se especifica un ID de solicitud de inserción, este debe tener el formato `pr/pull-request-ID` (por ejemplo, `pr/25`). Si se especifica un nombre de ramificación, se usa el ID de confirmación HEAD de la ramificación. Si no se especifica, se usa el ID de confirmación HEAD de la ramificación personalizada.
- Para GitLab: el ID de confirmación, el ID de la solicitud de extracción, el nombre de la sucursal, el nombre de la etiqueta o la referencia y un ID de confirmación. Para obtener más información, consulte [Ejemplo de la versión fuente con AWS CodeBuild](#).
- Para Bitbucket, el ID de confirmación, el nombre de ramificación o el nombre de etiqueta correspondiente a la versión de código fuente que desea compilar. Si se especifica un nombre de ramificación, se usa el ID de confirmación HEAD de la ramificación. Si no se especifica, se usa el ID de confirmación HEAD de la ramificación personalizada.
- En Amazon S3, el ID de versión del objeto que representa el archivo ZIP de entrada de compilación que se va a utilizar.

Si se especifica `sourceVersion` en el nivel de compilación y, a continuación, esa versión prevalece sobre `sourceVersion` (en el nivel del proyecto). Para obtener más información, consulte [Ejemplo de la versión fuente con AWS CodeBuild](#).

artifacts

Obligatorio. Un [ProjectArtifacts](#) objeto que contiene información sobre la configuración de los artefactos de salida de este proyecto de compilación. Después de añadir un objeto `artifacts`, puede añadir hasta 12 artefactos más mediante el `.`. Esta configuración incluye lo siguiente:

artifacts/type

Obligatorio. El tipo de artefacto de salida de la compilación. Los valores válidos son:

- CODEPIPELINE
- NO_ARTIFACTS
- S3

artifacts/location

Solo se usa con el tipo de artefacto S3. No se usa para otros tipos de artefactos.

Nombre del bucket de salida creado o identificado en los requisitos previos.

artifacts/path

Solo se usa con el tipo de artefacto S3. No se usa para otros tipos de artefactos.

Ruta del bucket de salida para colocar el archivo ZIP o la carpeta. Si no especifica un valor para `path`, CodeBuild utiliza `namespaceType` (si se ha especificado) y `name` para determinar la ruta y el nombre de la carpeta o archivo ZIP de salida de la compilación. Por ejemplo, si especifica `MyPath` en `path` y `MyArtifact.zip` en `name`, la ruta y el nombre serán `MyPath/MyArtifact.zip`.

artefactos/ NamespaceType

Solo se usa con el tipo de artefacto S3. No se usa para otros tipos de artefactos.

Espacio de nombres y archivo ZIP o carpeta de salida de la compilación. Los valores válidos son `BUILD_ID` y `NONE`. Utilice `BUILD_ID` para insertar el ID de compilación en la ruta del archivo ZIP o carpeta de salida de la compilación. De lo contrario, utilice `NONE`. Si no especifica un valor para `namespaceType`, CodeBuild utiliza `path` (si se ha especificado) y `name` para determinar la ruta y el nombre de la carpeta o archivo ZIP de salida de la compilación. Por ejemplo, si especifica `MyPath` en `path`, `BUILD_ID` en `namespaceType` y `MyArtifact.zip` en `name`, la ruta y el nombre serán `MyPath/build-ID/MyArtifact.zip`.

artifacts/name

Solo se usa con el tipo de artefacto S3. No se usa para otros tipos de artefactos.

Nombre del archivo ZIP o la carpeta de salida de la compilación dentro de `location`. Por ejemplo, si especifica `MyPath` en `path` y `MyArtifact.zip` en `name`, la ruta y el nombre serán `MyPath/MyArtifact.zip`.

artefactos/ overrideArtifactName

Solo se usa con el tipo de artefacto de S3. No se usa para otros tipos de artefactos.

Opcional. Si se establece con `true`, el nombre especificado en el bloque `artifacts` del archivo de especificación de compilación anulará `name`. Para obtener más información, consulte [Referencia de especificación de compilación para CodeBuild](#).

artifacts/packaging

Solo se usa con el tipo de artefacto S3. No se usa para otros tipos de artefactos.

Opcional. Especifica cómo empaquetar los artefactos. Los valores permitidos son:

NONE

Cree una carpeta que contenga los artefactos de compilación. Este es el valor predeterminado.

ZIP

Cree un archivo ZIP que contenga los artefactos de compilación.

SecondaryArtifacts

Opcional. Conjunto de [ProjectArtifacts](#) objetos que contienen información sobre la configuración de los artefactos secundarios de un proyecto de construcción. Puede añadir hasta 12 artefactos secundarios. El objeto `secondaryArtifacts` usa muchos de los mismos valores que utiliza el objeto .

cache

Obligatorio. [ProjectCache](#) Objeto que contiene información sobre la configuración de caché de este proyecto de compilación. Para obtener más información, consulte [Almacenamiento en caché de compilaciones](#).

environment

Obligatorio. Un [ProjectEnvironment](#) objeto que contiene información sobre la configuración del entorno de compilación de este proyecto. Esta configuración incluye:

environment/type

Obligatorio. El tipo del entorno de compilación. Para obtener más información, consulta el [artículo](#) en la referencia CodeBuild de la API.

environment/image

Obligatorio. El identificador de imagen de Docker usado por este entorno de compilación. Normalmente, este identificador se expresa como *imagen-name:etiqueta*. Por ejemplo, en el repositorio de Docker que CodeBuild utiliza para administrar sus imágenes de Docker, esto podría ser *aws/codebuild/standard:5.0*. En Docker Hub, *maven:3.3.9-jdk-8*. En Amazon ECR, *account-id.dkr.ecr.region-id.amazonaws.com/your-Amazon-ECR-repo-name:tag*. Para obtener más información, consulte [Imágenes de Docker proporcionadas por CodeBuild](#).

environment/computeType

Obligatorio. Especifica los recursos de computación que utiliza este entorno de compilación. Para obtener más información, consulte [ComputeType](#) en CodeBuild la referencia de la API.

environment/certificate

Opcional. El ARN del bucket de Amazon S3, el prefijo de la ruta y la clave de objeto que contienen el certificado codificado en PEM. La clave de objeto puede ser únicamente el archivo *.pem* o un archivo *.zip* que contenga el certificado codificado en PEM. Por ejemplo, si el nombre del bucket de Amazon S3 es *<my-bucket>*, el prefijo de ruta es *<cert>* y el nombre de la clave de objeto es *<certificate.pem>*, los formatos aceptables para *certificate* son *<my-bucket/cert/certificate.pem>* o *arn:aws:s3:::<my-bucket/cert/certificate.pem>*.

environment/environmentVariables

Opcional. Matriz de [EnvironmentVariable](#) objetos que contiene las variables de entorno que quieres especificar para este entorno de compilación. Cada variable de entorno se expresa como un objeto que contiene un *name*, *value* y *type* de *name*, *value* y *type*.

La consola y AWS CLI los usuarios pueden ver todas las variables de entorno. Si no le preocupa que la variable de entorno esté visible, establezca `name` y `value`, y establezca `type` con `PLAINTEXT`.

Le recomendamos que almacene las variables de entorno con valores confidenciales, como un identificador de clave de AWS acceso, una clave de acceso AWS secreta o una contraseña, como parámetro en el almacén AWS Secrets Manager de parámetros de Amazon EC2 Systems Manager o. Puesto que, para ese parámetro almacenado, defina un identificador CodeBuild para la referencia.

Si utiliza el almacén de parámetros de Amazon EC2 Systems Manager, `value` defina el nombre del parámetro tal como está almacenado en el almacén de parámetros. Establezca `type` en `PARAMETER_STORE`. Utilizando un parámetro denominado `/CodeBuild/dockerLoginPassword` como ejemplo, establezca `name` con `LOGIN_PASSWORD`. Establezca `value` en `/CodeBuild/dockerLoginPassword`. Establezca `type` en `PARAMETER_STORE`.

Important

Si utiliza el almacén de parámetros de Amazon EC2 Systems Manager, le recomendamos que almacene los parámetros con nombres de parámetros que comiencen por `/CodeBuild/` (por ejemplo, `/CodeBuild/dockerLoginPassword`). Puede utilizar la CodeBuild consola para crear un parámetro en Amazon EC2 Systems Manager. Seleccione Crear parámetro y siga las instrucciones del cuadro de diálogo. (En ese cuadro de diálogo, para la clave KMS, puede especificar el ARN de una AWS KMS clave de su cuenta. Amazon EC2 Systems Manager utiliza esta clave para cifrar el valor del parámetro durante el almacenamiento y descifrarlo durante la recuperación.)

Si utiliza la CodeBuild consola para crear un parámetro, la consola comienza con el nombre del parámetro a `/CodeBuild/` medida que se va almacenando. Para obtener más información, consulte [Almacén de parámetros de Systems Manager](#) y [Tutorial de la consola del almacén de parámetros de Systems Manager](#) en la Guía del usuario de Amazon EC2 Systems Manager.

Si el proyecto de compilación hace referencia a parámetros almacenados en el almacén de parámetros de Amazon EC2 Systems Manager, el rol de servicio del proyecto de compilación debe permitir la acción `ssm:GetParameters`. Si seleccionaste Nueva función de servicio anteriormente, CodeBuild incluye esta acción en la función de servicio predeterminada de tu proyecto de compilación. Sin embargo, si ha seleccionado Rol de servicio existente, deberá incluir esta acción en el rol de servicio por separado.

Si el proyecto de compilación hace referencia a parámetros que se encuentran en el almacén de parámetros de Amazon EC2 Systems Manager cuyos nombres no empiezan por `/CodeBuild/` y ha seleccionado Nuevo rol de servicio, debe actualizar ese rol de servicio para permitir el acceso a nombres de parámetro que no empiecen por `/CodeBuild/`. Esto es así porque el rol de servicio permite el acceso únicamente a los nombres de parámetro que empiezan por `/CodeBuild/`.

Si se selecciona Nuevo rol de servicio, el rol de servicio incluye permisos para descifrar todos los parámetros en el espacio de nombres de `/CodeBuild/` en el almacén de parámetros de Amazon EC2 Systems Manager.

Las variables de entorno que defina reemplazan las variables de entorno existentes.

Por ejemplo, si la imagen de Docker ya contiene una variable de entorno denominada `MY_VAR` con un valor de `my_value` y establece una variable de entorno denominada `MY_VAR` con un valor de `other_value`, `my_value` se reemplaza por `other_value`.

Asimismo, si la imagen de Docker ya contiene una variable de entorno denominada `PATH` con un valor de `/usr/local/sbin:/usr/local/bin` y establece una variable de entorno denominada `PATH` con un valor de `$PATH:/usr/share/ant/bin`, `/usr/local/sbin:/usr/local/bin` se reemplaza por el valor literal `$PATH:/usr/share/ant/bin`.

No establezca variables de entorno con un nombre que empiece por `CODEBUILD_`. Este prefijo se reserva para uso interno de .

Si se define una variable de entorno con el mismo nombre en varios lugares, el valor se determina de la siguiente manera:

- El valor de la llamada a la operación de inicio de la compilación tiene la máxima prioridad.
- El valor de la definición del proyecto de compilación es el siguiente en orden de prioridad.
- El valor en la declaración de especificación de compilación es el que menos prioridad tiene.

Si usa Secrets Manager, para `value`, defina el nombre del parámetro tal como está almacenado en Secrets Manager. Establezca `type` en `SECRETS_MANAGER`. Utilizando un secreto denominado `/CodeBuild/dockerLoginPassword` como ejemplo, establezca `name` con `LOGIN_PASSWORD`. Establezca `value` en `/CodeBuild/dockerLoginPassword`. Establezca `type` en `SECRETS_MANAGER`.

⚠ Important

Si se utiliza Secrets Manager, recomendamos almacenar los secretos con nombres que comiencen por `/CodeBuild/` (por ejemplo, `/CodeBuild/dockerLoginPassword`). Para obtener más información, consulte [AWS Secrets Manager](#) en la Guía del usuario de AWS Secrets Manager .

Si el proyecto de compilación hace referencia a secretos almacenados Secrets Manager, el rol de servicio del proyecto de compilación debe permitir la acción `secretsmanager:GetSecretValue`. Si seleccionaste un nuevo rol de servicio anteriormente, CodeBuild incluye esta acción en el rol de servicio predeterminado para tu proyecto de construcción. Sin embargo, si ha seleccionado Rol de servicio existente, deberá incluir esta acción en el rol de servicio por separado.

Si el proyecto de compilación hace referencia a secretos almacenados en Secrets Manager cuyos nombres no empiezan por `/CodeBuild/` y se selecciona Nuevo rol de servicio, es necesario actualizar ese rol de servicio para permitir el acceso a nombres de parámetro que no empiecen por `/CodeBuild/`. Esto es así porque el rol de servicio permite el acceso únicamente a los nombres de secreto que empiezan por `/CodeBuild/`.

Si elige Nuevo rol de servicio, el rol de servicio incluye permisos para descifrar todos los secretos en el espacio de nombres de `/CodeBuild/` en Secrets Manager.

environment/registryCredential

Opcional. Un [RegistryCredential](#) objeto que especifica las credenciales que proporcionan acceso a un registro de Docker privado.

environment/registryCredential/credential

Especifica el ARN o nombre de las credenciales creadas mediante AWS Managed Services. Puede utilizar el nombre de las credenciales solo si existen en su región actual.

environment/registryCredential/credentialProvider

El único valor válido es `SECRETS_MANAGER`.

Cuando se ha establecido:

- `imagePullCredentials` se debe establecer en `SERVICE_ROLE`.
- La imagen no puede ser una imagen seleccionada ni una imagen de Amazon ECR.

entorno/ tipo imagePullCredentials

Opcional. El tipo de credenciales que se CodeBuild utilizan para extraer imágenes de tu compilación. Hay dos valores válidos:

CODEBUILD

CODEBUILD especifica que CodeBuild usa sus propias credenciales. Debe editar la política de repositorios de Amazon ECR para confiar en el principal del CodeBuild servicio.

SERVICE_ROLE

Especifica que CodeBuild utiliza la función de servicio de su proyecto de compilación.

Cuando utilice una imagen de registro entre cuentas o privada, debe usar credenciales de SERVICE_ROLE. Cuando usas una imagen CodeBuild seleccionada, debes usar CODEBUILD credenciales.

environment/privilegedMode

Establézcalo como `true` si va a usar este proyecto de compilación para crear imágenes de Docker. De lo contrario, todas las compilaciones asociadas que intenten interactuar con el daemon de Docker producirán un error. También debe iniciar el daemon de Docker para que las compilaciones puedan interactuar con él. Una forma de hacerlo es iniciar el daemon Docker en la fase de `install` de su archivo `buildspec` ejecutando los siguientes comandos de compilación. No ejecute estos comandos si especificó una imagen de entorno de compilación proporcionada por el soporte CodeBuild de Docker.

Note

De forma predeterminada, el daemon de Docker está habilitado para compilaciones que no son de VPC. Si quieres usar contenedores Docker para compilaciones de VPC, [consulta Privilegios de tiempo de ejecución y capacidades de Linux](#) en el sitio web de Docker Docs y habilita el modo privilegiado. Además, Windows no admite el modo privilegiado.

```
- nohup /usr/local/bin/dockerd --host=unix:///var/run/docker.sock --  
host=tcp://127.0.0.1:2375 --storage-driver=overlay2 &  
- timeout 15 sh -c "until docker info; do echo .; sleep 1; done"
```

serviceRole

Obligatorio. El ARN del rol de servicio que se CodeBuild utiliza para interactuar con los servicios en nombre del usuario (por ejemplo, `arn:aws:iam::account-id:role/role-name`).

timeoutInMinutes

Opcional. El número de minutos, entre 5 y 480 (8 horas), tras los cuales CodeBuild se detiene la compilación si no se completa. Si no se especifica, se usa el valor predeterminado de 60. Para determinar si una compilación se CodeBuild detuvo debido a un tiempo de espera y cuándo, ejecuta el `batch-get-builds` comando. Para determinar si la compilación se ha detenido, busque en la salida un valor de `buildStatus` de `FAILED`. Para determinar cuándo se ha agotado el tiempo de espera de la compilación, busque en la salida el valor de `endTime` asociado a un valor de `phaseStatus` de `TIMED_OUT`.

queuedTimeoutInMinutes

Opcional. El número de minutos, entre 5 y 480 (8 horas), tras los que CodeBuild se detiene la compilación si sigue en cola. Si no se especifica, se usa el valor predeterminado de 60.

encryptionKey

Opcional. El alias o el ARN del AWS KMS key utilizado por CodeBuild para cifrar el resultado de la compilación. Si especifica un alias, utilice el formato `arn:aws:kms:region-ID:account-ID:key/key-ID` o si existe un alias, utilice el formato `alias/key-alias`. Si no se especifica, se utiliza la clave de KMS AWS administrada para Amazon S3.

etiquetas

Opcional. Matriz de objetos [Tag](#) que proporcionan las etiquetas que desea asociar a este proyecto de compilación. Puede especificar hasta 50 etiquetas. Cualquier AWS servicio que admita etiquetas de proyectos de CodeBuild compilación puede utilizar estas etiquetas. Cada etiqueta se expresa como un objeto con `key` y `value`.

vpcConfig

Opcional. Un [VpcConfig](#) objeto que contiene información sobre la configuración de la VPC del proyecto. Para obtener más información, consulte [Úselo AWS CodeBuild con Amazon Virtual Private Cloud](#).

Estas propiedades incluyen:

vpcl

Obligatorio. El ID de VPC que CodeBuild utiliza. Ejecute este comando para obtener una lista de todos los ID de la VPC de su región:

```
aws ec2 describe-vpcs --region <region-ID>
```

subnets

Obligatorio. Conjunto de identificadores de subred que incluyen los recursos utilizados por CodeBuild. Para obtener estos ID, ejecute este comando:

```
aws ec2 describe-subnets --filters "Name=vpc-id,Values=<vpc-id>" --region <region-ID>
```

securityGroupIds

Obligatorio. Conjunto de ID de grupos de seguridad que utiliza CodeBuild para permitir el acceso a los recursos de la VPC. Para obtener estos ID, ejecute este comando:

```
aws ec2 describe-security-groups --filters "Name=vpc-id,Values=<vpc-id>" --<region-ID>
```

badgeEnabled

Opcional. Especifica si se deben incluir distintivos de construcción en el proyecto. CodeBuild Configúrelo como `true` para activar las insignias de compilación o `false` en caso contrario. Para obtener más información, consulte [Crea un ejemplo de insignias con CodeBuild](#).

logsConfig

Un [LogsConfig](#) objeto que contiene información sobre la ubicación de los registros de esta compilación.

LogsConfig/cloudWatchLogs

[CloudWatchLogsConfig](#) Objeto que contiene información sobre cómo enviar registros a Logs CloudWatch

logsConfig/s3Logs

Un LogsConfig objeto de [S3](#) que contiene información sobre cómo enviar registros a Amazon S3.

fileSystemLocations

Opcional. Conjunto de [ProjectFileSystemsLocation](#) objetos que contiene información sobre la configuración de Amazon EFS.

buildBatchConfig

Opcional. El `buildBatchConfig` objeto es una [ProjectBuildBatchConfig](#) estructura que contiene la información de configuración de compilación por lotes del proyecto.

buildBatchConfig/ServiceRole

ARN del rol de servicio para el proyecto de compilación por lotes.

buildBatchConfig/Combine artefactos

Valor booleano que especifica si se deben combinar los artefactos de compilación de la compilación por lotes en una única ubicación de artefactos.

buildBatchConfig/restricciones/ maximumBuildsAllowed

Número máximo de compilaciones permitidas.

buildBatchConfig/restricciones/ computeTypesAllowed

Una matriz de cadenas que especifican los tipos de computación permitidos para la compilación por lotes. Consulte los valores en [Tipos de computación del entorno de compilación](#).

buildBatchConfig/timeoutInMinutes

Cantidad máxima de tiempo, en minutos, en que se debe completar la compilación por lotes.

buildBatchConfig/batchReportMode

Especifica cómo se envían los informes de estado de compilación al proveedor de origen para la compilación por lotes. Los valores válidos son:

REPORT_AGGREGATED_BATCH

(Predeterminado) Agregue todos los estados de compilación en un único informe de estado.

REPORT_INDIVIDUAL_BUILDS

Envíe un informe de estado independiente para cada compilación individual.

concurrentBuildLimit

Establece el número máximo de versiones simultáneas que están permitidas para este proyecto.

Solo se inician compilaciones nuevas si el número actual de compilaciones es menor o igual que este límite. Si el número de compilaciones actual alcanza este límite, se regulan las compilaciones nuevas y no se ejecutan.

Creación del proyecto

Para crear el proyecto, vuelva a ejecutar el comando [create-project](#) pasando su archivo JSON:

```
aws codebuild create-project --cli-input-json file://<json-file>
```

Si se ejecuta correctamente, la salida de la consola muestra la representación en JSON de un objeto [Project](#). Consulte la [sintaxis de CreateProject respuesta](#) para ver un ejemplo de estos datos.

Excepto el nombre del proyecto de compilación, puede cambiar cualquiera de los ajustes del proyecto de compilación más adelante. Para obtener más información, consulte [Cambiar la configuración de un proyecto de compilación \(AWS CLI\)](#).

Para empezar a ejecutar una compilación, consulte [Ejecutar una compilación \(AWS CLI\)](#).

Si el código fuente está almacenado en un GitHub repositorio y desea CodeBuild reconstruirlo cada vez que se introduzca un cambio de código en el repositorio, consulte [Iniciar la ejecución de compilaciones automáticamente \(AWS CLI\)](#).

Crear un proyecto de compilación (SDK de AWS)

Para obtener información sobre cómo usar AWS CodeBuild con los SDK de AWS, consulte [Referencia de los SDK y las herramientas de AWS](#).

Crear un proyecto de compilación (AWS CloudFormation)

Para obtener información acerca de cómo utilizar AWS CodeBuild con AWS CloudFormation, consulte [la plantilla de AWS CloudFormation para CodeBuild](#) en la Guía del usuario de AWS CloudFormation.

Creación de una regla de notificación

Puede utilizar reglas de notificación para notificar a los usuarios cuando se producen cambios importantes, como éxitos y errores de compilación. Las reglas de notificación especifican tanto los eventos como el tema de Amazon SNS que se utiliza para enviar notificaciones. Para obtener más información, consulte [¿Qué son las notificaciones?](#)

Puede utilizar la consola o la AWS CLI para crear reglas de notificación para AWS CodeBuild.

Para crear una regla de notificación (consola)

1. Inicie sesión en la AWS Management Console y abra la consola de CodeBuild en <https://console.aws.amazon.com/codebuild/>.
2. Elija Compilar, elija Proyectos de compilación y, a continuación, seleccione un proyecto de compilación en el que desee agregar notificaciones.
3. En la página del proyecto de compilación, elija Notificar y, a continuación, elija Crear regla de notificación. También puede ir a la página Configuración del proyecto de compilación y elegir Crear regla de notificación.
4. En Nombre de la notificación, introduzca un nombre para la regla.
5. En Tipo de detalle, elija Básico si desea que solo la información proporcionada en Amazon EventBridge se incluya en la notificación. Elija Completo si desea incluir la información proporcionada en Amazon EventBridge y otra información que CodePipeline o el administrador de notificaciones podría suministrar.

Para obtener más información, consulte [Descripción del contenido y la seguridad de las notificaciones](#).

6. En Eventos que activan notificaciones, seleccione los eventos para los que desea enviar notificaciones. Para obtener más información, consulte [Eventos de reglas de notificación en proyectos de compilación](#).
7. En Destinos, realice una de las siguientes operaciones:
 - Si ya ha configurado un recurso para utilizarlo con notificaciones, en Elegir tipo de destino elija AWS ChatbotSlack o Tema de SNS. En Elegir destino, elija el nombre del cliente (para un cliente de Slack configurado en AWS Chatbot) o el nombre de recurso de Amazon (ARN) del tema de Amazon SNS (para los temas de Amazon SNS ya configurados con la política necesaria para las notificaciones).
 - Si no ha configurado un recurso para utilizarlo con notificaciones, elija Crear destino y, a continuación, elija Tema de SNS. Indique el nombre del tema después de codestar-notifications- y, a continuación, elija Crear.

Note

- Si crea el tema de Amazon SNS durante la creación de la regla de notificación, se aplica la política que permite a la característica de notificaciones publicar eventos en el tema. El uso de un tema creado para las reglas de notificación lo ayuda a garantizar que solo suscriba a los usuarios que desea recibir notificaciones sobre este recurso.
- No se puede crear un cliente de AWS Chatbot durante la creación de una regla de notificación. Si elige AWS Chatbot (Slack), aparecerá un botón que le llevará a configurar un cliente en AWS Chatbot. Al elegir esa opción, se abrirá la consola de AWS Chatbot. Para obtener más información, consulte [Configurar la integración entre notificaciones y AWS Chatbot](#).
- Si desea utilizar un tema de Amazon SNS ya existente como destino, debe agregar la política necesaria para las notificaciones de AWS CodeStar además de otras políticas que puedan existir para ese tema. Para obtener más información, consulte [Configuración de temas de Amazon SNS para notificaciones](#) y [Descripción del contenido y la seguridad de las notificaciones](#).

8. Para terminar de crear la regla, elija Enviar.
9. Debe suscribir a los usuarios al tema de Amazon SNS de la regla antes de que puedan recibir notificaciones. Para obtener más información, vea [Suscribir usuarios a temas de Amazon SNS que son destinos](#). También puedes configurar la integración entre notificaciones y AWS Chatbot para enviar notificaciones a salas de chat de Amazon Chime. Para obtener más información, vea [Configurar la integración entre notificaciones y AWS Chatbot](#).

Para crear una regla de notificación (AWS CLI)

1. En un terminal o símbolo del sistema, ejecute el comando create-notification rule para generar el esqueleto JSON:

```
aws codestarnotifications create-notification-rule --generate-cli-skeleton  
> rule.json
```

Puede asignar al archivo el nombre que desee. En este ejemplo, el archivo se denomina *rule.json*.

- Abra el archivo JSON en un editor de texto sin formato y edítelo para incluir el recurso, los tipos de eventos y el destino que desea para la regla. En el ejemplo siguiente se muestra una regla de notificación denominada **MyNotificationRule** para un proyecto de compilación denominado *MyBuildProject* en una cuenta de AWS con el ID *123456789012*. Las notificaciones se envían con el tipo de detalle completo a un tema de Amazon SNS denominado *codestarnotifications-MyNotificationTopic* cuando las compilaciones son correctas:

```
{
  "Name": "MyNotificationRule",
  "EventTypeId": [
    "codebuild-project-build-state-succeeded"
  ],
  "Resource": "arn:aws:codebuild:us-east-2:123456789012:MyBuildProject",
  "Targets": [
    {
      "TargetType": "SNS",
      "TargetAddress": "arn:aws:sns:us-east-2:123456789012:codestarnotifications-MyNotificationTopic"
    }
  ],
  "Status": "ENABLED",
  "DetailType": "FULL"
}
```

Guarde el archivo.

- Mediante el archivo que acaba de modificar, en el terminal o línea de comandos, vuelva a ejecutar el comando `create-notification-rule` para crear la regla de notificación:

```
aws codestarnotifications create-notification-rule --cli-input-json
file://rule.json
```

- Si se ejecuta correctamente, el comando devuelve el ARN de la regla de notificación, similar a lo siguiente:

```
{
  "Arn": "arn:aws:codestarnotifications:us-east-1:123456789012:notificationrule/
dc82df7a-EXAMPLE"
}
```

Ver una lista de nombres de proyectos de compilación en AWS CodeBuild

Puede usar la consola de AWS CodeBuild, la AWS CLI o los SDK de AWS para ver una lista de proyectos de compilación en CodeBuild.

Temas

- [Ver una lista de nombres de proyectos de compilación \(consola\)](#)
- [Ver una lista de nombres de proyectos de compilación \(AWS CLI\)](#)
- [Ver una lista de nombres de proyectos de compilación \(SDK de AWS\)](#)

Ver una lista de nombres de proyectos de compilación (consola)

Puede ver una lista de proyectos de compilación en una región de AWS en la consola. La información incluye el nombre, el proveedor de origen, el repositorio, el estado de compilación más reciente y la descripción, en su caso.

1. Abra la consola de AWS CodeBuild en <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. En el panel de navegación, elija Build projects.

Note

De forma predeterminada, solo se muestran los 10 últimos proyectos de compilación. Para ver más proyectos de compilación, elija el icono de rueda dentada y, a continuación, seleccione un valor diferente para Projects per page (Proyectos por página) o use las flechas atrás y adelante.

Ver una lista de nombres de proyectos de compilación (AWS CLI)

Ejecute el comando list-projects:

```
aws codebuild list-projects --sort-by sort-by --sort-order sort-order --next-token next-token
```

En el comando anterior, sustituya los siguientes marcadores de posición:

- ***sort-by***: cadena opcional que se utiliza para indicar el criterio que se utiliza para mostrar los nombres de proyectos de compilación. Los valores válidos son:
 - **CREATED_TIME**: muestra los nombres de proyectos de compilación en función de la fecha de creación de cada proyecto.
 - **LAST_MODIFIED_TIME**: muestra los nombres de proyectos de compilación en función de la fecha de última modificación de cada proyecto.
 - **NAME**: muestra los nombres de proyectos de compilación en función del nombre de cada proyecto.
- ***sort-order***: cadena opcional que se utiliza para indicar el orden en que se muestran los proyectos de compilación, en función del valor de ***sort-by***. Los valores válidos son **ASCENDING** y **DESCENDING**.
- ***next-token***: cadena opcional. Durante una ejecución anterior, si hubiera más de 100 elementos en la lista, solo se devolverían los 100 primeros, junto con una única cadena denominada **next token**. Para obtener el siguiente lote de elementos de la lista, ejecute de nuevo este comando añadiendo el siguiente token a la llamada. Para obtener todos los elementos de la lista, siga ejecutando el comando con cada uno de los siguientes tokens hasta que no se devuelvan más tokens.

Por ejemplo, si ejecuta este comando:

```
aws codebuild list-projects --sort-by NAME --sort-order ASCENDING
```

Un resultado similar al siguiente podría aparecer en la salida:

```
{
  "nextToken": "Ci33ACF6...The full token has been omitted for brevity...U+AkMx8=",
  "projects": [
    "codebuild-demo-project",
    "codebuild-demo-project2",
    ... The full list of build project names has been omitted for brevity ...
    "codebuild-demo-project99"
  ]
}
```

Si ejecuta este comando de nuevo:

```
aws codebuild list-projects --sort-by NAME --sort-order ASCENDING --next-token  
Ci33ACF6...The full token has been omitted for brevity...U+AkMx8=
```

Un resultado similar al siguiente podría aparecer en la salida:

```
{  
  "projects": [  
    "codebuild-demo-project100",  
    "codebuild-demo-project101",  
    ... The full list of build project names has been omitted for brevity ...  
    "codebuild-demo-project122"  
  ]  
}
```

Ver una lista de nombres de proyectos de compilación (SDK de AWS)

Para obtener más información sobre cómo usar AWS CodeBuild con los SDK de AWS, consulte [Referencia de los SDK y las herramientas de AWS](#).

Ver los detalles de un proyecto de compilación en AWS CodeBuild

Puede usar la consola de AWS CodeBuild, la AWS CLI o los SDK de AWS para ver los detalles de un proyecto de compilación en CodeBuild.

Temas

- [Ver los detalles de un proyecto de compilación \(consola\)](#)
- [Ver los detalles de un proyecto de compilación \(AWS CLI\)](#)
- [Ver los detalles de un proyecto de compilación \(SDK de AWS\)](#)

Ver los detalles de un proyecto de compilación (consola)

1. Abra la consola de AWS CodeBuild en <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. En el panel de navegación, elija Build projects.

Note

De forma predeterminada, solo se muestran los 10 últimos proyectos de compilación. Para ver más proyectos de compilación, elija el icono de rueda dentada y, a continuación, seleccione un valor diferente para Projects per page (Proyectos por página) o use las flechas atrás y adelante.

3. En la lista de proyectos de compilación, en la columna Name (Nombre), elija el enlace del proyecto de compilación.
4. En la página Build project: (Proyecto de compilación:) **nombre-proyecto**, elija Build details (Detalles de compilación).

Ver los detalles de un proyecto de compilación (AWS CLI)

Ejecute el comando batch-get-projects:

```
aws codebuild batch-get-projects --names names
```

En el comando anterior, sustituya el siguiente marcador de posición:

- **names**: cadena obligatoria que se utiliza para indicar uno o varios nombres de proyectos de compilación para los que se desea ver los detalles. Para especificar varios proyectos de compilación, separe el nombre de cada proyecto de compilación con un espacio. Puede especificar hasta 100 nombres de proyecto de compilación. Para obtener una lista de proyectos de compilación, consulte [Ver una lista de nombres de proyectos de compilación \(AWS CLI\)](#).

Por ejemplo, si ejecuta este comando:

```
aws codebuild batch-get-projects --names codebuild-demo-project codebuild-demo-project2  
my-other-demo-project
```

Un resultado similar al siguiente podría aparecer en la salida. Los puntos suspensivos (. . .) se utilizan para representar datos omitidos por razones de brevedad.

```
{  
  "projectsNotFound": [  
    . . .  
  ]  
}
```

```
    "my-other-demo-project"
  ],
  "projects": [
    {
      ...
      "name": "codebuild-demo-project",
      ...
    },
    {
      ...
      "name": "codebuild-demo-project2",
      ...
    }
  ]
}
```

En el resultado anterior, la matriz `projectsNotFound` muestra todos los nombres de proyectos de compilación especificados, pero no encontrados. La matriz `projects` muestra los detalles de cada proyecto de compilación donde se encontró información. Los detalles del proyecto de compilación se han omitido del resultado anterior por razones de brevedad. Para obtener más información, consulte el resultado de [Crear un proyecto de compilación \(AWS CLI\)](#).

El comando `batch-get-projects` no admite el filtrado de determinados valores de propiedades, pero es posible escribir un script que enumere las propiedades de un proyecto. Por ejemplo, el siguiente script de intérprete de comandos de Linux enumera los proyectos de la región actual para la cuenta actual e imprime la imagen utilizada por cada proyecto.

```
#!/usr/bin/sh

# This script enumerates all of the projects for the current account
# in the current region and prints out the image that each project is using.

imageName=""

function getImageName(){
  local environmentValues=(${1//$/\t/ })
  imageName=${environmentValues[1]}
}

function processProjectInfo() {
  local projectInfo=$1
```

```
while IFS=$'\t' read -r section value; do
  if [[ "$section" == *"ENVIRONMENT"* ]]; then
    getImageName "$value"
  fi
done <<< "$projectInfo"
}

# Get the list of projects.
projectList=$(aws codebuild list-projects --output=text)

for projectName in $projectList
do
  if [[ "$projectName" != *"PROJECTS"* ]]; then
    echo "====="

    # Get the detailed information for the project.
    projectInfo=$(aws codebuild batch-get-projects --output=text --names
"$projectName")

    processProjectInfo "$projectInfo"

    printf 'Project "%s" has image "%s"\n' "$projectName" "$imageName"
  fi
done
```

Para obtener más información sobre cómo usar la AWS CLI con AWS CodeBuild, consulte [Referencia de la línea de comandos](#).

Ver los detalles de un proyecto de compilación (SDK de AWS)

Para obtener más información sobre cómo usar AWS CodeBuild con los SDK de AWS, consulte [Referencia de los SDK y las herramientas de AWS](#).

Almacenamiento en caché de compilaciones de AWS CodeBuild

Puede ahorrar tiempo si compila su proyecto utilizando una caché. Una caché puede almacenar fragmentos reutilizables del entorno de compilación y utilizarlos en diferentes compilaciones. El proyecto de compilación puede utilizar uno de los dos tipos de almacenamiento en caché: Amazon S3 o local. Si utiliza una caché local, debe elegir uno o varios de los tres modos disponibles: caché de origen, caché de capas de Docker y caché personalizada.

Note

El modo de caché de capas de Docker solamente está disponible en los entornos Linux. Si eliges este modo, debes ejecutar tu compilación en modo privilegiado. CodeBuild los proyectos a los que se les concede el modo privilegiado otorgan a su contenedor acceso a todos los dispositivos. Para obtener más información, consulte la sección sobre [privilegios en tiempo de ejecución y capacidades de Linux](#) en el sitio web de Docker Docs.

Temas

- [Almacenamiento en caché de Amazon S3](#)
- [Almacenamiento en la caché local](#)

Almacenamiento en caché de Amazon S3

El almacenamiento en caché de Amazon S3 aloja la caché en un bucket de Amazon S3 disponible en varios hosts de compilación. Esta opción resulta adecuada para artefactos de compilación de tamaño pequeño a intermedio que son más costosos de compilar que de descargar. Sin embargo, no es la mejor opción para los artefactos de compilación grandes, ya que tardan bastante tiempo en transferirse a través de la red, lo que puede afectar al rendimiento de la compilación. Tampoco es la mejor opción si usa capas de Docker.

Almacenamiento en la caché local

El almacenamiento en caché local aloja en una caché localmente en un host de compilación, que es el único host de compilación para el que está disponible la caché. Esta opción resulta adecuada para artefactos de compilación de tamaño intermedio a grande, ya que la caché está disponible de forma inmediata en el host de compilación. Esta no es la mejor opción si no hace muchas compilaciones. De este modo, el rendimiento de la compilación no se verá afectado por el tiempo de transferencia de la red.

Si elige la opción de almacenamiento en caché local, debe elegir uno o varios de los siguientes modos de caché:

- El modo de caché de origen almacena en la caché los metadatos de Git del origen principal y los orígenes secundarios. Una vez que se ha creado la caché, las compilaciones que se realicen posteriormente solo extraerán los cambios realizados entre las confirmaciones. Este modo es

conveniente para los proyectos que tienen un directorio de trabajo limpio y un origen que es un gran repositorio de Git. Si eliges esta opción y tu proyecto no usa un repositorio de Git (AWS CodeCommit GitHub, GitHub Enterprise Server o Bitbucket), la opción se ignora.

- El modo de caché de capas de Docker almacena en caché las capas de Docker existentes. Este modo es conveniente para los proyectos que compilan o extraen imágenes grandes de Docker. Puede evitar los problemas de rendimiento derivados de extraer imágenes grandes de Docker de la red.

Note

- Las cachés de capas de Docker solo pueden utilizarse en entornos Linux.
- Es necesario establecer la marca `privileged` para que el proyecto tenga los permisos de Docker necesarios.

De forma predeterminada, el daemon de Docker está habilitado para compilaciones que no son de VPC. Si quieres usar contenedores Docker para compilaciones de VPC, [consulta Privilegios de tiempo de ejecución y capacidades de Linux](#) en el sitio web de Docker Docs y habilita el modo privilegiado. Además, Windows no admite el modo privilegiado.

- Debe tener en cuenta la implicación de seguridad antes de utilizar una caché de capas de Docker.

- El modo de caché personalizada almacena en caché los directorios que se especifican en el archivo `buildspec`. Este modo es conveniente si el escenario de compilación no se ajusta a ninguno de los otros dos modos de caché local. Si utiliza una caché personalizada:
 - Solo se pueden especificar directorios para el almacenamiento en caché. No se pueden especificar archivos individuales.
 - Para hacer referencia a los directorios almacenados en la caché, se utiliza Symlinks.
 - Los directorios de la caché se asocian a la compilación antes de que se descarguen los orígenes del proyecto. Los elementos almacenados en la memoria caché invalidan los elementos de origen si tienen el mismo nombre. Los directorios se especifican utilizando rutas de caché en el archivo `buildspec`. Para obtener más información, consulte [Sintaxis de buildspec](#).
 - Evite los nombres de directorio que sean los mismos en el origen y en la memoria caché. Los directorios almacenados localmente en la memoria caché pueden invalidar o eliminar el contenido de los directorios del repositorio de origen que tienen el mismo nombre.

Note

El tipo de entorno de LINUX_GPU_CONTAINER y el tipo de procesamiento de BUILD_GENERAL1_2XLARGE no son compatibles con el almacenamiento en caché local. Para obtener más información, consulte [Modos y tipos de computación del entorno de compilación](#).

Note

No se admite el almacenamiento en caché local cuando se configura CodeBuild para trabajar con una VPC. Para obtener más información sobre el uso de VPC con CodeBuild, consulte [Úselo AWS CodeBuild con Amazon Virtual Private Cloud](#)

Temas

- [Especificar el almacenamiento en la caché local \(CLI\)](#)
- [Especificar el almacenamiento en la caché local \(consola\)](#)
- [Especificar el almacenamiento en la caché local \(AWS CloudFormation\)](#)

Puede utilizar la AWS CLI, la consola, el SDK o AWS CloudFormation para especificar una caché local.

Especificar el almacenamiento en la caché local (CLI)

Puede utilizar el parámetro `--cache` de la AWS CLI para especificar cada uno de los tres tipos de caché local.

- Para especificar una caché de origen:

```
--cache type=LOCAL,mode=[LOCAL_SOURCE_CACHE]
```

- Para especificar una caché de capas de Docker:

```
--cache type=LOCAL,mode=[LOCAL_DOCKER_LAYER_CACHE]
```

- Para especificar una caché personalizada:

```
--cache type=LOCAL,mode=[LOCAL_CUSTOM_CACHE]
```

Para obtener más información, consulte [Crear un proyecto de compilación \(AWS CLI\)](#).

Especificar el almacenamiento en la caché local (consola)

Puede especificar una caché en la sección Artifacts (Artefactos) de la consola. En Tipo de caché, seleccione Amazon S3 o Local. Si elige Local, seleccione una o varias de las tres opciones de caché local.

The screenshot shows the 'Cache type' dropdown menu with 'Local' selected. Below it, there is a section titled 'Select one or more local cache options.' with three checkboxes:

- Docker layer cache
Caches existing Docker layers so they can be reused. Requires privileged mode.
- Source cache
Caches .git metadata so subsequent builds only pull the change in commits.
- Custom cache
Caches directories specified in the buildspec file.

Para obtener más información, consulte [Creación de un proyecto de compilación \(consola\)](#).

Especificar el almacenamiento en la caché local (AWS CloudFormation)

Si utiliza AWS CloudFormation para especificar una caché local, en la propiedad Cache, en Type, especifique LOCAL. En el siguiente ejemplo de código de AWS CloudFormation con formato YAML, se especifican los tres tipos de caché local. Puede especificar cualquier combinación de tipos. Si utiliza una caché de capas de Docker, en Environment, debe establecer PrivilegedMode en true y Type en LINUX_CONTAINER.

```
CodeBuildProject:
  Type: AWS::CodeBuild::Project
  Properties:
    Name: MyProject
    ServiceRole: <service-role>
    Artifacts:
      Type: S3
```

```
Location: <bucket-name>
Name: myArtifact
EncryptionDisabled: true
OverrideArtifactName: true
Environment:
  Type: LINUX_CONTAINER
  ComputeType: BUILD_GENERAL1_SMALL
  Image: aws/codebuild/standard:5.0
  Certificate: <bucket/cert.zip>
  # PrivilegedMode must be true if you specify LOCAL_DOCKER_LAYER_CACHE
  PrivilegedMode: true
Source:
  Type: GITHUB
  Location: <github-location>
  InsecureSsl: true
  GitCloneDepth: 1
  ReportBuildStatus: false
TimeoutInMinutes: 10
Cache:
  Type: LOCAL
  Modes: # You can specify one or more cache mode,
    - LOCAL_CUSTOM_CACHE
    - LOCAL_DOCKER_LAYER_CACHE
    - LOCAL_SOURCE_CACHE
```

Note

De forma predeterminada, el daemon de Docker está habilitado para compilaciones que no son de VPC. Si quieres usar contenedores Docker para compilaciones de VPC, [consulta Privilegios de tiempo de ejecución y capacidades de Linux](#) en el sitio web de Docker Docs y habilita el modo privilegiado. Además, Windows no admite el modo privilegiado.

Para obtener más información, consulte [Crear un proyecto de compilación \(AWS CloudFormation\)](#).

Incorpora activadores AWS CodeBuild

Temas

- [Crear desencadenadores de AWS CodeBuild](#)
- [Editar desencadenadores de AWS CodeBuild](#)

Crear desencadenadores de AWS CodeBuild

Creación de desencadenadores de AWS CodeBuild (consola)

Puede crear un disparador en un proyecto para programar una compilación una vez cada hora, día o semana. También puedes crear un disparador mediante una regla personalizada con una expresión CloudWatch cron de Amazon. Por ejemplo, mediante una expresión cron puede programar una compilación en un momento específico todos los días de la semana.

Note

No es posible iniciar una compilación por lotes a partir de un activador de compilación, un EventBridge evento de Amazon o una AWS Step Functions tarea.

Para crear un disparador

1. Abra la consola de AWS CodeBuild en <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. En el panel de navegación, elija Proyectos de compilación.
3. Elija el enlace del proyecto de compilación al que desea agregar un disparador y, a continuación, elija la pestaña Build triggers (Disparadores de compilación).

Note

De forma predeterminada, solo se muestran los 100 últimos proyectos de compilación. Para ver más proyectos de compilación, elija el icono de rueda dentada y, a continuación, seleccione un valor diferente para Projects per page (Proyectos por página) o use las flechas atrás y adelante.

4. Elija Create trigger.
5. Escriba un nombre en Trigger name (Nombre del disparador).
6. En la lista desplegable Frequency (Frecuencia), elija la frecuencia del disparador. Si desea crear la frecuencia con una expresión Cron, seleccione Custom (Personalizado).
7. Especifique los parámetros de frecuencia del disparador. Puede introducir los primeros caracteres de su selección en el cuadro de texto para filtrar los elementos del menú desplegable.

Note

Las horas y minutos de inicio están basados en cero. El minuto de inicio es un número entre cero y 59. La hora de inicio es un número entre cero y 23. Por ejemplo, un desencadenador diario que comienza todos los días a las 12:15 PM, tiene 12 como hora de inicio y 15 como minuto de inicio. Un desencadenador diario que comienza cada día a medianoche tiene una hora de inicio 0 y un minuto de inicio 0. Un desencadenador diario que comienza todos los días a las 11:59 PM, tiene 23 como hora de inicio y 59 como minuto de inicio.

Frecuencia	Parámetros requeridos	Detalles
Por hora	Minuto de inicio	Utilice el menú desplegable Start minute (Minuto de inicio).
Por día	Minuto de inicio Hora de inicio	Utilice el menú desplegable Start minute (Minuto de inicio). Utilice el menú desplegable Start hour (Hora de inicio).
Semanal	Minuto de inicio Hora de inicio Día de inicio	Utilice el menú desplegable Start minute (Minuto de inicio). Utilice el menú desplegable Start hour (Hora de inicio). Utilice el menú desplegable Start day (Día de inicio).

Frecuencia	Parámetros requeridos	Detalles
Personalizada	Expresión Cron	<p>Introduzca una expresión Cron en Cron expression (Expresión Cron). Las expresiones Cron tienen seis campos obligatorios, que están separados por un espacio en blanco. Los campos especifican un valor de inicio de minuto, hora, día del mes, día de la semana y año. Puede utilizar caracteres comodín para especificar un intervalo, valores adicionales, etc. Por ejemplo, la expresión cron <code>0 9 ? * MON-FRI *</code> programa una compilación todos los días de la semana a las 9:00 a.m. Para obtener más información, consulte Cron Expressions en la Guía del usuario de Amazon CloudWatch Events.</p>

8. Seleccione **Enable this trigger** (Activar este disparador).
9. (Opcional) Expanda la sección **Advanced** (Avanzado). En **Source version** (Versión de código fuente), escriba una versión del código fuente.
 - En Amazon S3, introduzca el ID de versión que corresponda a la versión del artefacto de entrada que desea compilar. Si la **Source version** (Versión del código fuente) se deja en blanco, se usará la versión más reciente.
 - En AWS CodeCommit, especifique un ID de confirmación. Si **Source version** (Versión del código fuente) se deja en blanco, se utilizará el ID de confirmación HEAD de la ramificación predeterminada.

- En el GitHub caso de GitHub Enterprise, escribe un ID de confirmación, un ID de solicitud de extracción, un nombre de sucursal o un nombre de etiqueta que corresponda a la versión del código fuente que quieres crear. Si especifica un ID de solicitud de extracción, este debe tener el formato `pr/pull-request-ID` (por ejemplo, `pr/25`). Si especifica un nombre de ramificación, se usa el ID de confirmación HEAD de la ramificación. Si Source version (Versión del código fuente) se deja en blanco, se usa el ID de confirmación HEAD de la ramificación predeterminada.
 - En Bitbucket, especifique un ID de confirmación, un nombre de ramificación o un nombre de etiqueta que se corresponda con la versión de código fuente que desea compilar. Si especifica un nombre de ramificación, se usa el ID de confirmación HEAD de la ramificación. Si Source version (Versión del código fuente) se deja en blanco, se usa el ID de confirmación HEAD de la ramificación predeterminada.
10. (Opcional) Especifique un tiempo de espera entre 5 minutos y 480 minutos (8 horas). Este valor especifica durante cuánto tiempo AWS CodeBuild intenta ejecutar una compilación antes de detenerse. Si los campos Hours (Horas) y Minutes (Minutos) se dejan en blanco, se usa el valor de tiempo de espera predeterminado especificado en el proyecto.
 11. Elija Create trigger.

Creación de desencadenadores de AWS CodeBuild mediante programación

CodeBuild utiliza EventBridge las reglas de Amazon para crear activadores. Puedes usar la EventBridge API para crear activadores de compilación para tus CodeBuild proyectos de forma programática. Consulta la [referencia EventBridge de API de Amazon](#) para obtener más información.

Editar desencadenadores de AWS CodeBuild

Edición de desencadenadores de AWS CodeBuild (consola)

Puede editar un disparador en un proyecto para programar una compilación una vez cada hora, día o semana. También puedes editar un disparador para usar una regla personalizada con una expresión CloudWatch cron de Amazon. Por ejemplo, mediante una expresión cron puede programar una compilación en un momento específico todos los días de la semana. Para obtener información acerca de la creación de un disparador, consulte [Crear desencadenadores de AWS CodeBuild](#).

Para editar un desencadenador

1. Abra la consola de AWS CodeBuild en <https://console.aws.amazon.com/codesuite/codebuild/home>.

2. En el panel de navegación, elija Proyectos de compilación.
3. Elija el enlace del proyecto de compilación que desea cambiar y, a continuación, seleccione la pestaña Build triggers (Desencadenadores de compilación).

 Note

De forma predeterminada, solo se muestran los 100 últimos proyectos de compilación. Para ver más proyectos de compilación, elija el icono de rueda dentada y, a continuación, seleccione un valor diferente para Projects per page (Proyectos por página) o use las flechas atrás y adelante.

4. Elija el botón de opción junto al desencadenador que desea cambiar y, a continuación, elija Edit (Editar).
5. En la lista desplegable Frequency (Frecuencia), elija la frecuencia del disparador. Si desea crear la frecuencia con una expresión Cron, seleccione Custom (Personalizado).
6. Especifique los parámetros de frecuencia del disparador. Puede introducir los primeros caracteres de su selección en el cuadro de texto para filtrar los elementos del menú desplegable.

 Note

Las horas y minutos de inicio están basados en cero. El minuto de inicio es un número entre cero y 59. La hora de inicio es un número entre cero y 23. Por ejemplo, un desencadenador diario que comienza todos los días a las 12:15 PM, tiene 12 como hora de inicio y 15 como minuto de inicio. Un desencadenador diario que comienza cada día a medianoche tiene una hora de inicio 0 y un minuto de inicio 0. Un desencadenador diario que comienza todos los días a las 11:59 PM, tiene 23 como hora de inicio y 59 como minuto de inicio.

Frecuencia	Parámetros requeridos	Detalles
Por hora	Minuto de inicio	Utilice el menú desplegable Start minute (Minuto de inicio).

Frecuencia	Parámetros requeridos	Detalles
Por día	Minuto de inicio Hora de inicio	Utilice el menú desplegable Start minute (Minuto de inicio). Utilice el menú desplegable Start hour (Hora de inicio).
Semanal	Minuto de inicio Hora de inicio Día de inicio	Utilice el menú desplegable Start minute (Minuto de inicio). Utilice el menú desplegable Start hour (Hora de inicio). Utilice el menú desplegable Start day (Día de inicio).

Frecuencia	Parámetros requeridos	Detalles
Personalizada	Expresión Cron	<p>Introduzca una expresión Cron en Cron expresio n (Expresión Cron). Las expresiones Cron tienen seis campos obligatorios, que están separados por un espacio en blanco. Los campos especifican un valor de inicio de minuto, hora, día del mes, día de la semana y año. Puede utilizar caracteres comodín para especificar un intervalo , valores adicionales, etc. Por ejemplo, la expresión cron 0 9 ? * MON-FRI * programa una compilación todos los días de la semana a las 9:00 a.m. Para obtener más información, consulte Cron Expressions en la Guía del usuario de Amazon CloudWatch Events.</p>

7. Seleccione Enable this trigger (Activar este disparador).

 Note

Puedes usar la CloudWatch consola de Amazon en <https://console.aws.amazon.com/cloudwatch/> para editar la versión fuente, el tiempo de espera y otras opciones que no estén disponibles en AWS CodeBuild.

Edición de desencadenadores de AWS CodeBuild mediante programación

CodeBuild utiliza EventBridge las reglas de Amazon para crear activadores. Puedes usar la EventBridge API para editar mediante programación los activadores de compilación de tus CodeBuild proyectos. Consulta la [referencia EventBridge de API de Amazon](#) para obtener más información.

GitLab conexiones

Las conexiones le permiten autorizar y establecer configuraciones que asocian a su proveedor externo con AWS los recursos que utiliza AWS CodeConnections. Para asociar tu repositorio de terceros como fuente para tu proyecto de compilación, utilizas una conexión.

Para añadir un proveedor de fuentes GitLab autogestionado GitLab o autogestionado CodeBuild, puedes elegir entre las siguientes opciones:

- Utilice el asistente de creación de proyectos de compilación de la CodeBuild consola o la página Editar fuente para elegir la GitLab opción de proveedor GitLab autogestionado. Consulte [Crea una conexión a GitLab \(consola\)](#) para agregar el proveedor de fuentes. La consola le ayuda a crear un recurso de conexiones.
- Utilice la CLI para crear sus recursos de conexiones, consulte [Crear una conexión a GitLab \(CLI\)](#) para crear un recurso de conexiones con la CLI.

Note

También puede crear una conexión mediante la consola de Herramientas para desarrolladores, en Configuración. Consulte [Crear una conexión](#).

Note

Al autorizar la instalación de esta conexión en GitLab, usted otorga a nuestro servicio permisos para procesar sus datos al acceder a su cuenta, y puede revocar los permisos en cualquier momento desinstalando la aplicación.

Antes de empezar

- Debe haber creado ya una cuenta con. GitLab

Note

Las conexiones solo dan acceso a los repositorios que pertenecen a la cuenta que se utilizó para crear y autorizar la conexión.

Note

Puede crear conexiones a un repositorio en el que tenga el rol de propietario y GitLab, a continuación, la conexión se puede utilizar con el repositorio con recursos como CodeBuild: En el caso de los repositorios en grupos, no es necesario que sea el propietario del grupo.

- Para especificar una fuente para su proyecto de compilación, debe haber creado ya un repositorio en él GitLab.

Temas

- [Crea una conexión a GitLab \(consola\)](#)
- [Crear una conexión a GitLab \(CLI\)](#)

Crea una conexión a GitLab (consola)

Sigue estos pasos para usar la CodeBuild consola y añadir una conexión para tu proyecto (repositorio) en GitLab.

Para crear o editar tu proyecto de compilación

1. Inicia sesión en la CodeBuild consola.
2. Elija una de las siguientes opciones.
 - Elija crear un proyecto de compilación. Siga los pasos que se indican [Creación de un proyecto de compilación \(consola\)](#) para completar la primera pantalla y, en la sección Fuente, en Proveedor de código fuente, elija GitLab.
 - Elija editar un proyecto de construcción existente. Seleccione Editar y, a continuación, elija Fuente. En la página Editar fuente, en Proveedor de fuentes, elija GitLab.
3. Seleccione una de las siguientes opciones:

- En Conexión, selecciona Conexión predeterminada. La conexión predeterminada aplica una GitLab conexión predeterminada en todos los proyectos.
 - En Conexión, selecciona Conexión personalizada. La conexión personalizada aplica una GitLab conexión personalizada que anula la configuración predeterminada de tu cuenta.
4. Realice una de las siguientes acciones siguientes:
- En Conexión predeterminada o Conexión personalizada, si aún no ha creado una conexión con su proveedor, elija Crear una nueva GitLab conexión. Continúe con el paso 5 para crear la conexión.
 - En Conexión, si ya ha creado una conexión con su proveedor, seleccione la conexión. Continúe con el paso 10.

 Note

Si cierra la ventana emergente antes de crear una GitLab conexión, tendrá que actualizar la página.

5. Para crear una conexión a un GitLab repositorio, en Seleccione un proveedor, elija GitLab. En Nombre de la conexión, introduzca el nombre de la conexión que desea crear. Selecciona Conectar a GitLab.

Developer Tools > [Connections](#) > Create connection

Create a connection Info

Create GitLab connection Info

Connection name

► **Tags - optional**

[Connect to GitLab](#)

6. Cuando aparezca la página de inicio de GitLab sesión, inicia sesión con tus credenciales y, a continuación, selecciona Iniciar sesión.
7. Si es la primera vez que autorizas la conexión, aparecerá una página de autorización con un mensaje solicitando la autorización de la conexión para acceder a tu GitLab cuenta.

Seleccione Autorizar.

Authorize **AWS Connector for GitLab** to use your account?

An application called **AWS Connector for GitLab** is requesting access to your GitLab account. This application was created by **Amazon AWS**. Please note that this application is not provided by GitLab and you should verify its authenticity before allowing access.

This application will be able to:

- **Access the authenticated user's API**
Grants complete read/write access to the API, including all groups and projects, the container registry, the dependency proxy, and the package registry.
- **Read the authenticated user's personal information**
Grants read-only access to the authenticated user's profile through the /user API endpoint, which includes username, public email, and full name. Also grants access to read-only API endpoints under /users.
- **Read Api**
Grants read access to the API, including all groups and projects, the container registry, and the package registry.
- **Allows read-only access to the repository**
Grants read-only access to repositories on private projects using Git-over-HTTP or the Repository Files API.
- **Allows read-write access to the repository**
Grants read-write access to repositories on private projects using Git-over-HTTP (not using the API).

8. El navegador vuelve a la página de la consola de conexiones. En la configuración de la GitLab conexión, la nueva conexión se muestra en el nombre de la conexión.
9. Elija Conectar.

Cuando la GitLab conexión se haya creado correctamente, aparecerá un cartel de éxito en la parte superior.

10. En la página Crear proyecto de compilación, en la lista desplegable Conexión predeterminada o Conexión personalizada, asegúrese de que aparezca el ARN de su conexión. Si no es así, selecciona el botón de actualización para que aparezca.
11. En el repositorio, elija el nombre de su proyecto GitLab especificando la ruta del proyecto con el espacio de nombres. Por ejemplo, para un repositorio a nivel de grupo, introduzca el nombre del repositorio en el siguiente formato: `group-name/repository-name`. [Para obtener más información sobre la ruta y el espacio de nombres, consulta el `path_with_namespace` campo en `https://docs.gitlab.com/ee/api/projects.html#_get-single-project`](https://docs.gitlab.com/ee/api/projects.html#_get-single-project) [Para obtener más información sobre el espacio de nombres en GitLab, consulte `https://docs.gitlab.com/ee/user/namespace/`](https://docs.gitlab.com/ee/user/namespace/).

Note

En el caso de los grupos GitLab incluidos, debe especificar manualmente la ruta del proyecto con el espacio de nombres. Por ejemplo, para un repositorio con el nombre `myrepo` en un grupo `mygroup`, introduzca lo siguiente: `mygroup/myrepo`. Puedes encontrar la ruta del proyecto con el espacio de nombres en la URL de [GitLab](https://docs.gitlab.com/ee/user/namespace/)

12. En la versión fuente (opcional), introduce un ID de solicitud de extracción, una rama, un ID de confirmación, una etiqueta o una referencia y un ID de confirmación. Para obtener más información, consulte [Ejemplo de la versión fuente con AWS CodeBuild](#).

Note

Recomendamos seleccionar nombres de ramificaciones de Git que no parezcan identificadores de confirmación, como `811dd1ba1aba14473856cee38308caed7190c0d` o `5392f7`. Esto ayuda a evitar que la retirada desde Git colisione con confirmaciones reales.

13. En Git clone depth (opcional), puedes crear un clon superficial con un historial truncado al número especificado de confirmaciones. Si desea un clon completo, elija Completo.
14. En Estado de compilación (opcional), selecciona Informar del estado de la compilación al proveedor de origen cuando las compilaciones comiencen y terminen si quieres que se informe al proveedor de fuentes del estado de inicio y finalización de la compilación.

Para poder informar del estado de la compilación al proveedor de fuentes, el usuario asociado al proveedor de fuentes debe tener acceso de escritura al repositorio. Si el usuario no tiene acceso de escritura, no es posible actualizar el estado de compilación. Para obtener más información, consulte [Acceso al proveedor de fuentes](#).

Crear una conexión a GitLab (CLI)

Puede usar AWS Command Line Interface (AWS CLI) para crear una conexión.

Para ello, utilice el comando create-connection.

Important

Una conexión creada a través del AWS CLI o AWS CloudFormation está en PENDING estado de forma predeterminada. Después de crear una conexión con la CLI o AWS CloudFormation, utilice la consola para editar la conexión y establecer su estadoAVAILABLE.

Para crear una conexión

- Siga las instrucciones de la Guía del usuario de la consola Developer Tools para [crear una conexión a GitLab \(CLI\)](#).

Uso de webhooks con AWS CodeBuild

AWS CodeBuild admite la integración de webhook con GitHub GitHub Enterprise Server GitLab, GitLab Self Managed y Bitbucket.

Temas

- [Prácticas recomendadas para utilizar webhooks con AWS CodeBuild](#)
- [Eventos de webhooks de Bitbucket](#)

- [GitHub eventos de webhook](#)
- [GitLab eventos de webhook](#)

Prácticas recomendadas para utilizar webhooks con AWS CodeBuild

Para proyectos que utilizan repositorios públicos para configurar webhooks, le recomendamos las opciones siguientes:

Configurar filtros de ACTOR_ACCOUNT_ID

Añada ACTOR_ACCOUNT_ID filtros a los grupos de filtros de webhook de su proyecto para especificar qué usuarios pueden activar una compilación. Cada evento de webhook enviado CodeBuild incluye información sobre el remitente que especifica el identificador del actor. CodeBuild filtrará los webhooks en función del patrón de expresión regular proporcionado en los filtros. Es posible especificar los usuarios específicos que pueden activar compilaciones con este filtro. Para obtener más información, consulte [GitHub eventos de webhook](#) y [Eventos de webhooks de Bitbucket](#).

Configurar filtros de FILE_PATH

Agregue filtros de FILE_PATH a los grupos de filtros de webhook del proyecto para incluir o excluir los archivos que pueden activar una compilación cuando se modifican. Por ejemplo, es posible denegar las solicitudes de creación de cambios en el archivo `buildspec.yml` mediante un patrón de expresión regular como `^buildspec.yml$`, junto con la propiedad `excludeMatchedPattern`. Para obtener más información, consulte [GitHub eventos de webhook](#) y [Eventos de webhooks de Bitbucket](#).

Defina los permisos para su rol de IAM de compilación

Las compilaciones activadas por un webhook utilizan el rol de servicio de IAM especificado en el proyecto. Recomendamos establecer los permisos del rol de servicio en el conjunto mínimo de permisos necesario para ejecutar la compilación. Por ejemplo, en un escenario de prueba e implementación, cree un proyecto para las pruebas y otro para la implementación. El proyecto de prueba acepta compilaciones de webhook desde el repositorio, pero no proporciona permisos de escritura a sus recursos. El proyecto de implementación proporciona permisos de escritura a sus recursos y el filtro de webhook está configurado para permitir que solo los usuarios de confianza puedan activar las compilaciones.

Utilice una especificación de compilación en línea o almacenada en Amazon S3

Si define su especificación de compilación en línea dentro del propio proyecto o almacena el archivo de especificaciones de compilación en un bucket de Amazon S3, el archivo de especificaciones de compilación solo estará visible para el propietario del proyecto. Esto evita que las solicitudes de extracción realicen cambios en el código del archivo de especificaciones de compilación y generen compilaciones no deseadas. Para obtener más información, consulta [ProjectSource.buildspec](#) en la referencia de la API. CodeBuild

Eventos de webhooks de Bitbucket

Puede utilizar grupos de filtros de webhooks para especificar qué eventos de webhooks de Bitbucket van a desencadenar una compilación. Por ejemplo, es posible especificar que se desencadene una compilación solo con cambios en ramificaciones específicas.

Puede crear uno o varios grupos de filtros de webhooks para especificar qué eventos de webhooks van a desencadenar una compilación. Se activa una compilación si algún grupo de filtros da true como resultado, lo que ocurre cuando todos los filtros del grupo dan este resultado. Cuando cree un grupo de filtros, deberá especificar:

Un evento

En Bitbucket, puede seleccionar uno o más de los eventos siguientes:

- PUSH
- PULL_REQUEST_CREATED
- PULL_REQUEST_UPDATED
- PULL_REQUEST_MERGED
- PULL_REQUEST_CLOSED

El tipo de evento de webhook está en su encabezado en el campo X-Event-Key. En la siguiente tabla se muestra cómo los valores del encabezado X-Event-Key se asignan a los tipos de eventos.

Note

Debe habilitar el evento `merged` en su configuración de webhook de Bitbucket si crea un grupo de filtros de webhook que utilice el tipo de evento `PULL_REQUEST_MERGED`.

También debes habilitar el `declined` evento en la configuración de webhook de Bitbucket si creas un grupo de filtros de webhook que utilice el `PULL_REQUEST_CLOSED` tipo de evento.

Valor del encabezado X-Event-Key	Tipo de evento
<code>repo:push</code>	<code>PUSH</code>
<code>pullrequest:created</code>	<code>PULL_REQUEST_CREATED</code>
<code>pullrequest:updated</code>	<code>PULL_REQUEST_UPDATED</code>
<code>pullrequest:fulfilled</code>	<code>PULL_REQUEST_MERGED</code>
<code>pullrequest:rejected</code>	<code>PULL_REQUEST_CLOSED</code>

Para `PULL_REQUEST_MERGED`, si una solicitud de extracción se fusiona con la estrategia de compresión y la rama de solicitudes de extracción se cierra, la confirmación original de la solicitud de extracción deja de existir. En este caso, la variable de entorno `CODEBUILD_WEBHOOK_MERGE_COMMIT` contiene el identificador de la confirmación de fusión anulada.

Uno o más filtros opcionales

Utilice una expresión regular para especificar los filtros. En el caso de los eventos que desencadenan una compilación, todos los filtros de un grupo asociados con ellos deben dar `true` como resultado.

`ACTOR_ACCOUNT_ID` (`ACTOR_ID` en la consola)

Un evento de webhook desencadena una compilación cuando el ID de una cuenta de Bitbucket coincide con el patrón de la expresión regular. Este valor se encuentra en la propiedad `account_id` del objeto `actor` de la carga del filtro de webhook.

`HEAD_REF`

Un evento de webhook desencadena una compilación cuando la referencia del encabezado coincide con el patrón de la expresión regular (por ejemplo, `refs/heads/branch-name` y `refs/tags/tag-name`). Un filtro `HEAD_REF` evalúa el nombre de referencia de Git de la

ramificación o etiqueta. El nombre de la ramificación o la etiqueta se encuentra en el campo `name` del objeto `new` incluido en el objeto `push` de la carga del webhook. En el caso de los eventos de las solicitudes de extracción, el nombre de la ramificación se encuentra en el campo `name` del objeto `branch` incluido en el objeto `source` de la carga del webhook.

BASE_REF

Un evento de webhook desencadena una compilación cuando la referencia de base coincide con el patrón de la expresión regular. Un filtro `BASE_REF` solamente funciona con eventos de solicitudes de extracción (por ejemplo, `refs/heads/branch-name`). Un filtro `BASE_REF` evalúa el nombre de referencia de Git de la ramificación. El nombre de la ramificación se encuentra en el campo `name` del objeto `branch` incluido en el objeto `destination` de la carga del webhook.

FILE_PATH

Un webhook desencadena una compilación cuando la ruta de un archivo modificado coincide con el patrón de la expresión regular.

COMMIT_MESSAGE

Un webhook desencadena una compilación cuando el mensaje de confirmación del encabezado coincide con la expresión regular.

WORKFLOW_NAME

Un webhook activa una compilación cuando el nombre del flujo de trabajo coincide con el patrón de expresiones regulares.

Note

Puede encontrar la carga de webhook en la configuración de webhook del repositorio de Bitbucket.

Temas

- [Filtrar eventos de webhooks de Bitbucket \(consola\)](#)
- [Filtrar eventos de webhooks de Bitbucket \(SDK\)](#)
- [Filtrar eventos de webhooks de Bitbucket \(AWS CloudFormation\)](#)

Filtrar eventos de webhooks de Bitbucket (consola)

Para usar el AWS Management Console para filtrar los eventos del webhook:

1. Cuando cree el proyecto, seleccione Rebuild every time a code change is pushed to this repository (Volver a compilar cada vez que se inserte un cambio de código en este repositorio).
2. En Event type (Tipo de evento), seleccione uno o varios eventos.
3. Para filtrar en función de cuándo un evento va a desencadenar una compilación, en Start a build under these conditions (Iniciar una compilación en estas condiciones), añada uno o varios filtros opcionales.
4. Para filtrar en función de cuándo no se va a desencadenar un evento, en Don't start a build under these conditions (No iniciar una compilación en estas condiciones), añada uno o varios filtros opcionales.
5. Seleccione Add filter group (Añadir grupo de filtros) para añadir otro grupo de filtros.

Para obtener más información, consulta [Creación de un proyecto de compilación \(consola\)](#) y [WebhookFilter](#) en la Referencia de la AWS CodeBuild API.

En este ejemplo, un grupo de filtros de webhooks desencadena una compilación únicamente con solicitudes de extracción:

Filter group 1

Remove filter group

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PULL_REQUEST_CREATED ✕

PULL_REQUEST_UPDATED ✕

PULL_REQUEST_MERGED ✕

PULL_REQUEST_CLOSED ✕

► Start a build under these conditions - optional

► Don't start a build under these conditions - optional

Si utilizamos un ejemplo con dos grupos de filtros, la compilación se desencadenaría cuando uno de los grupos o los dos se evalúen como true:

- El primer grupo de filtros especifica las solicitudes de extracción que se crean o actualizan en ramificaciones con nombres de referencia de Git que coinciden con la expresión regular `^refs/heads/main$` y las referencias de encabezado que coinciden con `^refs/heads/branch1!`.
- El segundo grupo de filtros especifica solicitudes de inserción en ramificaciones con nombres de referencia de Git que coinciden con la expresión regular `^refs/heads/branch1$`.

Webhook event filter group 1

Event type
Add one or more a webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

▼ **Start a build under these conditions**

ACTOR_ID - optional <input type="text"/>	HEAD_REF - optional <input type="text" value="^refs/heads/branch1\$"/>	BASE_REF - optional <input type="text" value="^refs/heads/main\$"/>	FILE_PATH - optional <input type="text"/>
--	--	---	---

COMMIT_MESSAGE - optional

▶ **Don't start a build under these conditions**

Webhook event filter group 2

Event type
Add one or more a webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

▼ **Start a build under these conditions**

ACTOR_ID - optional <input type="text"/>	HEAD_REF - optional <input type="text" value="^refs/heads/branch1\$"/>	BASE_REF - optional <input type="text"/>	FILE_PATH - optional <input type="text"/>
--	--	--	---

COMMIT_MESSAGE - optional

▶ **Don't start a build under these conditions**

En este ejemplo, un grupo de filtros de webhooks desencadena una compilación de todas las solicitudes excepto los eventos de etiquetas.

Filter group 1 Remove filter group

Event type
Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PUSH X

PULL_REQUEST_CREATED X

PULL_REQUEST_UPDATED X

PULL_REQUEST_MERGED X

PULL_REQUEST_CLOSED X

▶ Start a build under these conditions - optional

▼ Don't start a build under these conditions - optional Add filter

Filter 1

Type

HEAD_REF▼

Pattern

^refs/tags/.*

En este ejemplo, un grupo de filtros de webhooks desencadena una compilación solo cuando se modifican los archivos cuyos nombres coinciden con la expresión regular `^buildspec.*`.

Webhook event filter group 1

Event type

PUSH X

▼ Start a build under these conditions

ACTOR_ID - optional

HEAD_REF - optional

BASE_REF - optional

FILE_PATH - optional

COMMIT_MESSAGE - optional

▶ Don't start a build under these conditions

En este ejemplo, un grupo de filtros de webhook activa una compilación solo cuando se cambian archivos en las carpetas `src` o `test`.

Webhook event filter group 1

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PUSH X

▼ Start a build under these conditions

ACTOR_ID - optional

HEAD_REF - optional

BASE_REF - optional

FILE_PATH - optional

COMMIT_MESSAGE - optional

▶ Don't start a build under these conditions

En este ejemplo, un grupo de filtros de webhooks desencadena una compilación únicamente cuando hay un cambio realizado por un usuario de Bitbucket que no tiene un ID de cuenta que coincida con la expresión regular `actor-account-id`.

Note

Para obtener información acerca de cómo encontrar el ID de cuenta de Bitbucket, consulte <https://api.bitbucket.org/2.0/users/nombre-usuario>, donde *nombre-usuario* es el nombre de usuario de Bitbucket.

Filter group 1

[Remove filter group](#)

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

[PUSH](#) ✕[PULL_REQUEST_CREATED](#) ✕[PULL_REQUEST_UPDATED](#) ✕[PULL_REQUEST_MERGED](#) ✕[PULL_REQUEST_CLOSED](#) ✕

▼ **Start a build under these conditions - optional**

[Add filter](#)

Filter 2

Type

Pattern

En este ejemplo, un grupo de filtros de webhook desencadena una compilación para un evento de inserción cuando el mensaje de confirmación de la cabeza coincide con la expresión regular `\[CodeBuild\]`.

Webhook event filter group 1

Event type

▼ Start a build under these conditions

ACTOR_ID - optional

HEAD_REF - optional

BASE_REF - optional

FILE_PATH - optional

COMMIT_MESSAGE - optional

► Don't start a build under these conditions

Filtrar eventos de webhooks de Bitbucket (SDK)

Si quieres usar el AWS CodeBuild SDK para filtrar los eventos de webhook, usa el `filterGroups` campo de la sintaxis de solicitud de los métodos `CreateWebhook` o de la `UpdateWebhook` API. Para obtener más información, consulta [WebhookFilter](#) la referencia de la CodeBuild API.

Si desea crear un filtro de webhook que desencadene una compilación únicamente con las solicitudes de extracción, inserte lo siguiente en la sintaxis de la solicitud:

```
"filterGroups": [
  [
    {
      "type": "EVENT",
      "pattern": "PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED, PULL_REQUEST_MERGED,
PULL_REQUEST_CLOSED"
    }
  ]
]
```

Si desea crear un filtro de webhooks que desencadene una compilación únicamente con las ramificaciones especificadas, utilice el parámetro `pattern` para especificar una expresión regular que filtre los nombres de las ramificaciones. Si utilizamos un ejemplo con dos grupos de filtros, la compilación se desencadenaría cuando uno de los grupos o los dos se evalúen como `true`:

- El primer grupo de filtros especifica las solicitudes de extracción que se crean o actualizan en ramificaciones con nombres de referencia de Git que coinciden con la expresión regular `^refs/heads/main$` y las referencias de encabezado que coinciden con `^refs/heads/myBranch$`.
- El segundo grupo de filtros especifica solicitudes de inserción en ramificaciones con nombres de referencia de Git que coinciden con la expresión regular `^refs/heads/myBranch$`.

```
"filterGroups": [  
  [  
    {  
      "type": "EVENT",  
      "pattern": "PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED, PULL_REQUEST_CLOSED"  
    },  
    {  
      "type": "HEAD_REF",  
      "pattern": "^refs/heads/myBranch$"  
    },  
    {  
      "type": "BASE_REF",  
      "pattern": "^refs/heads/main$"  
    }  
  ],  
  [  
    {  
      "type": "EVENT",  
      "pattern": "PUSH"  
    },  
    {  
      "type": "HEAD_REF",  
      "pattern": "^refs/heads/myBranch$"  
    }  
  ]  
]
```

Puede utilizar el parámetro `excludeMatchedPattern` para especificar qué eventos no desencadenan una compilación. En este ejemplo, se desencadena una compilación con todas las solicitudes, excepto los eventos de etiquetas.

```
"filterGroups": [  
  [  
    {  
      "type": "EVENT",
```

```

    "pattern": "PUSH, PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,
PULL_REQUEST_MERGED, PULL_REQUEST_CLOSED"
  },
  {
    "type": "HEAD_REF",
    "pattern": "^refs/tags/.*",
    "excludeMatchedPattern": true
  }
]
]

```

Puede crear un filtro que desencadene una compilación solo cuando un usuario de Bitbucket con el ID de cuenta `actor-account-id` realice algún cambio.

Note

Para obtener información acerca de cómo encontrar el ID de cuenta de Bitbucket, consulte <https://api.bitbucket.org/2.0/users/nombre-usuario>, donde *nombre-usuario* es el nombre de usuario de Bitbucket.

```

"filterGroups": [
  [
    {
      "type": "EVENT",
      "pattern": "PUSH, PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,
PULL_REQUEST_MERGED, PULL_REQUEST_CLOSED"
    },
    {
      "type": "ACTOR_ACCOUNT_ID",
      "pattern": "actor-account-id"
    }
  ]
]

```

Puede crear un filtro que desencadene una compilación solo cuando se modifiquen los archivos cuyos nombres coincidan con la expresión regular del argumento `pattern`. En este ejemplo, el grupo de filtros especifica que la compilación solo debe desencadenarse cuando se modifiquen los archivos cuyos nombres coincidan con la expresión regular `^buildspec.*`.

```

"filterGroups": [

```

```
[
  {
    "type": "EVENT",
    "pattern": "PUSH"
  },
  {
    "type": "FILE_PATH",
    "pattern": "^buildspec.*"
  }
]
```

En este ejemplo, el grupo de filtros especifica que solo se activa una compilación cuando se cambian archivos en las carpetas `src` o `test`.

```
"filterGroups": [
  [
    {
      "type": "EVENT",
      "pattern": "PUSH"
    },
    {
      "type": "FILE_PATH",
      "pattern": "^src/.+|^test/.+"
    }
  ]
]
```

Puede crear un filtro que desencadene una compilación solo cuando el mensaje de confirmación del encabezado coincida con la expresión regular en el argumento `pattern`. En este ejemplo, el grupo de filtros especifica que una compilación se desencadena solo cuando el mensaje de confirmación del encabezado del evento de inserción coincide con la expresión regular `\[CodeBuild\]`.

```
"filterGroups": [
  [
    {
      "type": "EVENT",
      "pattern": "PUSH"
    },
    {
      "type": "COMMIT_MESSAGE",
      "pattern": "\[CodeBuild\]"
    }
  ]
]
```

```

    }
  ]
]
```

Filtrar eventos de webhooks de Bitbucket (AWS CloudFormation)

Para usar una AWS CloudFormation plantilla para filtrar los eventos de webhook, usa la `FilterGroups` propiedad del AWS CodeBuild proyecto. El siguiente fragmento con formato YAML de una plantilla AWS CloudFormation crea dos grupos de filtros. Juntos, desencadenan una compilación cuando uno de los grupos o los dos se evalúan como `true`:

- El primer grupo de filtros especifica las solicitudes de extracción que un usuario de Bitbucket que no tiene el ID de cuenta 12345 crea o actualiza en ramificaciones con nombres de referencia de Git que coinciden con la expresión regular `^refs/heads/main$`.
- El segundo grupo de filtros especifica solicitudes de inserción que se crean en ramificaciones con nombres de referencia de Git que coinciden con la expresión regular `^refs/heads/.*`.
- El tercer grupo de filtros especifica una solicitud de inserción con un mensaje de confirmación del encabezado que coincida con la expresión regular `\[CodeBuild\]`.

```

CodeBuildProject:
  Type: AWS::CodeBuild::Project
  Properties:
    Name: MyProject
    ServiceRole: service-role
    Artifacts:
      Type: NO_ARTIFACTS
    Environment:
      Type: LINUX_CONTAINER
      ComputeType: BUILD_GENERAL1_SMALL
      Image: aws/codebuild/standard:5.0
    Source:
      Type: BITBUCKET
      Location: source-location
    Triggers:
      Webhook: true
      FilterGroups:
        - - Type: EVENT
          Pattern: PULL_REQUEST_CREATED,PULL_REQUEST_UPDATED
        - Type: BASE_REF
          Pattern: ^refs/heads/main$
```

```

    ExcludeMatchedPattern: false
  - Type: ACTOR_ACCOUNT_ID
    Pattern: 12345
    ExcludeMatchedPattern: true
  - - Type: EVENT
    Pattern: PUSH
  - Type: HEAD_REF
    Pattern: ^refs/heads/.+
  - Type: FILE_PATH
    Pattern: README
    ExcludeMatchedPattern: true
  - - Type: EVENT
    Pattern: PUSH
  - Type: COMMIT_MESSAGE
    Pattern: \[CodeBuild\]
  - Type: FILE_PATH
    Pattern: ^src/.+|^test/.+

```

GitHub eventos de webhook

Puedes usar grupos de filtros de webhook para especificar qué eventos de GitHub webhook activan una compilación. Por ejemplo, es posible especificar que se desencadene una compilación solo con cambios en ramificaciones específicas.

Puede crear uno o varios grupos de filtros de webhooks para especificar qué eventos de webhooks van a desencadenar una compilación. Se activa una compilación si algún grupo de filtros da true como resultado, lo que ocurre cuando todos los filtros del grupo dan este resultado. Cuando cree un grupo de filtros, deberá especificar:

Un evento

Para GitHub ello, puedes elegir uno o más de los siguientes eventos: PUSH, PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED, PULL_REQUEST_REOPENED, PULL_REQUEST_CLOSED, RELEASED, PRERELEASED, y. WORKFLOW_JOB_QUEUED El tipo de evento de webhook está en el encabezado X-GitHub-Event de la carga de webhook. En el encabezado X-GitHub-Event, es posible que vea pull_request o push. Para un evento de solicitud de extracción, el tipo está en el campo action de la carga del evento de webhook. En la siguiente tabla se muestra cómo los valores del encabezado X-GitHub-Event y los valores del campo action de la carga de solicitud de extracción de webhook se asignan a los tipos de eventos disponibles.

Valor del encabezado X-GitHub-Event	Valor action de la carga del evento de webhook	Tipo de evento
pull_request	opened	PULL_REQUEST_CREATED
pull_request	reopened	PULL_REQUEST_REOPENED
pull_request	synchronize	PULL_REQUEST_UPDATED
pull_request	closed y el campo merged es true	PULL_REQUEST_MERGED
pull_request	closed y el campo merged es false	PULL_REQUEST_CLOSED
push	n/a	PUSH
release	publicado	RELEASED
release	preeditado	PRERELEASED
workflow_job	queued	WORKFLOW_JOB_QUEUED

 Note

El tipo de PULL_REQUEST_REOPENED evento solo se puede usar con GitHub GitHub Enterprise Server. El tipo de WORKFLOW_JOB_QUEUED evento RELEASEDPRERELEASED, y GitHub solo se puede utilizar con. Para obtener más información sobre WORKFLOW_JOB_QUEUED, consulte [Tutorial: Configurar un ejecutor de acciones CodeBuild autohospedado GitHub](#).

Uno o más filtros opcionales

Utilice una expresión regular para especificar los filtros. En el caso de los eventos que desencadenan una compilación, todos los filtros de un grupo asociados con ellos deben dar true como resultado.

ACTOR_ACCOUNT_ID (ACTOR_ID en la consola)

Un evento de webhook desencadena una compilación cuando un identificador de cuenta de GitHub Enterprise Server GitHub o Enterprise Server coincide con el patrón de expresiones regulares. Este valor se encuentra en la propiedad `id` del objeto `sender` en la carga de webhook.

HEAD_REF

Un evento de webhook desencadena una compilación cuando la referencia del encabezado coincide con el patrón de la expresión regular (por ejemplo, `refs/heads/branch-name` o `refs/tags/tag-name`). Para un evento, el nombre de referencia se encuentra en la propiedad `ref` de la carga de webhook. Para los eventos de solicitud de extracción, el nombre de la ramificación se encuentra en la propiedad `ref` del objeto `head` de la carga de webhook.

BASE_REF

Un evento de webhook desencadena una compilación cuando la referencia de base coincide con el patrón de la expresión regular (por ejemplo, `refs/heads/branch-name`). Un filtro `BASE_REF` solo se puede utilizar con eventos de solicitud de extracción. El nombre de la ramificación se encuentra en la propiedad `ref` del objeto `base` de la carga de webhook.

FILE_PATH

Un evento de webhook desencadena una compilación cuando la ruta de un archivo modificado coincide con el patrón de la expresión regular. Se puede usar un `FILE_PATH` filtro con los eventos de solicitudes de GitHub inserción y extracción y con los eventos de inserción de GitHub Enterprise Server. No se puede usar con los eventos de solicitudes de extracción de GitHub Enterprise Server.

COMMIT_MESSAGE

Un webhook desencadena una compilación cuando el mensaje de confirmación del encabezado coincide con la expresión regular. Se puede usar un `COMMIT_MESSAGE` filtro con los eventos de solicitudes de GitHub inserción y extracción y con los eventos de inserción de GitHub Enterprise Server. No se puede usar con los eventos de solicitudes de extracción de GitHub Enterprise Server.

TAG_NAME

Un webhook activa una compilación cuando el nombre de la etiqueta de la publicación coincide con el patrón de expresiones regulares. Se puede usar un `TAG_NAME` filtro con los eventos de GitHub solicitud publicados y prepublicados.

RELEASE_NAME

Un webhook activa una compilación cuando el nombre de la versión coincide con el patrón de expresiones regulares. Se puede usar un RELEASE_NAME filtro con los eventos de GitHub solicitud publicados y prepublicados.

WORKFLOW_NAME

Un webhook activa una compilación cuando el nombre del flujo de trabajo coincide con el patrón de expresión regular. Se puede usar un WORKFLOW_NAME filtro con los eventos de solicitud de trabajos en cola del flujo de trabajo de GitHub Actions.

Note

Puedes encontrar la carga útil del webhook en la configuración del webhook de tu repositorio. GitHub

Temas

- [Filtra los eventos de GitHub webhook \(consola\)](#)
- [Filtra los eventos de GitHub webhook \(SDK\)](#)
- [Filtra los eventos de GitHub webhook \(AWS CloudFormation\)](#)

Filtra los eventos de GitHub webhook (consola)

En Eventos de webhook de origen principal, seleccione lo siguiente. Esta sección solo está disponible si seleccionas Repositorio en mi GitHub cuenta como repositorio de origen.

1. Cuando cree el proyecto, seleccione Rebuild every time a code change is pushed to this repository (Volver a compilar cada vez que se inserte un cambio de código en este repositorio).
2. En Event type (Tipo de evento), seleccione uno o varios eventos.
3. Para filtrar en función de cuándo un evento va a desencadenar una compilación, en Start a build under these conditions (Iniciar una compilación en estas condiciones), añada uno o varios filtros opcionales.
4. Para filtrar en función de cuándo no se va a desencadenar un evento, en Don't start a build under these conditions (No iniciar una compilación en estas condiciones), añada uno o varios filtros opcionales.

5. Seleccione Añadir grupo de filtros para añadir otro grupo de filtros.

Para obtener más información, consulta [Creación de un proyecto de compilación \(consola\)](#) y [WebhookFilter](#) en la referencia de la AWS CodeBuild API.

En este ejemplo, un grupo de filtros de webhooks desencadena una compilación únicamente con solicitudes de extracción:

The screenshot shows the configuration for a filter group in the AWS CodeBuild console. The title is "Filter group 1" and there is a "Remove filter group" button in the top right. Under "Event type", there is a text description: "Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository." Below this is a dropdown menu. Five event types are listed in blue boxes with an 'X' to remove them: "PULL_REQUEST_CREATED", "PULL_REQUEST_UPDATED", "PULL_REQUEST_REOPENED", "PULL_REQUEST_MERGED", and "PULL_REQUEST_CLOSED". At the bottom, there are two expandable sections: "Start a build under these conditions - optional" and "Don't start a build under these conditions - optional".

Si utilizamos un ejemplo con dos grupos de filtros de webhooks, la compilación se desencadenará cuando uno de los grupos o los dos se evalúen como true:

- El primer grupo de filtros especifica las solicitudes de extracción que se crean, actualizan o vuelven a abrirse en ramificaciones con nombres de referencia de Git que coinciden con la expresión regular `^refs/heads/main$` y las referencias de encabezado que coinciden con `^refs/heads/branch1$`.
- El segundo grupo de filtros especifica solicitudes de inserción en ramificaciones con nombres de referencia de Git que coinciden con la expresión regular `^refs/heads/branch1$`.

Webhook event filter group 1

Event type

Add one or more a webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PULL_REQUEST_CREATED X

PULL_REQUEST_UPDATED X

PULL_REQUEST_REOPENED X

▼ Start a build under these conditions

ACTOR_ID - optional

HEAD_REF - optional

^refs/heads/branch1\$

BASE_REF - optional

^refs/heads/main\$

FILE_PATH - optional

COMMIT_MESSAGE - optional

► Don't start a build under these conditions

Webhook event filter group 2

Remove filter group

Event type

Add one or more a webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PUSH X

▼ Start a build under these conditions

ACTOR_ID - optional

HEAD_REF - optional

^refs/heads/branch1\$

BASE_REF - optional

FILE_PATH - optional

COMMIT_MESSAGE - optional

► Don't start a build under these conditions

En este ejemplo, un grupo de filtros de webhooks desencadena una compilación de todas las solicitudes excepto los eventos de etiquetas.

Filter group 1

[Remove filter group](#)

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

[PUSH X](#)[PULL_REQUEST_CREATED X](#)[PULL_REQUEST_UPDATED X](#)[PULL_REQUEST_REOPENED X](#)[PULL_REQUEST_MERGED X](#)[PULL_REQUEST_CLOSED X](#)

► Start a build under these conditions - *optional*

▼ Don't start a build under these conditions - *optional*

[Add filter](#)

Filter 1

Type

Pattern

En este ejemplo, un grupo de filtros de webhooks desencadena una compilación solo cuando se modifican los archivos cuyos nombres coinciden con la expresión regular `^buildspec.*`.

Webhook event filter group 1

Event type

▼ Start a build under these conditions

ACTOR_ID - optional

HEAD_REF - optional

BASE_REF - optional

FILE_PATH - optional

 COMMIT_MESSAGE -
optional

► Don't start a build under these conditions

En este ejemplo, un grupo de filtros de webhook activa una compilación solo cuando se cambian archivos en las carpetas `src` o `test`.

Webhook event filter group 1

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

▼ Start a build under these conditions

ACTOR_ID - optional

HEAD_REF - optional

BASE_REF - optional

FILE_PATH - optional

 COMMIT_MESSAGE -
optional

► Don't start a build under these conditions

En este ejemplo, un grupo de filtros webhook activa una compilación solo cuando un usuario específico GitHub o de GitHub Enterprise Server con un identificador de cuenta que coincide con la expresión `actor-account-id` normal realiza un cambio.

Note

Para obtener información sobre cómo encontrar el ID de su GitHub cuenta, consulte [https://api.github.com/users/ *user-name*](https://api.github.com/users/user-name), donde *user-name* es su GitHub nombre de usuario.

Filter group 1

[Remove filter group](#)

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

[PUSH](#) ✕[PULL_REQUEST_CREATED](#) ✕[PULL_REQUEST_UPDATED](#) ✕[PULL_REQUEST_REOPENED](#) ✕[PULL_REQUEST_MERGED](#) ✕[PULL_REQUEST_CLOSED](#) ✕

▼ Start a build under these conditions - optional

[Add filter](#)

Filter 2

Type

Pattern

[Remove](#)

► Don't start a build under these conditions - optional

En este ejemplo, un grupo de filtros de webhook desencadena una compilación para un evento de inserción cuando el mensaje de confirmación de la cabeza coincide con la expresión regular `\[CodeBuild\]`.

Webhook event filter group 1

Event type

PUSH X

▼ Start a build under these conditions

ACTOR_ID - optional

HEAD_REF - optional

BASE_REF - optional

FILE_PATH - optional

COMMIT_MESSAGE - optional

► Don't start a build under these conditions

En este ejemplo, un grupo de filtros de webhook activa únicamente una compilación para eventos de trabajo del flujo de trabajo de GitHub Actions.

Note

CodeBuild solo procesará los trabajos del flujo de trabajo de GitHub Actions si un webhook tiene grupos de filtros que contengan el filtro de eventos `WORKFLOW_JOB_QUEUED`.

Filter group 1

Remove filter group

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

WORKFLOW_JOB_QUEUED X

► Start a build under these conditions - optional

► Don't start a build under these conditions - optional

En este ejemplo, un grupo de filtros de webhooks activa la creación de un nombre de flujo de trabajo que coincide con la expresión regular. `CI-CodeBuild`

Filter group 1

[Remove filter group](#)

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

WORKFLOW_JOB_QUEUED ✕

▼ Start a build under these conditions - optional

[Add filter](#)

Filter 1

Type

Pattern

[Remove](#)

▶ Don't start a build under these conditions - optional

Filtra los eventos de GitHub webhook (SDK)

Para usar el AWS CodeBuild SDK para filtrar los eventos de webhook, usa el `filterGroups` campo de la sintaxis de solicitud de los métodos `CreateWebhook` o de la `UpdateWebhook` API. Para obtener más información, consulta [WebhookFilter](#) la referencia de la CodeBuild API.

Si desea crear un filtro de webhook que desencadene una compilación únicamente con las solicitudes de extracción, inserte lo siguiente en la sintaxis de la solicitud:

```
"filterGroups": [
  [
    {
      "type": "EVENT",
      "pattern": "PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,
PULL_REQUEST_REOPENED, PULL_REQUEST_MERGED, PULL_REQUEST_CLOSED"
    }
  ]
]
```

Si desea crear un filtro de webhooks que desencadene una compilación únicamente con las ramificaciones especificadas, utilice el parámetro `pattern` para especificar una expresión regular que filtre los nombres de las ramificaciones. Si utilizamos un ejemplo con dos grupos de filtros, la compilación se desencadenaría cuando uno de los grupos o los dos se evalúen como `true`:

- El primer grupo de filtros especifica las solicitudes de extracción que se crean, actualizan o vuelven a abrirse en ramificaciones con nombres de referencia de Git que coinciden con la expresión regular `^refs/heads/main$` y las referencias de encabezado que coinciden con `^refs/heads/myBranch$`.
- El segundo grupo de filtros especifica solicitudes de inserción en ramificaciones con nombres de referencia de Git que coinciden con la expresión regular `^refs/heads/myBranch$`.

```
"filterGroups": [  
  [  
    {  
      "type": "EVENT",  
      "pattern": "PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,  
PULL_REQUEST_REOPENED"  
    },  
    {  
      "type": "HEAD_REF",  
      "pattern": "^refs/heads/myBranch$"  
    },  
    {  
      "type": "BASE_REF",  
      "pattern": "^refs/heads/main$"  
    }  
  ],  
  [  
    {  
      "type": "EVENT",  
      "pattern": "PUSH"  
    },  
    {  
      "type": "HEAD_REF",  
      "pattern": "^refs/heads/myBranch$"  
    }  
  ]  
]
```

Puede utilizar el parámetro `excludeMatchedPattern` para especificar qué eventos no desencadenan una compilación. Por ejemplo, en este caso, se desencadena una compilación con todas las solicitudes, excepto los eventos de etiquetas.

```
"filterGroups": [  
  [  
    {  
      "type": "EVENT",  
      "pattern": "PUSH, PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,  
PULL_REQUEST_REOPENED, PULL_REQUEST_MERGED, PULL_REQUEST_CLOSED"  
    },  
    {  
      "type": "HEAD_REF",  
      "pattern": "^refs/tags/.*",  
      "excludeMatchedPattern": true  
    }  
  ]  
]
```

Puede crear un filtro que desencadene una compilación solo cuando se modifiquen los archivos cuyos nombres coincidan con la expresión regular del argumento `pattern`. En este ejemplo, el grupo de filtros especifica que la compilación solo debe desencadenarse cuando se modifiquen los archivos cuyos nombres coincidan con la expresión regular `^buildspec.*`.

```
"filterGroups": [  
  [  
    {  
      "type": "EVENT",  
      "pattern": "PUSH"  
    },  
    {  
      "type": "FILE_PATH",  
      "pattern": "^buildspec.*"  
    }  
  ]  
]
```

En este ejemplo, el grupo de filtros especifica que solo se activa una compilación cuando se cambian archivos en las carpetas `src` o `test`.

```
"filterGroups": [  
  [  
    {  
      "type": "EVENT",  
      "pattern": "PUSH"  
    },  
    {  
      "type": "FILE_PATH",  
      "pattern": "src|test"    }  
  ]  
]
```

```
[
  {
    "type": "EVENT",
    "pattern": "PUSH"
  },
  {
    "type": "FILE_PATH",
    "pattern": "^src/.+|^test/.+"
  }
]
```

Puedes crear un filtro que active una compilación solo cuando un usuario específico GitHub o de GitHub Enterprise Server con un ID de cuenta realice un `actor-account-id`.

Note

Para obtener información sobre cómo encontrar el ID de su GitHub cuenta, consulte <https://api.github.com/users/user-name>, donde *user-name es su nombre* de GitHub usuario.

```
"filterGroups": [
  [
    {
      "type": "EVENT",
      "pattern": "PUSH, PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,
PULL_REQUEST_REOPENED, PULL_REQUEST_MERGED, PULL_REQUEST_CLOSED"
    },
    {
      "type": "ACTOR_ACCOUNT_ID",
      "pattern": "actor-account-id"
    }
  ]
]
```

Puede crear un filtro que desencadene una compilación solo cuando el mensaje de confirmación del encabezado coincida con la expresión regular en el argumento `pattern`. En este ejemplo, el grupo de filtros especifica que una compilación se desencadena solo cuando el mensaje de confirmación del encabezado del evento de inserción coincide con la expresión regular `\[CodeBuild\]`.

```
"filterGroups": [
```

```
[
  {
    "type": "EVENT",
    "pattern": "PUSH"
  },
  {
    "type": "COMMIT_MESSAGE",
    "pattern": "\\[CodeBuild\\]"
  }
]
```

Para crear un filtro de webhook que active únicamente trabajos de flujo de trabajo de creación para GitHub Actions, inserta lo siguiente en la sintaxis de la solicitud:

```
"filterGroups": [
  [
    {
      "type": "EVENT",
      "pattern": "WORKFLOW_JOB_QUEUED"
    }
  ]
]
```

Filtra los eventos de GitHub webhook ()AWS CloudFormation

Para usar una AWS CloudFormation plantilla para filtrar los eventos de webhook, usa la propiedad del AWS CodeBuild `FilterGroups` proyecto. El siguiente fragmento con formato YAML de una plantilla AWS CloudFormation crea dos grupos de filtros. Juntos, desencadenan una compilación cuando uno de los grupos o los dos se evalúan como true:

- El primer grupo de filtros especifica que las solicitudes de cambios se crean o actualizan en las sucursales con nombres de referencia de Git que coinciden con la expresión regular `^refs/heads/main$` por parte de un GitHub usuario que no tiene un ID de cuenta12345.
- El segundo grupo de filtros especifica las solicitudes de inserción que se crean en archivos cuyos nombres coinciden con la expresión regular `README` en las ramificaciones que tienen un nombre de referencia de Git que coincide con la expresión regular `^refs/heads/.*`.
- El tercer grupo de filtros especifica una solicitud de inserción con un mensaje de confirmación del encabezado que coincida con la expresión regular `\\[CodeBuild\\]`.

- El cuarto grupo de filtros especifica una solicitud de trabajo de flujo de trabajo de GitHub Actions con un nombre de flujo de trabajo que coincide con la expresión regular `\[CI-CodeBuild\]`.

```
CodeBuildProject:
  Type: AWS::CodeBuild::Project
  Properties:
    Name: MyProject
    ServiceRole: service-role
    Artifacts:
      Type: NO_ARTIFACTS
    Environment:
      Type: LINUX_CONTAINER
      ComputeType: BUILD_GENERAL1_SMALL
      Image: aws/codebuild/standard:5.0
    Source:
      Type: GITHUB
      Location: source-location
    Triggers:
      Webhook: true
      FilterGroups:
        - - Type: EVENT
          Pattern: PULL_REQUEST_CREATED,PULL_REQUEST_UPDATED
        - Type: BASE_REF
          Pattern: ^refs/heads/main$
          ExcludeMatchedPattern: false
        - Type: ACTOR_ACCOUNT_ID
          Pattern: 12345
          ExcludeMatchedPattern: true
        - - Type: EVENT
          Pattern: PUSH
        - Type: HEAD_REF
          Pattern: ^refs/heads/.+
        - Type: FILE_PATH
          Pattern: README
          ExcludeMatchedPattern: true
        - - Type: EVENT
          Pattern: PUSH
        - Type: COMMIT_MESSAGE
          Pattern: \[CodeBuild\]
        - Type: FILE_PATH
          Pattern: ^src/.+|^test/.+
        - - Type: EVENT
```

```

Pattern: WORKFLOW_JOB_QUEUED
- Type: WORKFLOW_NAME
Pattern: \[CI-CodeBuild\]

```

GitLab eventos de webhook

Puedes usar grupos de filtros de webhook para especificar qué eventos de GitLab webhook activan una compilación. Por ejemplo, es posible especificar que se desencadene una compilación solo con cambios en ramificaciones específicas.

Puede crear uno o varios grupos de filtros de webhooks para especificar qué eventos de webhooks van a desencadenar una compilación. Se activa una compilación si algún grupo de filtros da true como resultado, lo que ocurre cuando todos los filtros del grupo dan este resultado. Cuando cree un grupo de filtros, deberá especificar:

Un evento

Para GitLab ello, puedes elegir uno o más de los siguientes eventos:

- PUSH
- PULL_REQUEST_CREATED
- PULL_REQUEST_UPDATED
- PULL_REQUEST_MERGED

El tipo de evento de webhook está en su encabezado en el campo `X-Event-Key`. En la siguiente tabla se muestra cómo los valores del encabezado `X-Event-Key` se asignan a los tipos de eventos.

Note

Debes habilitar el `merged` evento en la configuración de tu GitLab webhook si creas un grupo de filtros de webhook que utilice el tipo de `PULL_REQUEST_MERGED` evento.

Valor del encabezado <code>X-Event-Key</code>	Tipo de evento
<code>repo:push</code>	PUSH
<code>pullrequest:created</code>	PULL_REQUEST_CREATED

Valor del encabezado X-Event-Key	Tipo de evento
<code>pullrequest:updated</code>	<code>PULL_REQUEST_UPDATED</code>
<code>pullrequest:fulfilled</code>	<code>PULL_REQUEST_MERGED</code>

Para `PULL_REQUEST_MERGED`, si una solicitud de extracción se fusiona con la estrategia de compresión y la rama de solicitudes de extracción se cierra, la confirmación original de la solicitud de extracción deja de existir. En este caso, la variable de entorno `CODEBUILD_WEBHOOK_MERGE_COMMIT` contiene el identificador de la confirmación de fusión anulada.

Uno o más filtros opcionales

Utilice una expresión regular para especificar los filtros. En el caso de los eventos que desencadenan una compilación, todos los filtros de un grupo asociados con ellos deben dar `true` como resultado.

`ACTOR_ACCOUNT_ID` (`ACTOR_ID` en la consola)

Un evento de webhook desencadena una compilación cuando el ID de una GitLab cuenta coincide con el patrón de expresión regular. Este valor se encuentra en la propiedad `account_id` del objeto `actor` de la carga del filtro de webhook.

`HEAD_REF`

Un evento de webhook desencadena una compilación cuando la referencia del encabezado coincide con el patrón de la expresión regular (por ejemplo, `refs/heads/branch-name` y `refs/tags/tag-name`). Un filtro `HEAD_REF` evalúa el nombre de referencia de Git de la ramificación o etiqueta. El nombre de la ramificación o la etiqueta se encuentra en el campo `name` del objeto `new` incluido en el objeto `push` de la carga del webhook. En el caso de los eventos de las solicitudes de extracción, el nombre de la ramificación se encuentra en el campo `name` del objeto `branch` incluido en el objeto `source` de la carga del webhook.

`BASE_REF`

Un evento de webhook desencadena una compilación cuando la referencia de base coincide con el patrón de la expresión regular. Un filtro `BASE_REF` solamente funciona con eventos de solicitudes de extracción (por ejemplo, `refs/heads/branch-name`). Un filtro `BASE_REF` evalúa el nombre de referencia de Git de la ramificación. El nombre de la ramificación se

encuentra en el campo name del objeto branch incluido en el objeto destination de la carga del webhook.

FILE_PATH

Un webhook desencadena una compilación cuando la ruta de un archivo modificado coincide con el patrón de la expresión regular.

COMMIT_MESSAGE

Un webhook desencadena una compilación cuando el mensaje de confirmación del encabezado coincide con la expresión regular.

Note

Puedes encontrar la carga útil del webhook en la configuración del webhook de tu repositorio. GitLab

Temas

- [Filtra los eventos de GitLab webhook \(consola\)](#)
- [Filtrar eventos de GitLab webhook \(SDK\)](#)
- [Filtrar eventos de GitLab webhook \(AWS CloudFormation\)](#)

Filtra los eventos de GitLab webhook (consola)

Para usar el AWS Management Console para filtrar los eventos de webhook:

1. Cuando cree el proyecto, seleccione Rebuild every time a code change is pushed to this repository (Volver a compilar cada vez que se inserte un cambio de código en este repositorio).
2. En Event type (Tipo de evento), seleccione uno o varios eventos.
3. Para filtrar en función de cuándo un evento va a desencadenar una compilación, en Start a build under these conditions (Iniciar una compilación en estas condiciones), añada uno o varios filtros opcionales.
4. Para filtrar en función de cuándo no se va a desencadenar un evento, en Don't start a build under these conditions (No iniciar una compilación en estas condiciones), añada uno o varios filtros opcionales.

5. Seleccione Add filter group (Añadir grupo de filtros) para añadir otro grupo de filtros.

Para obtener más información, consulta [Creación de un proyecto de compilación \(consola\)](#) y [WebhookFilter](#) en la Referencia de la AWS CodeBuild API.

En este ejemplo, un grupo de filtros de webhooks desencadena una compilación únicamente con solicitudes de extracción:

Filter group 1

Remove filter group

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PULL_REQUEST_CREATED ✕

PULL_REQUEST_UPDATED ✕

PULL_REQUEST_MERGED ✕

► Start a build under these conditions - *optional*

► Don't start a build under these conditions - *optional*

Si utilizamos un ejemplo con dos grupos de filtros, la compilación se desencadenaría cuando uno de los grupos o los dos se evalúen como true:

- El primer grupo de filtros especifica las solicitudes de extracción que se crean o actualizan en ramificaciones con nombres de referencia de Git que coinciden con la expresión regular `^refs/heads/main$` y las referencias de encabezado que coinciden con `^refs/heads/branch1!`.
- El segundo grupo de filtros especifica solicitudes de inserción en ramificaciones con nombres de referencia de Git que coinciden con la expresión regular `^refs/heads/branch1$`.

Webhook event filter groups

A build is triggered if any filter group evaluates to true, which occurs when all the filters in the group evaluate to true.

Filter group 1

[Remove filter group](#)

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PULL_REQUEST_CREATED ✕

PULL_REQUEST_UPDATED ✕

▼ Start a build under these conditions - optional

[Add filter](#)

Filter 1

Type

Pattern

[Remove](#)

Filter 2

Type

Pattern

[Remove](#)

► Don't start a build under these conditions - optional

Filter group 2

[Remove filter group](#)

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PUSH ✕

Filter 1

Type

En este ejemplo, un grupo de filtros de webhooks desencadena una compilación de todas las solicitudes excepto los eventos de etiquetas.

Filter group 1

[Remove filter group](#)

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

[PUSH X](#)[PULL_REQUEST_CREATED X](#)[PULL_REQUEST_UPDATED X](#)[PULL_REQUEST_MERGED X](#)

► Start a build under these conditions - *optional*

▼ Don't start a build under these conditions - *optional*

[Add filter](#)

Filter 1

Type

Pattern

En este ejemplo, un grupo de filtros de webhooks desencadena una compilación solo cuando se modifican los archivos cuyos nombres coinciden con la expresión regular `^buildspec.*`.

Webhook event filter groups

A build is triggered if any filter group evaluates to true, which occurs when all the filters in the group evaluate to true.

Filter group 1

[Remove filter group](#)

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

[PUSH X](#)

▼ Start a build under these conditions - *optional*

[Add filter](#)

Filter 1

Type

Pattern

[Remove](#)

► Don't start a build under these conditions - *optional*

En este ejemplo, un grupo de filtros de webhook activa una compilación solo cuando se cambian archivos en las carpetas `src` o `test`.

Webhook event filter groups

A build is triggered if any filter group evaluates to true, which occurs when all the filters in the group evaluate to true.

Filter group 1

[Remove filter group](#)

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PUSH X

▼ Start a build under these conditions - optional

[Add filter](#)

Filter 1

Type

FILE_PATH

Pattern

`^src/.+|^test/.+`

Remove

► Don't start a build under these conditions - optional

En este ejemplo, un grupo de filtros de webhook activa una compilación solo cuando un GitLab usuario que no tiene un ID de cuenta que coincida con la expresión `actor-account-id` normal realiza un cambio.

Note

Para obtener información sobre cómo encontrar tu ID de GitLab cuenta, consulta <https://api.github.com/users/user-name>, donde *user-name* es tu GitLab nombre de usuario.

Webhook event filter groups

A build is triggered if any filter group evaluates to true, which occurs when all the filters in the group evaluate to true.

Filter group 1

[Remove filter group](#)

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

[PUSH X](#)

▼ Start a build under these conditions - optional

[Add filter](#)

Filter 1

Type

Pattern

[Remove](#)

► Don't start a build under these conditions - optional

En este ejemplo, un grupo de filtros de webhook desencadena una compilación para un evento de inserción cuando el mensaje de confirmación de la cabeza coincide con la expresión regular `\[CodeBuild\]`.

Webhook event filter groups

A build is triggered if any filter group evaluates to true, which occurs when all the filters in the group evaluate to true.

Filter group 1

Remove filter group

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PUSH X

▼ Start a build under these conditions - optional

Add filter

Filter 1

Type

COMMIT_MESSAGE

Pattern

\[CodeBuild]\

Remove

► Don't start a build under these conditions - optional

Filtrar eventos de GitLab webhook (SDK)

Para usar el AWS CodeBuild SDK para filtrar los eventos de webhook, usa el `filterGroups` campo de la sintaxis de solicitud de los métodos `CreateWebhook` o de la `UpdateWebhook` API. Para obtener más información, consulta [WebhookFilter](#) la referencia de la CodeBuild API.

Si desea crear un filtro de webhook que desencadene una compilación únicamente con las solicitudes de extracción, inserte lo siguiente en la sintaxis de la solicitud:

```
"filterGroups": [
  [
    {
      "type": "EVENT",
      "pattern": "PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED, PULL_REQUEST_MERGED"
    }
  ]
]
```

```
]
]
```

Si desea crear un filtro de webhooks que desencadene una compilación únicamente con las ramificaciones especificadas, utilice el parámetro `pattern` para especificar una expresión regular que filtre los nombres de las ramificaciones. Si utilizamos un ejemplo con dos grupos de filtros, la compilación se desencadenaría cuando uno de los grupos o los dos se evalúen como `true`:

- El primer grupo de filtros especifica las solicitudes de extracción que se crean o actualizan en ramificaciones con nombres de referencia de Git que coinciden con la expresión regular `^refs/heads/main$` y las referencias de encabezado que coinciden con `^refs/heads/myBranch$`.
- El segundo grupo de filtros especifica solicitudes de inserción en ramificaciones con nombres de referencia de Git que coinciden con la expresión regular `^refs/heads/myBranch$`.

```
"filterGroups": [
  [
    {
      "type": "EVENT",
      "pattern": "PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED"
    },
    {
      "type": "HEAD_REF",
      "pattern": "^refs/heads/myBranch$"
    },
    {
      "type": "BASE_REF",
      "pattern": "^refs/heads/main$"
    }
  ],
  [
    {
      "type": "EVENT",
      "pattern": "PUSH"
    },
    {
      "type": "HEAD_REF",
      "pattern": "^refs/heads/myBranch$"
    }
  ]
]
```

Puede utilizar el parámetro `excludeMatchedPattern` para especificar qué eventos no desencadenan una compilación. En este ejemplo, se desencadena una compilación con todas las solicitudes, excepto los eventos de etiquetas.

```
"filterGroups": [  
  [  
    {  
      "type": "EVENT",  
      "pattern": "PUSH, PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,  
PULL_REQUEST_MERGED"  
    },  
    {  
      "type": "HEAD_REF",  
      "pattern": "^refs/tags/.*",  
      "excludeMatchedPattern": true  
    }  
  ]  
]
```

Puedes crear un filtro que active una compilación solo cuando un GitLab usuario con un ID de cuenta realice un cambio `actor-account-id`.

Note

Para obtener información sobre cómo encontrar tu ID de GitLab cuenta, consulta <https://api.github.com/users/user-name>, donde *user-name es tu nombre* de GitLab usuario.

```
"filterGroups": [  
  [  
    {  
      "type": "EVENT",  
      "pattern": "PUSH, PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,  
PULL_REQUEST_MERGED"  
    },  
    {  
      "type": "ACTOR_ACCOUNT_ID",  
      "pattern": "actor-account-id"  
    }  
  ]  
]
```

Puede crear un filtro que desencadene una compilación solo cuando se modifiquen los archivos cuyos nombres coincidan con la expresión regular del argumento `pattern`. En este ejemplo, el grupo de filtros especifica que la compilación solo debe desencadenarse cuando se modifiquen los archivos cuyos nombres coincidan con la expresión regular `^buildspec.*`.

```
"filterGroups": [  
  [  
    {  
      "type": "EVENT",  
      "pattern": "PUSH"  
    },  
    {  
      "type": "FILE_PATH",  
      "pattern": "^buildspec.*"  
    }  
  ]  
]
```

En este ejemplo, el grupo de filtros especifica que solo se activa una compilación cuando se cambian archivos en las carpetas `src` o `test`.

```
"filterGroups": [  
  [  
    {  
      "type": "EVENT",  
      "pattern": "PUSH"  
    },  
    {  
      "type": "FILE_PATH",  
      "pattern": "^src/.+|^test/.+"  
    }  
  ]  
]
```

Puede crear un filtro que desencadene una compilación solo cuando el mensaje de confirmación del encabezado coincida con la expresión regular en el argumento `pattern`. En este ejemplo, el grupo de filtros especifica que una compilación se desencadena solo cuando el mensaje de confirmación del encabezado del evento de inserción coincide con la expresión regular `\[CodeBuild\]`.

```
"filterGroups": [  
  [  

```

```

    {
      "type": "EVENT",
      "pattern": "PUSH"
    },
    {
      "type": "COMMIT_MESSAGE",
      "pattern": "\\[CodeBuild\\]"
    }
  ]
]

```

Filtrar eventos de GitLab webhook ()AWS CloudFormation

Para usar una AWS CloudFormation plantilla para filtrar los eventos de webhook, usa la propiedad del AWS CodeBuild `FilterGroups` proyecto. El siguiente fragmento con formato YAML de una plantilla AWS CloudFormation crea dos grupos de filtros. Juntos, desencadenan una compilación cuando uno de los grupos o los dos se evalúan como true:

- El primer grupo de filtros especifica que las solicitudes de cambios se crean o actualizan en las sucursales con nombres de referencia de Git que coinciden con la expresión regular `^refs/heads/main$` por parte de un GitLab usuario que no tiene un ID de cuenta12345.
- El segundo grupo de filtros especifica solicitudes de inserción que se crean en ramificaciones con nombres de referencia de Git que coinciden con la expresión regular `^refs/heads/.*`.
- El tercer grupo de filtros especifica una solicitud de inserción con un mensaje de confirmación del encabezado que coincida con la expresión regular `\\[CodeBuild\\]`.

```

CodeBuildProject:
  Type: AWS::CodeBuild::Project
  Properties:
    Name: MyProject
    ServiceRole: service-role
    Artifacts:
      Type: NO_ARTIFACTS
    Environment:
      Type: LINUX_CONTAINER
      ComputeType: BUILD_GENERAL1_SMALL
      Image: aws/codebuild/standard:5.0
    Source:
      Type: GITLAB
      Location: source-location

```

```
Triggers:
  Webhook: true
  FilterGroups:
    - - Type: EVENT
      Pattern: PULL_REQUEST_CREATED,PULL_REQUEST_UPDATED
    - - Type: BASE_REF
      Pattern: ^refs/heads/main$
      ExcludeMatchedPattern: false
    - - Type: ACTOR_ACCOUNT_ID
      Pattern: 12345
      ExcludeMatchedPattern: true
    - - Type: EVENT
      Pattern: PUSH
    - - Type: HEAD_REF
      Pattern: ^refs/heads/.*
    - - Type: EVENT
      Pattern: PUSH
    - - Type: COMMIT_MESSAGE
      Pattern: \[CodeBuild\]
```

Cambiar la configuración de un proyecto de compilación en AWS CodeBuild

Puede usar la consola de AWS CodeBuild, la AWS CLI o los SDK de AWS para cambiar la configuración de un proyecto de compilación.

Si añade un informe de pruebas a un proyecto de compilación, asegúrese de que su rol de IAM tenga los permisos descritos en [Uso de permisos de informes de pruebas](#).

Temas

- [Cambiar la configuración de un proyecto de compilación \(consola\)](#)
- [Cambiar la configuración de un proyecto de compilación \(AWS CLI\)](#)
- [Cambiar la configuración de un proyecto de compilación \(SDK de AWS\)](#)

Cambiar la configuración de un proyecto de compilación (consola)

Para cambiar la configuración de un proyecto de compilación, realice el procedimiento siguiente:

1. Abra la AWS CodeBuild consola en <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. En el panel de navegación, elija Proyectos de compilación.
3. Haga una de las acciones siguientes:

- Elija el enlace del proyecto de compilación que desea cambiar y, a continuación, seleccione Detalles de compilación.
- Seleccione el botón de opción situado junto al proyecto de compilación que desea cambiar, elija Ver detalles y, a continuación, elija Detalles de compilación.

Puede modificar las secciones siguientes:

Secciones

- [Configuración del proyecto](#)
- [Origen](#)
- [Entorno](#)
- [Buildspec](#)
- [Configuración de lotes](#)
- [Artefactos](#)
- [Registros](#)

Configuración del proyecto

En la sección Controladores de dominio, elija Editar. Cuando haya completado los cambios, seleccione Actualizar configuración para guardar la configuración nueva.

Puede modificar las propiedades siguientes.

Descripción

Introduzca una descripción opcional del proyecto de compilación para ayudar a otros usuarios a entender para qué se utiliza el proyecto.

Insignia de compilación

Seleccione Habilitar insignia de compilación para convertir el estado de compilación del proyecto en visible e incrustable. Para obtener más información, consulte [Ejemplo de insignias de compilación](#).

Note

La insignia de compilación no se aplica si el proveedor de código fuente es Amazon S3.

Habilitar el límite de compilaciones simultáneas

Si desea limitar el número de compilaciones simultáneas del proyecto, siga los pasos siguientes:

1. Seleccione **Restringir el número de compilaciones simultáneas** que puede iniciar este proyecto.
2. En **Límite de compilación simultánea**, introduzca el número máximo de compilaciones simultáneas permitidas para este proyecto. Este límite no puede ser superior al límite de compilaciones simultáneas establecido para la cuenta. Si intenta introducir un número mayor que el límite de la cuenta, aparecerá un mensaje de error.

Solo se inician compilaciones nuevas si el número actual de compilaciones es menor o igual que este límite. Si el número de compilaciones actual alcanza este límite, se regulan las compilaciones nuevas y no se ejecutan.

Habilitar el acceso a compilaciones públicas

Para que los resultados de compilación de tu proyecto estén disponibles para el público, incluidos los usuarios sin acceso a una AWS cuenta, selecciona **Habilitar el acceso público a la compilación** y confirma que deseas hacer públicos los resultados de la compilación. Las propiedades siguientes se utilizan en los proyectos de compilación pública:

Rol de servicio de compilación pública

Selecciona **Nuevo rol de servicio** si quieres CodeBuild crear un nuevo rol de servicio para ti, o **Rol de servicio existente** si quieres usar un rol de servicio existente.

La función de servicio de compilación pública CodeBuild permite leer los CloudWatch registros y descargar los artefactos de Amazon S3 para las compilaciones del proyecto. Esto es necesario para que los registros de compilación y los artefactos del proyecto estén disponibles para el público.

Rol de servicio

Introduzca el nombre del nuevo rol de servicio o de un rol de servicio existente.

Para que los resultados de compilación del proyecto sean privados, desactive la opción **Habilitar el acceso a la compilación pública**.

Para obtener más información, consulte [Proyectos de compilación pública en AWS CodeBuild](#).

⚠ Warning

A la hora de hacer públicos los resultados de compilación de un proyecto, es necesario tener en cuenta lo siguiente:

- Todos los resultados, registros y artefactos de la compilación del proyecto, incluidas las compilaciones que se hayan ejecutado cuando el proyecto era privado, están disponibles para el público.
- Todos los registros y artefactos de compilación están disponibles para el público. Es posible que las variables de entorno, el código fuente y otra información confidencial estén presentes en los registros y artefactos de compilación. Es necesario tener cuidado con la información que se envía a los registros de compilación. Algunas de las mejores prácticas son:
 - No almacene valores confidenciales, especialmente los ID de las claves de AWS acceso y las claves de acceso secretas, en las variables de entorno. Le recomendamos que utilice un almacén de parámetros de Amazon EC2 Systems Manager AWS Secrets Manager o que almacene valores confidenciales.
 - Siga [Prácticas recomendadas para utilizar webhooks](#) para limitar qué entidades pueden activar una compilación y no almacene las especificaciones de compilación en el propio proyecto para garantizar que los webhooks sean lo más seguros posible.
 - Un usuario malintencionado puede utilizar las compilaciones públicas para distribuir artefactos maliciosos. Recomendamos a los administradores del proyecto que revisen todas las solicitudes de extracción para comprobar que se trata de cambios legítimos. También recomendamos validar todos los artefactos con sus sumas de verificación para asegurarse de que se están descargando los artefactos correctos.

Información adicional

En el caso de las etiquetas, introduzca el nombre y el valor de las etiquetas que desee que utilicen AWS los servicios de soporte. Utilice Agregar fila para añadir una etiqueta. Puede añadir hasta 50 etiquetas.

Origen

En la sección Fuente, elija Editar. Cuando haya completado los cambios, seleccione Actualizar configuración para guardar la configuración nueva.

Puede modificar las propiedades siguientes:

Proveedor de fuentes

Seleccione el tipo de proveedor de código fuente. Utilice las listas siguientes para realizar las selecciones apropiadas para el proveedor de código fuente:

Note

CodeBuild no es compatible con Bitbucket Server.

Amazon S3

Bucket

Seleccione el nombre del bucket de entrada que contiene el código fuente.

Clave de objeto de S3 o carpeta de S3

Introduzca el nombre del archivo ZIP o la ruta a la carpeta que contiene el código fuente. Introduzca una barra diagonal (/) para descargarlo todo en el bucket de S3.

Versión de fuente

Introduzca el ID de versión del objeto que representa la compilación del archivo de entrada. Para obtener más información, consulte [Ejemplo de la versión fuente con AWS CodeBuild](#).

CodeCommit

Repositorio

Seleccione el repositorio que desea utilizar.

Tipo de referencia

Seleccione Ramificación, Etiqueta de Git o ID de confirmación para especificar la versión de su código fuente. Para obtener más información, consulte [Ejemplo de la versión fuente con AWS CodeBuild](#).

Note

Recomendamos seleccionar nombres de ramificaciones de Git que no parezcan identificadores de confirmación, como 811dd1ba1aba14473856cee38308caed7190c0d o 5392f7. Esto ayuda a evitar que la retirada desde Git colisione con confirmaciones reales.

Profundidad del clon de Git

Elija crear un clon superficial con un historial truncado en el número de confirmaciones especificado. Si desea un clon completo, elija Completo.

Submódulos de Git

Seleccione Usar submódulos de Git si desea incluir submódulos de Git en el repositorio.

Bitbucket

Repositorio

Elija Conectarse mediante OAuth o Conectarse con una contraseña de la aplicación Bitbucket y siga las instrucciones para conectarse o volver a conectarse a Bitbucket.

Seleccione un repositorio público o un repositorio de su cuenta.

Versión de fuente

Introduzca una ramificación, un ID de confirmación, una etiqueta o una referencia y un ID de confirmación. Para obtener más información, consulte [Ejemplo de la versión fuente con AWS CodeBuild](#).

Note

Recomendamos seleccionar nombres de ramificaciones de Git que no parezcan identificadores de confirmación, como 811dd1ba1aba14473856cee38308caed7190c0d o 5392f7. Esto ayuda a evitar que la retirada desde Git colisione con confirmaciones reales.

Profundidad del clon de Git

Seleccione Git clone depth (Profundidad del clon de Git) para crear un clon superficial con un historial truncado en el número de confirmaciones especificado. Si desea un clon completo, elija Completo.

Submódulos de Git

Seleccione Usar submódulos de Git si desea incluir submódulos de Git en el repositorio.

Estado de la compilación

Seleccione Notificar los estados de compilación al proveedor de fuentes al inicio y al final de las compilaciones si desea que se notifique el estado del inicio y finalización de la compilación al proveedor de fuentes.

Para poder informar del estado de la compilación al proveedor de fuentes, el usuario asociado al proveedor de fuentes debe tener acceso de escritura al repositorio. Si el usuario no tiene acceso de escritura, no es posible actualizar el estado de compilación. Para obtener más información, consulte [Acceso al proveedor de fuentes](#).

En Contexto de estado, introduzca el valor que se debe utilizar en el parámetro name en el estado de confirmación de Bitbucket. Para obtener más información, consulte [compilación](#) en la documentación de la API de Bitbucket.

En URL de destino, introduzca el valor que se debe utilizar en el parámetro url en el estado de confirmación de Bitbucket. Para obtener más información, consulte [compilación](#) en la documentación de la API de Bitbucket.

El estado de una compilación activada por un webhook siempre se notifica al proveedor de origen. Para que se notifique al proveedor de fuentes el estado de una compilación iniciada desde la consola o desde una llamada a la API, es necesario seleccionar esta opción.

Si las compilaciones de un proyecto se activan mediante un webhook, es necesario enviar una nueva confirmación al repositorio para que se aplique cualquier cambio en esta opción.

En los eventos de webhook de fuente principal, selecciona Reconstruir cada vez que se inserte un cambio de código en este repositorio si quieres CodeBuild compilar el código fuente cada vez que se inserte un cambio de código en este repositorio. Para obtener más información acerca de los webhooks y los grupos de filtros, consulte [Eventos de webhooks de Bitbucket](#).

GitHub

Repositorio

Elige Conectar mediante OAuth o Conectar con un token de acceso GitHub personal y sigue las instrucciones para conectarte (o volver a conectarte) GitHub y autorizar el acceso a AWS CodeBuild.

Seleccione un repositorio público o un repositorio de su cuenta.

Versión de fuente

Introduzca una ramificación, un ID de confirmación, una etiqueta o una referencia y un ID de confirmación. Para obtener más información, consulte [Ejemplo de la versión fuente con AWS CodeBuild](#).

Note

Recomendamos seleccionar nombres de ramificaciones de Git que no parezcan identificadores de confirmación, como 811dd1ba1aba14473856cee38308caed7190c0d o 5392f7. Esto ayuda a evitar que la retirada desde Git colisione con confirmaciones reales.

Profundidad del clon de Git

Seleccione Git clone depth (Profundidad del clon de Git) para crear un clon superficial con un historial truncado en el número de confirmaciones especificado. Si desea un clon completo, elija Completo.

Submódulos de Git

Seleccione Usar submódulos de Git si desea incluir submódulos de Git en el repositorio.

Estado de la compilación

Seleccione Notificar los estados de compilación al proveedor de fuentes al inicio y al final de las compilaciones si desea que se notifique el estado del inicio y finalización de la compilación al proveedor de fuentes.

Para poder informar del estado de la compilación al proveedor de fuentes, el usuario asociado al proveedor de fuentes debe tener acceso de escritura al repositorio. Si el usuario no tiene

acceso de escritura, no es posible actualizar el estado de compilación. Para obtener más información, consulte [Acceso al proveedor de fuentes](#).

En el contexto de estado, introduce el valor que se va a utilizar para el `context` parámetro en el GitHub estado de confirmación. Para obtener más información, consulta Cómo [crear un estado de confirmación](#) en la guía para GitHub desarrolladores.

En la URL de destino, introduce el valor que se va a utilizar para el `target_url` parámetro en el estado de GitHub confirmación. Para obtener más información, consulta Cómo [crear un estado de confirmación](#) en la guía para GitHub desarrolladores.

El estado de una compilación activada por un webhook siempre se notifica al proveedor de origen. Para que se notifique al proveedor de fuentes el estado de una compilación iniciada desde la consola o desde una llamada a la API, es necesario seleccionar esta opción.

Si las compilaciones de un proyecto se activan mediante un webhook, es necesario enviar una nueva confirmación al repositorio para que se aplique cualquier cambio en esta opción.

En los eventos de webhook de código fuente principal, selecciona Reconstruir cada vez que se envíe un cambio de código a este repositorio si quieres CodeBuild compilar el código fuente cada vez que se inserte un cambio de código en este repositorio. Para obtener más información acerca de los webhooks y los grupos de filtros, consulte [GitHub eventos de webhook](#).

GitHub Enterprise Server

GitHub Token de acceso personal empresarial

Consulte [GitHub Ejemplo de Enterprise Server](#) para obtener información sobre cómo copiar un token de acceso personal en el portapapeles. Pegue el token en el campo de texto y seleccione Guardar token.

Note

Solo es necesario introducir y guardar el token de acceso personal una vez. CodeBuild utiliza este token en todos los proyectos futuros.

Versión de fuente

Introduzca una solicitud de recuperación, ramificación, ID de confirmación, etiqueta o referencia y un ID de confirmación. Para obtener más información, consulte [Ejemplo de la versión fuente con AWS CodeBuild](#).

Note

Recomendamos seleccionar nombres de ramificaciones de Git que no parezcan identificadores de confirmación, como 811dd1ba1aba14473856cee38308caed7190c0d o 5392f7. Esto ayuda a evitar que la retirada desde Git colisione con confirmaciones reales.

Profundidad del clon de Git

Seleccione Git clone depth (Profundidad del clon de Git) para crear un clon superficial con un historial truncado en el número de confirmaciones especificado. Si desea un clon completo, elija Completo.

Submódulos de Git

Seleccione Usar submódulos de Git si desea incluir submódulos de Git en el repositorio.

Estado de la compilación

Seleccione Notificar los estados de compilación al proveedor de fuentes al inicio y al final de las compilaciones si desea que se notifique el estado del inicio y finalización de la compilación al proveedor de fuentes.

Para poder informar del estado de la compilación al proveedor de fuentes, el usuario asociado al proveedor de fuentes debe tener acceso de escritura al repositorio. Si el usuario no tiene acceso de escritura, no es posible actualizar el estado de compilación. Para obtener más información, consulte [Acceso al proveedor de fuentes](#).

En el contexto de estado, introduzca el valor que se utilizará para el context parámetro en el estado de GitHub confirmación. Para obtener más información, consulta Cómo [crear un estado de confirmación](#) en la guía para GitHub desarrolladores.

En la URL de destino, introduce el valor que se va a utilizar para el target_url parámetro en el estado de GitHub confirmación. Para obtener más información, consulta Cómo [crear un estado de confirmación](#) en la guía para GitHub desarrolladores.

El estado de una compilación activada por un webhook siempre se notifica al proveedor de origen. Para que se notifique al proveedor de fuentes el estado de una compilación iniciada desde la consola o desde una llamada a la API, es necesario seleccionar esta opción.

Si las compilaciones de un proyecto se activan mediante un webhook, es necesario enviar una nueva confirmación al repositorio para que se aplique cualquier cambio en esta opción.

SSL no segura

Selecciona Activar SSL inseguro para ignorar las advertencias de SSL al conectarte al repositorio de proyectos de tu GitHub empresa.

En Eventos de webhook de fuente principal, selecciona Reconstruir cada vez que se inserte un cambio de código en este repositorio si quieres CodeBuild compilar el código fuente cada vez que se inserte un cambio de código en este repositorio. Para obtener más información acerca de los webhooks y los grupos de filtros, consulte [GitHub eventos de webhook](#).

GitLab

Conexión

Conecta tu GitLab cuenta mediante AWS CodeConnections y usa la conexión para asociar tu repositorio de terceros como fuente para tu proyecto de compilación.

Elige Conexión predeterminada o Conexión personalizada.

La conexión predeterminada aplica una GitLab conexión predeterminada en todos los proyectos. La conexión personalizada aplica una GitLab conexión personalizada que anula la configuración predeterminada de tu cuenta.

Conexión predeterminada

El nombre de la conexión predeterminada asociada a tu cuenta.

Si aún no ha creado una conexión con su proveedor, consulte las [Crea una conexión a GitLab \(consola\)](#) instrucciones.

Conexión personalizada

Elija el nombre de la conexión personalizada que desee usar.

Si aún no ha creado una conexión con su proveedor, consulte las [Crea una conexión a GitLab \(consola\)](#) instrucciones.

Repositorio

Seleccione el repositorio que desea utilizar.

Versión de fuente

Introduce un ID de solicitud de extracción, una sucursal, un ID de confirmación, una etiqueta o una referencia y un ID de confirmación. Para obtener más información, consulte [Ejemplo de la versión fuente con AWS CodeBuild](#).

Note

Recomendamos seleccionar nombres de ramificaciones de Git que no parezcan identificadores de confirmación, como 811dd1ba1aba14473856cee38308caed7190c0d o 5392f7. Esto ayuda a evitar que la retirada desde Git colisione con confirmaciones reales.

Profundidad del clon de Git

Seleccione Git clone depth (Profundidad del clon de Git) para crear un clon superficial con un historial truncado en el número de confirmaciones especificado. Si desea un clon completo, elija Completo.

Estado de la compilación

Seleccione Notificar los estados de compilación al proveedor de fuentes al inicio y al final de las compilaciones si desea que se notifique el estado del inicio y finalización de la compilación al proveedor de fuentes.

Para poder informar del estado de la compilación al proveedor de fuentes, el usuario asociado al proveedor de fuentes debe tener acceso de escritura al repositorio. Si el usuario no tiene acceso de escritura, no es posible actualizar el estado de compilación. Para obtener más información, consulte [Acceso al proveedor de fuentes](#).

GitLab Self Managed

Conexión

Conecta tu GitLab cuenta mediante AWS CodeConnections y usa la conexión para asociar tu repositorio de terceros como fuente para tu proyecto de compilación.

Elige Conexión predeterminada o Conexión personalizada.

La conexión predeterminada aplica una conexión GitLab autogestionada predeterminada en todos los proyectos. La conexión personalizada aplica una conexión GitLab autogestionada personalizada que anula la configuración predeterminada de la cuenta.

Conexión predeterminada

El nombre de la conexión predeterminada asociada a tu cuenta.

Si aún no ha creado una conexión con su proveedor, consulte [Crear una conexión para GitLab autogestionarse](#) en la Guía del usuario de la consola Developer Tools para obtener instrucciones.

Conexión personalizada

Elija el nombre de la conexión personalizada que desee usar.

Si aún no ha creado una conexión con su proveedor, consulte [Crear una conexión GitLab autogestionada](#) en la Guía del usuario de la consola Developer Tools para obtener instrucciones.

Repositorio

Seleccione el repositorio que desea utilizar.

Versión de fuente

Introduce un ID de solicitud de extracción, una rama, un ID de confirmación, una etiqueta o una referencia y un ID de confirmación. Para obtener más información, consulte [Ejemplo de la versión fuente con AWS CodeBuild](#).

Note

Recomendamos seleccionar nombres de ramificaciones de Git que no parezcan identificadores de confirmación, como 811dd1ba1aba14473856cee38308caed7190c0d o 5392f7. Esto ayuda a evitar que la retirada desde Git colisione con confirmaciones reales.

Profundidad del clon de Git

Seleccione Git clone depth (Profundidad del clon de Git) para crear un clon superficial con un historial truncado en el número de confirmaciones especificado. Si desea un clon completo, elija Completo.

Estado de la compilación

Seleccione Notificar los estados de compilación al proveedor de fuentes al inicio y al final de las compilaciones si desea que se notifique el estado del inicio y finalización de la compilación al proveedor de fuentes.

Para poder informar del estado de la compilación al proveedor de fuentes, el usuario asociado al proveedor de fuentes debe tener acceso de escritura al repositorio. Si el usuario no tiene acceso de escritura, no es posible actualizar el estado de compilación. Para obtener más información, consulte [Acceso al proveedor de fuentes](#).

Entorno

En la sección Entorno, elija Editar. Cuando haya completado los cambios, seleccione Actualizar configuración para guardar la configuración nueva.

Puede modificar las propiedades siguientes:

Modelo de aprovisionamiento

Para cambiar el modelo de aprovisionamiento, elija Cambiar modelo de aprovisionamiento y realice una de las siguientes acciones:

- Para usar flotas bajo demanda administradas por AWS CodeBuild, elija On-Demand. Con flotas bajo demanda, CodeBuild proporciona computación para sus construcciones. Las máquinas se destruyen cuando finaliza la compilación. Las flotas bajo demanda están completamente administradas e incluyen capacidades de escalado automático para gestionar los picos de demanda.
- Para usar las flotas de capacidad reservada gestionadas por AWS CodeBuild, elija Capacidad reservada y, a continuación, seleccione un nombre de flota. Con las flotas de capacidad reservada, configura un conjunto de instancias dedicadas para su entorno de compilación. Estas máquinas permanecen inactivas, listas para procesar compilaciones o pruebas de forma inmediata y reducen la duración de las compilaciones. Con flotas de capacidad reservada, sus máquinas estarán siempre en funcionamiento y seguirán incurriendo en costos mientras estén aprovisionadas.

Para obtener más información, consulte [Trabajando con capacidad reservada en AWS CodeBuild](#).

Imagen del entorno

Para cambiar la imagen de la compilación, elija Anular imagen y realice una de las acciones siguientes:

- Para usar una imagen de Docker gestionada por AWS CodeBuild, elija Imagen gestionada y, a continuación, seleccione Sistema operativo, Tiempo de ejecución, Imagen y Versión de la imagen. Realice una selección en Environment type (Tipo de entorno) si está disponible.
- Para usar otra imagen de Docker, elija Custom image (Imagen personalizada). En Tipo de entorno, elija ARM, Linux, GPU de Linux o Windows. Si elige Otro registro, en URL de registro externo, introduzca el nombre y etiqueta de la imagen de Docker en Docker Hub, usando el formato *docker repository/docker image name*. Si elige Amazon ECR, utilice el repositorio de Amazon ECR y la imagen de Amazon ECR para elegir la imagen de Docker en su cuenta. AWS
- Para usar una imagen de Docker privada, elija Imagen personalizada. En Tipo de entorno, elija ARM, Linux, GPU de Linux o Windows. En Image registry (Registro de imagen), elija Other registry (Otro registro) y, a continuación, introduzca el ARN de las credenciales para su imagen de Docker privada. Es necesario crear las credenciales con Secrets Manager. Para obtener más información, consulte [AWS Secrets Manager](#) en la Guía del usuario de AWS Secrets Manager .

Note

CodeBuild anula la de las imágenes de Docker personalizadas `ENTRYPOINT`.

Rol de servicio

Realice una de las siguientes acciones siguientes:

- Si no tiene un rol de CodeBuild servicio, elija Nuevo rol de servicio. Escriba un nombre para el nuevo rol en el campo Role Name.
- Si tiene un rol CodeBuild de servicio, elija el rol de servicio existente. En ARN de rol, seleccione el rol de servicio.

Note

Cuando usas la consola para crear un proyecto de compilación, puedes crear un rol de CodeBuild servicio al mismo tiempo. De forma predeterminada, el rol funciona solamente con ese proyecto de compilación. Si utiliza la consola para asociar este rol de servicio a otro proyecto de compilación, el rol se actualizará para que funcione con el otro proyecto de compilación. Un rol de servicio puede funcionar con hasta diez proyectos de compilación.

Configuración adicional

Timeout (Tiempo de espera)

Especifique un valor, entre 5 minutos y 8 horas, tras el cual CodeBuild se detenga la compilación si no se completa. Si se dejan los campos de horas y minutos en blanco, se usará el valor predeterminado de 60 minutos.

Privilegiado

Seleccione Habilitar este indicador si desea crear imágenes de Docker o si desea que sus compilaciones obtengan privilegios elevados. solo si planea usar este proyecto de compilación para crear imágenes de Docker. De lo contrario, todas las compilaciones asociadas que intenten interactuar con el daemon de Docker producirán un error. También debe iniciar el daemon de Docker para que las compilaciones puedan interactuar con él. Una forma de hacer esto es inicializar el daemon de Docker en la fase `install` de la especificación de compilación ejecutando los siguientes comandos de compilación. No ejecute estos comandos si ha elegido una imagen de entorno de compilación proporcionada por el soporte CodeBuild de Docker.

Note

De forma predeterminada, el daemon de Docker está habilitado para compilaciones que no son de VPC. Si quieres usar contenedores Docker para compilaciones de VPC, [consulta Privilegios de tiempo de ejecución y capacidades de Linux](#) en el sitio web de Docker Docs y habilita el modo privilegiado. Además, Windows no admite el modo privilegiado.

```
- nohup /usr/local/bin/dockerd --host=unix:///var/run/docker.sock --  
host=tcp://127.0.0.1:2375 --storage-driver=overlay2 &  
- timeout 15 sh -c "until docker info; do echo .; sleep 1; done"
```

VPC

Si quieres CodeBuild trabajar con tu VPC:

- Para la VPC, elija el ID de VPC que utilice. CodeBuild
- Para las subredes de VPC, elija las subredes que incluyen los recursos que utiliza. CodeBuild
- Para los grupos de seguridad de VPC, elija los grupos de seguridad que se CodeBuild utilizan para permitir el acceso a los recursos de las VPC.

Para obtener más información, consulte [Úselo AWS CodeBuild con Amazon Virtual Private Cloud](#).

Computación

Seleccione una de las opciones disponibles.

Variables de entorno

Introduzca el nombre y el valor y, a continuación, seleccione el tipo de cada variable de entorno que se vaya a utilizar en las compilaciones.

Note

CodeBuild establece automáticamente la variable de entorno de su AWS región. Debe definir las siguientes variables de entorno, si no las ha agregado a su archivo `buildspec.yml`:

- `AWS_ACCOUNT_ID`
- `IMAGE_REPO_NAME`
- `IMAGE_TAG`

La consola y AWS CLI los usuarios pueden ver las variables de entorno. Si no le preocupa que la variable de entorno esté visible, establezca los campos Name y Value, y después establezca Type en Plaintext.

Se recomienda almacenar una variable de entorno con un valor confidencial, como un identificador de clave de AWS acceso, una clave de acceso AWS secreta o una contraseña, como parámetro en el almacén AWS Secrets Manager de parámetros de Amazon EC2 Systems Manager o.

Si utiliza el almacén de parámetros de Amazon EC2 Systems Manager, en Tipo, elija Parámetro. En Nombre, introduzca un identificador CodeBuild para la referencia. En Valor, escriba el nombre del parámetro tal como está almacenado en el almacén de parámetros de Amazon EC2 Systems Manager. Utilizando el parámetro denominado `/CodeBuild/dockerLoginPassword` como ejemplo, en Tipo, seleccione Parámetro. En Nombre, escriba `LOGIN_PASSWORD`. En Valor, introduzca `/CodeBuild/dockerLoginPassword`.

Important

Si utiliza el almacén de parámetros de Amazon EC2 Systems Manager, le recomendamos que almacene los parámetros con nombres de parámetros que comiencen por `/CodeBuild/` (por ejemplo, `/CodeBuild/dockerLoginPassword`). Puede utilizar la CodeBuild consola para crear un parámetro en Amazon EC2 Systems Manager. Seleccione Crear parámetro y siga las instrucciones del cuadro de diálogo. (En ese cuadro de diálogo, para la clave KMS, puede especificar el ARN de una AWS KMS clave de su cuenta. Amazon EC2 Systems Manager utiliza esta clave para cifrar el valor del parámetro durante el almacenamiento y descifrarlo durante la recuperación.) Si utiliza la CodeBuild consola para crear un parámetro, la consola comienza con el nombre del parámetro a `/CodeBuild/` medida que se va almacenando. Para obtener más información, consulte [Almacén de parámetros de Systems Manager](#) y [Tutorial de la consola del almacén de parámetros de Systems Manager](#) en la Guía del usuario de Amazon EC2 Systems Manager.

Si el proyecto de compilación hace referencia a parámetros almacenados en el almacén de parámetros de Amazon EC2 Systems Manager, el rol de servicio del proyecto de compilación debe permitir la acción `ssm:GetParameters`. Si seleccionaste Nueva función de servicio anteriormente, CodeBuild incluye esta acción en la función de servicio predeterminada de tu proyecto de compilación. Sin embargo, si ha seleccionado Rol de servicio existente, deberá incluir esta acción en el rol de servicio por separado.

Si el proyecto de compilación hace referencia a parámetros que se encuentran en el almacén de parámetros de Amazon EC2 Systems Manager cuyos nombres no empiezan por `/CodeBuild/` y ha seleccionado Nuevo rol de servicio, debe actualizar

ese rol de servicio para permitir el acceso a nombres de parámetro que no empiecen por `/CodeBuild/`. Esto es así porque el rol de servicio permite el acceso únicamente a los nombres de parámetro que empiezan por `/CodeBuild/`.

Si se selecciona Nuevo rol de servicio, el rol de servicio incluye permisos para descifrar todos los parámetros en el espacio de nombres de `/CodeBuild/` en el almacén de parámetros de Amazon EC2 Systems Manager.

Las variables de entorno que defina reemplazan las variables de entorno existentes. Por ejemplo, si la imagen de Docker ya contiene una variable de entorno denominada `MY_VAR` con un valor de `my_value` y establece una variable de entorno denominada `MY_VAR` con un valor de `other_value`, `my_value` se reemplaza por `other_value`. Asimismo, si la imagen de Docker ya contiene una variable de entorno denominada `PATH` con un valor de `/usr/local/sbin:/usr/local/bin` y establece una variable de entorno denominada `PATH` con un valor de `$PATH:/usr/share/ant/bin`, `/usr/local/sbin:/usr/local/bin` se reemplaza por el valor literal `$PATH:/usr/share/ant/bin`.

No establezca variables de entorno con un nombre que empiece por `CODEBUILD_`. Este prefijo se reserva para uso interno de .

Si se define una variable de entorno con el mismo nombre en varios lugares, el valor se determina de la siguiente manera:

- El valor de la llamada a la operación de inicio de la compilación tiene la máxima prioridad.
- El valor de la definición del proyecto de compilación es el siguiente en orden de prioridad.
- El valor en la declaración de especificación de compilación es el que menos prioridad tiene.

Si se utiliza Secrets Manager, seleccione Secrets Manager en Tipo. En Nombre, introduce un identificador para CodeBuild hacer referencia. En Valor, introduzca un `reference-key` utilizando el patrón `secret-id:json-key:version-stage:version-id`. Para obtener más información, consulte [Secrets Manager reference-key in the buildspec file](#).

Important

Si se utiliza Secrets Manager, recomendamos almacenar los secretos con nombres que comiencen por `/CodeBuild/` (por ejemplo, `/CodeBuild/`

`dockerLoginPassword`). Para obtener más información, consulte [AWS Secrets Manager](#) en la Guía del usuario de AWS Secrets Manager .

Si el proyecto de compilación hace referencia a secretos almacenados en Secrets Manager, el rol de servicio del proyecto de compilación debe permitir la acción `secretsmanager:GetSecretValue`. Si anteriormente seleccionaste Nueva función de servicio, CodeBuild incluye esta acción en la función de servicio predeterminada de tu proyecto de compilación. Sin embargo, si ha seleccionado Rol de servicio existente, deberá incluir esta acción en el rol de servicio por separado.

Si el proyecto de compilación hace referencia a secretos almacenados en Secrets Manager cuyos nombres no empiezan por `/CodeBuild/` y se selecciona Nuevo rol de servicio, es necesario actualizar ese rol de servicio para permitir el acceso a nombres de parámetro que no empiecen por `/CodeBuild/`. Esto es así porque el rol de servicio permite el acceso únicamente a los nombres de secreto que empiezan por `/CodeBuild/`.

Si elige Nuevo rol de servicio, el rol de servicio incluye permisos para descifrar todos los secretos en el espacio de nombres de `/CodeBuild/` en Secrets Manager.

Buildspec

En la sección Especificación de compilación, elija Editar. Cuando haya completado los cambios, seleccione Actualizar configuración para guardar la configuración nueva.

Puede modificar las propiedades siguientes:

Especificaciones de compilación

Realice una de las siguientes acciones siguientes:

- Si el código fuente incluye un archivo de especificación de compilación, elija Use a buildspec file (Usar un archivo buildspec). De forma predeterminada, CodeBuild busca un archivo con un nombre `buildspec.yml` en el directorio raíz del código fuente. Si el archivo de especificación de compilación usa un nombre o una ubicación diferentes, introduzca su ruta desde la raíz de la fuente en el nombre de la especificación de compilación (por ejemplo, `buildspec-two.yml` o `configuration/buildspec.yml`). Si el archivo de especificación de compilación está en un bucket de S3, debe estar en la misma región de AWS que el proyecto de compilación. Especifique el archivo de especificación de compilación mediante su ARN (por ejemplo, `arn:aws:s3:::<my-codebuild-sample2>/buildspec.yml`).

- Si el código fuente no incluye un archivo de especificación de compilación o si desea ejecutar comandos de compilación diferentes de los especificados para la fase `build` en el archivo `buildspec.yml` del directorio raíz de código fuente, elija Insert build commands (Insertar comandos de compilación). En Build commands (Comandos de compilación), escriba los comandos que desea ejecutar en la fase `build`. En caso de que haya varios comandos, separe cada comando con `&&` (por ejemplo, `mvn test && mvn package`). Para ejecutar comandos en otras fases, o si tiene una lista especialmente larga de comandos para la fase `build`, añada un archivo `buildspec.yml` al directorio raíz de código fuente, añada los comandos al archivo y, a continuación, elija Usar el archivo `buildspec.yml` del directorio raíz del código fuente.

Para obtener más información, consulte [Referencia de la especificación de compilación](#).

Configuración de lotes

En la sección Configuración de lotes, elija Editar. Cuando haya completado los cambios, seleccione Actualizar configuración para guardar la configuración nueva. Para obtener más información, consulte [Batch se integra AWS CodeBuild](#).

Puede modificar las propiedades siguientes:

Rol de servicio por lotes

Proporciona el rol de servicio para las compilaciones por lotes.

Elija una de las opciones siguientes:

- Si no tiene un rol de servicio, elija Nuevo rol de servicio. En Rol de servicio, introduzca un nombre nuevo para el rol nuevo.
- Si tiene un rol de servicio por lotes de CodeBuild, elija Rol de servicio existente. En Rol de servicio, elija el rol de servicio.

Las compilaciones por lotes introducen un nuevo rol de seguridad en la configuración por lotes. Este nuevo rol es obligatorio, ya que CodeBuild debes poder llamar a `RetryBuild` las acciones `StartBuild``StopBuild`, y en tu nombre para ejecutar compilaciones como parte de un lote. Los clientes deberían utilizar un nuevo rol, y no el mismo rol que utilizan para su compilación, por dos motivos:

- Al asignar los permisos de `StartBuild`, `StopBuild` y `RetryBuild` en el rol de compilación, una sola compilación podría iniciar más compilaciones a través de la especificación de compilación.

- CodeBuild Las compilaciones por lotes proporcionan restricciones que restringen la cantidad de compilaciones y tipos de procesamiento que se pueden usar para las compilaciones del lote. Si la función de compilación tiene estos permisos, es posible que las propias compilaciones puedan eludir estas restricciones.

Tipos de computación permitidos para lotes

Seleccione los tipos de computación permitidos para el lote. Seleccione todos los que correspondan.

Número máximo de compilaciones permitidas en el lote

Introduzca el número máximo de compilaciones permitidas en el lote. Si un lote supera este límite, fallará.

Tiempo de espera del lote

Introduzca la cantidad máxima de tiempo que puede durar la compilación del lote.

Combinar artefactos

Seleccione Combinar todos los artefactos del lote en una sola ubicación para tener todos los artefactos del lote combinados en una sola ubicación.

Modo de informe por lotes

Seleccione el modo de informe de estados de compilación deseado para las compilaciones por lotes.

Note

Este campo solo está disponible cuando la fuente del proyecto es Bitbucket o GitHub Enterprise GitHub, y en Fuente se selecciona Informar del estado de la compilación al proveedor de la fuente cuando las compilaciones comienzan y terminan.

Compilaciones agregadas

Seleccione esta opción para que los estados de todas las compilaciones del lote se combinen en un único informe de estado.

Compilaciones individuales

Seleccione esta opción para que se notifiquen los estados de compilación de cada compilación del lote por separado.

Artefactos

En la sección Artefactos, seleccione Editar. Cuando haya completado los cambios, seleccione Actualizar configuración para guardar la configuración nueva.

Puede modificar las propiedades siguientes:

Tipo

Realice una de las siguientes acciones siguientes:

- Si no desea crear ningún artefacto de salida de la compilación, elija No artifacts. Por ejemplo, puede interesar esta opción si solo va a ejecutar pruebas de compilación o si desea insertar una imagen de Docker en un repositorio de Amazon ECR.
- Para almacenar la salida de la compilación en un bucket de S3, seleccione Amazon S3 y, a continuación, haga lo siguiente:
 - Si desea utilizar su nombre de proyecto para el archivo ZIP o carpeta de salida de la compilación, deje en blanco Nombre. De lo contrario, escriba el nombre. (Si desea producir un archivo ZIP y desea que el archivo ZIP tenga una extensión de archivo, asegúrese de incluirlo detrás del nombre del archivo ZIP).
 - Seleccione Habilitar control semántico de versiones si desea que el nombre especificado en el archivo buildspec invalide cualquier nombre especificado en la consola. El nombre en un archivo buildspec se calcula en tiempo de compilación y utiliza el lenguaje de comandos Shell. Por ejemplo, puede adjuntar una fecha y una hora al nombre del artefacto para que siempre sea único. Los nombres de artefactos únicos impiden que los artefactos se sobrescriban. Para obtener más información, consulte [Sintaxis de buildspec](#).
 - En Bucket name (Nombre del bucket), seleccione el nombre del bucket de salida.
 - Si eligió Insert build commands (Insertar comandos de compilación) anteriormente en este procedimiento, en Output files (Archivos de salida), escriba las ubicaciones de los archivos de la compilación que desea incluir en el archivo ZIP o carpeta de salida de la compilación. Si hay varias ubicaciones, separe cada una de las ubicaciones con una coma (por ejemplo, `appspec.yml, target/my-app.jar`). Para obtener más información, consulte la descripción de files en [Sintaxis de buildspec](#).
 - Si no desea que los artefactos de compilación se cifren, seleccione Remove artifacts encryption (Quitar cifrado de artefactos),

Para cada conjunto de artefactos secundarios que desee:

1. Para Source identifier (Identificador de artefacto), escriba un valor que tenga como máximo 128 caracteres, únicamente de tipo alfanumérico y guiones bajos.
2. Elija Add artifact (Añadir artefacto).
3. Siga los pasos anteriores para configurar los artefactos secundarios.
4. Elija Save artifact (Guardar artefacto).

Configuración adicional

Clave de cifrado

Realice una de las siguientes acciones siguientes:

- Para usar Clave administrada de AWS Amazon S3 en su cuenta para cifrar los artefactos de salida de la compilación, deje la clave de cifrado en blanco. Esta es la opción predeterminada.
- Para utilizar una clave administrada por el cliente para cifrar los artefactos de salida de la compilación, en Clave de cifrado, introduzca el ARN de la clave administrada por el cliente. Utilice el formato `arn:aws:kms:region-ID:account-ID:key/key-ID`.

Tipo de caché

En Tipo de caché, seleccione una de las opciones siguientes:

- Si no desea usar una memoria caché, seleccione Sin caché.
- Si prefiere utilizar una caché de Amazon S3, seleccione Amazon S3 y haga lo siguiente:
 - En Bucket, señecopme el nombre del bucket de S3 donde se almacena la caché.
 - (Opcional) Para Prefijo de ruta de caché, escriba un prefijo de ruta de Amazon S3. El valor Prefijo de ruta de caché es similar a un nombre de directorio. Le permite almacenar la caché en el mismo directorio en un bucket.

Important

No añada una barra (/) al final del prefijo de ruta.

- Si desea utilizar una caché local, seleccione Local y elija uno o varios modos de caché local.

Note

El modo de caché de capas de Docker solo está disponible para Linux. Si lo selecciona, el proyecto deberá ejecutarse en modo con privilegios.

El uso de memoria caché ahorra mucho tiempo de compilación, ya que algunas partes reutilizables del entorno de compilación se almacenan en ella y se usan en las distintas compilaciones. Para obtener información acerca de cómo especificar una caché en el archivo de especificación de compilación, consulte [Sintaxis de buildspec](#). Para obtener más información acerca del almacenamiento en caché, consulte [Almacenamiento en caché de compilaciones de AWS CodeBuild](#).

Registros

En la sección Registros, seleccione Editar. Cuando haya completado los cambios, seleccione Actualizar configuración para guardar la configuración nueva.

Puede modificar las propiedades siguientes:

En Registros, seleccione los registros que desea crear. Puede crear CloudWatch registros de Amazon Logs, Amazon S3 o ambos.

CloudWatch

Si quieres los CloudWatch registros de Amazon Logs:

CloudWatch registros

Seleccione CloudWatch los registros.

Nombre del grupo

Introduce el nombre de tu grupo de CloudWatch registros de Amazon Logs.

Nombre de transmisión

Introduce el nombre de la secuencia de CloudWatch registro de Amazon Logs.

S3

Si desea registros de Amazon S3:

Registros de S3

Seleccione Registros de S3.

Bucket

Seleccione el nombre del bucket de S3 de los registros.

Prefijo de ruta

Introduzca el prefijo de sus registros.

Desactive el cifrado de registros de S3

Seleccione si no desea cifrar los registros de S3.

Cambiar la configuración de un proyecto de compilación (AWS CLI)

Para obtener información sobre cómo usar la AWS CLI con AWS CodeBuild, consulte [Referencia de la línea de comandos](#).

Para actualizar un proyecto de CodeBuild con AWS CLI, debe crear un archivo JSON con las propiedades actualizadas y pasar ese archivo al comando [update-project](#). Las propiedades que no estén incluidas en el archivo de actualización permanecen sin cambios.

En el archivo JSON de actualización, solo se requieren la propiedad name y las propiedades modificadas. La propiedad name identifica el proyecto que se va a modificar. Para cualquier estructura modificada, también se deben incluir los parámetros necesarios para esas estructuras. Por ejemplo, para modificar el entorno del proyecto, se necesitan las propiedades environment/type y environment/computeType. Este es un ejemplo que actualiza la imagen del entorno:

```
{
  "name": "<project-name>",
  "environment": {
    "type": "LINUX_CONTAINER",
    "computeType": "BUILD_GENERAL1_SMALL",
    "image": "aws/codebuild/amazonlinux2-x86_64-standard:4.0"
  }
}
```

Si necesita obtener los valores de las propiedades actuales de un proyecto, utilice el comando [batch-get-projects](#) para obtener las propiedades actuales del proyecto que va a modificar y escriba el resultado en un archivo.

```
aws codebuild batch-get-projects --names "<project-name>" > project-info.json
```

El archivo *project-info.json* contiene una matriz de proyectos, por lo que no se puede usar directamente para actualizar un proyecto. Sin embargo, puede copiar las propiedades que desee modificar del archivo *project-info.json* y pegarlas en el archivo de actualización como referencia de las propiedades que desee modificar. Para obtener más información, consulte [Ver los detalles de un proyecto de compilación \(AWS CLI\)](#).

Modifique el archivo de actualización JSON como se describe en [Crear un proyecto de compilación \(AWS CLI\)](#) y guarde los resultados. Cuando termine de modificar el archivo de actualización JSON, ejecute el comando [update-project](#) y transfiera el archivo de actualización JSON.

```
aws codebuild update-project --cli-input-json file://<update-project-file>
```

Si se realiza correctamente, la salida muestra el JSON del proyecto actualizado. Si falta algún parámetro obligatorio, aparece un mensaje de error en el resultado que identifica los parámetros que faltan. Por ejemplo, este es el mensaje de error que se muestra si falta el parámetro `environment/type`:

```
aws codebuild update-project --cli-input-json file://update-project.json
```

```
Parameter validation failed:  
Missing required parameter in environment: "type"
```

Cambiar la configuración de un proyecto de compilación (SDK de AWS)

Para obtener información sobre cómo usar AWS CodeBuild con los SDK de AWS, consulte [Referencia de los SDK y las herramientas de AWS](#).

Eliminar un proyecto de compilación en AWS CodeBuild

Puede usar la consola de CodeBuild, la AWS CLI o los SDK de AWS para eliminar un proyecto de compilación en CodeBuild. Si elimina un proyecto, sus compilaciones no se eliminan.

⚠ Warning

No se puede eliminar un proyecto que tenga compilaciones y una política de recursos. Para eliminar un proyecto con una política de recursos y compilaciones, primero debe quitar la política de recursos y eliminar sus compilaciones.

Temas

- [Eliminar un proyecto de compilación \(consola\)](#)
- [Eliminar un proyecto de compilación \(AWS CLI\)](#)
- [Eliminar un proyecto de compilación \(SDK de AWS\)](#)

Eliminar un proyecto de compilación (consola)

1. Abra la consola de AWS CodeBuild en <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. En el panel de navegación, elija Build projects.
3. Haga una de las siguientes acciones:
 - Elija el botón de opción junto al proyecto de compilación que desea eliminar, elija Delete (Eliminar).
 - Elija el enlace del proyecto de compilación que desea eliminar y, a continuación, elija Delete.

i Note

De forma predeterminada, se muestran los 10 últimos proyectos de compilación. Para ver más proyectos de compilación, elija un valor diferente para Projects per page (Proyectos por página) o use las flechas atrás y adelante para ver los proyectos.

Eliminar un proyecto de compilación (AWS CLI)

1. Ejecute el comando `delete-project`:

```
aws codebuild delete-project --name name
```

Reemplace el siguiente marcador de posición:

- *name*: cadena necesaria. El nombre del proyecto de compilación que desea eliminar. Para obtener una lista de los proyectos de compilación disponibles, ejecute el comando `list-projects`. Para obtener más información, consulte [Ver una lista de nombres de proyectos de compilación \(AWS CLI\)](#).
2. Si el comando se ejecuta correctamente, no aparecerán datos ni errores en el resultado.

Para obtener más información sobre cómo usar la AWS CLI con AWS CodeBuild, consulte [Referencia de la línea de comandos](#).

Eliminar un proyecto de compilación (SDK de AWS)

Para obtener más información sobre cómo usar AWS CodeBuild con los SDK de AWS, consulte [Referencia de los SDK y las herramientas de AWS](#).

Uso de proyectos compartidos

El uso compartido de proyectos permite a los propietarios compartir sus proyectos de AWS CodeBuild con otras cuentas o usuarios de AWS. En este modelo, la cuenta propietaria del proyecto (propietario) comparte un proyecto con otras cuentas (consumidores). Un consumidor no puede editar ni ejecutar un proyecto.

Contenido

- [Requisitos previos para compartir proyectos](#)
- [Requisitos previos para acceder a proyectos compartidos con usted](#)
- [Servicios relacionados](#)
- [Compartir un proyecto](#)
- [Dejar de compartir un proyecto compartido](#)
- [Identificación de un proyecto compartido](#)
- [Permisos de los proyectos compartidos](#)

Requisitos previos para compartir proyectos

Para compartir un proyecto, su cuenta de AWS debe ser la propietaria. No puede compartir un proyecto que se ha compartido con usted.

Requisitos previos para acceder a proyectos compartidos con usted

Para tener acceso a un proyecto compartido, el rol de IAM de un consumidor requiere el permiso `BatchGetProjects`. Puede adjuntar la siguiente política a su rol de IAM:

```
{
  "Effect": "Allow",
  "Resource": [
    "*"
  ],
  "Action": [
    "codebuild:BatchGetProjects"
  ]
}
```

Para obtener más información, consulte [Uso de políticas basadas en la identidad para AWS CodeBuild](#).

Servicios relacionados

El uso compartido de proyectos se integra con AWS Resource Access Manager (AWS RAM), un servicio que le permite compartir sus recursos de AWS con cualquier cuenta de AWS o a través de AWS Organizations. Con AWS RAM, comparte los recursos mediante la creación de un recurso compartido que especifica los recursos y los consumidores con los que los comparte. Los consumidores pueden ser cuentas de AWS individuales, unidades organizativas en AWS Organizations o toda una organización de AWS Organizations.

Para obtener más información, consulte la Guía del usuario de [AWS RAM](#).

Compartir un proyecto

El consumidor puede usar las consolas de AWS CLI o AWS CodeBuild para ver el proyecto y las compilaciones que ha compartido. El consumidor no puede editar ni ejecutar el proyecto.

Puede añadir un proyecto a un recurso compartido existente o puede crear uno en la consola de [AWS RAM](#).

Note

No se puede eliminar un proyecto con compilaciones que se haya añadido a un recurso compartido.

Para compartir un proyecto con unidades organizativas o con toda la organización, debe habilitar el uso compartido con AWS Organizations. Para obtener más información, consulte [Habilitar el uso compartido con AWS Organizations](#) en la Guía del usuario de AWS RAM.

Puede usar la consola de AWS CodeBuild, la consola de AWS RAM o AWS CLI para compartir un proyecto de su propiedad.

Para compartir un proyecto de su propiedad (consola de CodeBuild)

1. Abra la consola de AWS CodeBuild en <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. En el panel de navegación, elija Build projects.

 Note

De forma predeterminada, solo se muestran los 10 últimos proyectos de compilación. Para ver más proyectos de compilación, elija el icono de rueda dentada y, a continuación, seleccione un valor diferente para Projects per page (Proyectos por página) o use las flechas atrás y adelante.

3. Elija el proyecto que desea compartir y, a continuación, elija Share (Compartir). Para obtener más información, consulte [Crear un recurso compartido](#) en la Guía del usuario de AWS RAM.

Para compartir un proyecto que posee (consola de AWS RAM)

Consulte [Crear un recurso compartido](#) en la Guía del usuario de AWS RAM.

Para compartir un proyecto que posee (comando AWS RAM)

Utilice el comando [create-resource-share](#).

Para compartir un proyecto de su propiedad (comando de CodeBuild)

Utilice el comando [put-resource-policy](#):

1. Cree un archivo llamado `policy.json` y copie lo siguiente en él.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
```

```

    "Principal":{
      "AWS":"<consumer-aws-account-id-or-user>"
    },
    "Action":[
      "codebuild:BatchGetProjects",
      "codebuild:BatchGetBuilds",
      "codebuild:ListBuildsForProject"],
    "Resource":"<arn-of-project-to-share>"
  ]
}

```

2. Actualice `policy.json` con el ARN del proyecto y los identificadores para compartirlo. En el ejemplo siguiente, se concede acceso de solo lectura al usuario raíz de la cuenta de AWS identificada por 123456789012.

```

{
  "Version":"2012-10-17",
  "Statement":[{
    "Effect":"Allow",
    "Principal":{
      "AWS": [
        "123456789012"
      ]
    },
    "Action":[
      "codebuild:BatchGetProjects",
      "codebuild:BatchGetBuilds",
      "codebuild:ListBuildsForProject"],
    "Resource":"arn:aws:codebuild:us-west-2:123456789012:project/my-project"
  }]
}

```

3. Ejecute el comando [put-resource-policy](#).

```
aws codebuild put-resource-policy --resource-arn <project-arn> --policy file://policy.json
```

4. Obtenga el ARN del recurso compartido de AWS RAM.

```
aws ram list-resources --resource-owner SELF --resource-arns <project-arn>
```

Esto devolverá una respuesta con un aspecto similar al siguiente:

```
{
  "resources": [
    {
      "arn": "<project-arn>",
      "type": "<type>",
      "resourceShareArn": "<resource-share-arn>",
      "creationTime": "<creation-time>",
      "lastUpdatedTime": "<last-update-time>"
    }
  ]
}
```

De la respuesta, copie el valor `<resource-share-arn>` valor para usarlo en el paso siguiente.

5. Ejecute el comando de AWS RAM [promote-resource-share-created-from-policy](#).

```
aws ram promote-resource-share-created-from-policy --resource-share-arn <resource-share-arn>
```

Dejar de compartir un proyecto compartido

Solo el propietario puede obtener acceso a un proyecto que se ha dejado de compartir, incluidas sus compilaciones. Si deja de compartir un proyecto, cualquier cuenta o usuario de AWS con el que lo haya compartido anteriormente no podrá obtener acceso al proyecto ni a sus compilaciones.

Para dejar de compartir un proyecto compartido que posee, debe quitarlo del recurso compartido. Puede usar la consola de AWS CodeBuild, la consola de AWS RAM o AWS CLI para realizar esta operación.

Para dejar de compartir un proyecto compartido que posee (consola de AWS RAM)

Consulte [Actualización de un recurso compartido](#) en la Guía del usuario de AWS RAM.

Para dejar de compartir un proyecto compartido que posee (AWS CLI)

Utilice el comando [disassociate-resource-share](#).

Para dejar de compartir un proyecto que posee (comando de CodeBuild)

Ejecute el comando [delete-resource-policy](#) y especifique el ARN del proyecto que desea dejar de compartir:

```
aws codebuild delete-resource-policy --resource-arn project-arn
```

Identificación de un proyecto compartido

Los propietarios y los consumidores pueden utilizar la AWS CLI para identificar proyectos compartidos.

Para identificar proyectos compartidos con su cuenta o usuario de AWS (AWS CLI)

Utilice el comando [lista de proyectos compartidos](#) para devolver los proyectos que se comparten con usted.

Permisos de los proyectos compartidos

Permisos de los propietarios

El propietario del proyecto puede editar el proyecto y utilizarlo para ejecutar compilaciones.

Permisos de los consumidores

Un consumidor de proyecto puede ver un proyecto y sus compilaciones, pero no puede editarlo ni usarlo para ejecutar compilaciones.

Etiquetado de proyectos en AWS CodeBuild

Una etiqueta es un atributo personalizado que usted o AWS asignan a un recurso de AWS. Cada etiqueta de AWS tiene dos partes:

- Una clave de etiqueta (por ejemplo, `CostCenter`, `Environment`, `Project` o `Secret`). Las claves de etiqueta distinguen entre mayúsculas y minúsculas.
- Un campo opcional que se denomina valor de etiqueta (por ejemplo, `111122223333` o `Production` o el nombre de un equipo). Omitir el valor de etiqueta es lo mismo que utilizar una cadena vacía. Al igual que las claves de etiqueta, los valores de etiqueta distinguen entre mayúsculas y minúsculas.

En conjunto, se conocen como pares clave-valor. Para obtener información sobre el número de etiquetas que puede tener en un proyecto y las restricciones en las claves y valores de las etiquetas, consulte [Etiquetas](#).

Las etiquetas le ayudan a identificar y organizar los recursos de AWS. Muchos servicios de AWS admiten el etiquetado, por lo que puede asignar la misma etiqueta a los recursos de diferentes servicios para indicar que los recursos están relacionados. Por ejemplo, es posible asignar la misma etiqueta a un proyecto de CodeBuild asignado a un bucket de S3. Para obtener más información sobre el uso de etiquetas, consulte [Prácticas recomendadas de etiquetado](#).

En CodeBuild, los recursos principales son el proyecto y el grupo de informes. Puede utilizar la consola de CodeBuild, la AWS CLI, las API de CodeBuild o los SDK de AWS para agregar, administrar y quitar etiquetas de un proyecto. Además de utilizar etiquetas para identificar, organizar y realizar el seguimiento de su proyecto, puede utilizarlas en las políticas de IAM para ayudar a controlar quién puede consultar el proyecto e interactuar con él. Para ver ejemplos de políticas de acceso basadas en etiquetas, consulte [Uso de etiquetas para controlar el acceso a los recursos de AWS CodeBuild](#).

Temas

- [Agregar una etiqueta a un proyecto](#)
- [Consultar etiquetas de un proyecto](#)
- [Editar etiquetas de un proyecto](#)
- [Quitar una etiqueta de un proyecto](#)

Agregar una etiqueta a un proyecto

Agregar etiquetas a un proyecto puede ayudarle a identificar y organizar sus recursos de AWS y a administrar el acceso a ellos. En primer lugar, agregue una o varias etiquetas (pares de clave-valor) a un proyecto. Tenga en cuenta que hay límites en el número de etiquetas que puede tener en un proyecto. Existen restricciones sobre los caracteres que se pueden utilizar en los campos de clave y valor. Para obtener más información, consulte [Etiquetas](#). Cuando tenga las etiquetas, puede crear políticas de IAM para administrar el acceso al proyecto en función de estas etiquetas. Puede usar la consola de CodeBuild o la AWS CLI para agregar etiquetas a un proyecto.

Important

Antes de agregar una etiqueta a un proyecto, asegúrese de revisar las políticas de IAM que es posible que utilicen etiquetas para controlar el acceso a recursos como los proyectos de compilación. Para ver ejemplos de políticas de acceso basadas en etiquetas, consulte [Uso de etiquetas para controlar el acceso a los recursos de AWS CodeBuild](#).

Para obtener más información sobre cómo agregar etiquetas a un proyecto al crearlo, consulte [Agregar una etiqueta a un proyecto \(consola\)](#).

Temas

- [Agregar una etiqueta a un proyecto \(consola\)](#)
- [Agregar una etiqueta a un proyecto \(AWS CLI\)](#)

Agregar una etiqueta a un proyecto (consola)

Puede usar la consola de CodeBuild para agregar una o más etiquetas a un proyecto de CodeBuild.

1. Abra la consola de CodeBuild en <https://console.aws.amazon.com/codebuild/>.
2. En Proyectos de compilación, elija el nombre del proyecto al que desea agregar etiquetas.
3. En el panel de navegación, seleccione Settings (Configuración). Elija Etiquetas de proyecto de compilación.
4. Si no se ha agregado ninguna etiqueta al proyecto, elija Agregar etiqueta. De lo contrario, elija Edit (Editar) y, a continuación, seleccione Add tag (Añadir etiqueta).
5. En Key (Clave), escriba un nombre para la etiqueta. Puede añadir un valor opcional para la etiqueta en Value (Valor).
6. (Opcional) Para añadir otra etiqueta, vuelva a elegir Add tag (Añadir etiqueta).
7. Cuando haya terminado de añadir etiquetas, seleccione Submit (Enviar).

Agregar una etiqueta a un proyecto (AWS CLI)

Para agregar una etiqueta a un proyecto al crearlo, consulte [Crear un proyecto de compilación \(AWS CLI\)](#). En `create-project.json`, agregue sus etiquetas.

En estos pasos, se presupone que ya ha instalado una versión reciente de la AWS CLI o que la ha actualizado a la versión actual. Para obtener más información, consulte [Instalación de AWS Command Line Interface](#).

Si se ejecuta correctamente, este comando no devuelve nada.

Consultar etiquetas de un proyecto

Las etiquetas pueden ayudarle a identificar y organizar sus recursos de AWS y administrar el acceso a ellos. Para obtener más información sobre el uso de etiquetas, consulte el documento técnico

[Prácticas recomendadas de etiquetado](#). Para ver ejemplos de políticas de acceso basadas en etiquetas, consulte [Uso de etiquetas para controlar el acceso a los recursos de AWS CodeBuild](#).

Consultar etiquetas de un proyecto (consola)

Puede usar la consola de CodeBuild para consultar las etiquetas asociadas con un proyecto de CodeBuild.

1. Abra la consola de CodeBuild en <https://console.aws.amazon.com/codebuild/>.
2. En Proyectos de compilación, elija el nombre del proyecto en el que desea consultar las etiquetas.
3. En el panel de navegación, seleccione Settings (Configuración). Elija Etiquetas de proyecto de compilación.

Consultar etiquetas de un proyecto (AWS CLI)

Para consultar las etiquetas de un proyecto de compilación, ejecute el siguiente comando. Utilice el nombre del proyecto para el parámetro `--names`.

```
aws codebuild batch-get-projects --names your-project-name
```

Si tiene éxito, este comando devuelve información con formato JSON sobre su proyecto de compilación que incluye algo como lo siguiente:

```
{
  "tags": {
    "Status": "Secret",
    "Team": "JanesProject"
  }
}
```

Si el proyecto no tiene etiquetas, la sección `tags` está vacía:

```
"tags": []
```

Editar etiquetas de un proyecto

Puede cambiar el valor de una etiqueta asociada con un proyecto. También puede cambiar el nombre de la clave, lo que equivale a eliminar la etiqueta actual y añadir otra distinta con el nuevo

nombre y el mismo valor que la otra clave. Tenga en cuenta que hay límites en los caracteres que puede utilizar en los campos de clave y valor. Para obtener más información, consulte [Etiquetas](#).

Important

La edición de etiquetas de un proyecto puede afectar el acceso a ese proyecto. Antes de editar el nombre (clave) o valor de una etiqueta de un proyecto, asegúrese de revisar cualquier política de IAM que pueda usar la clave o el valor de una etiqueta para controlar el acceso a recursos como los proyectos de compilación. Para ver ejemplos de políticas de acceso basadas en etiquetas, consulte [Uso de etiquetas para controlar el acceso a los recursos de AWS CodeBuild](#).

Editar una etiqueta de un proyecto (consola)

Es posible usar la consola de CodeBuild para editar las etiquetas asociadas con un proyecto de CodeBuild.

1. Abra la consola de CodeBuild en <https://console.aws.amazon.com/codebuild/>.
2. En Proyectos de compilación, elija el nombre del proyecto en el que desea editar las etiquetas.
3. En el panel de navegación, seleccione Settings (Configuración). Elija Etiquetas de proyecto de compilación.
4. Elija Edit (Editar).
5. Haga una de las siguientes acciones:
 - Para cambiar la etiqueta, escriba un nuevo nombre en Key (Clave). Cambiar el nombre de la etiqueta equivale a quitar una etiqueta y añadir otra nueva con el nuevo nombre de clave.
 - Para cambiar el valor de una etiqueta, escriba un nuevo valor. Si desea cambiar el valor por nada, elimine el valor actual y deje el campo en blanco.
6. Cuando haya terminado de editar las etiquetas, seleccione Submit (Enviar).

Editar etiquetas de un proyecto (AWS CLI)

Para agregar, cambiar o eliminar etiquetas de un proyecto de compilación, consulte [Cambiar la configuración de un proyecto de compilación \(AWS CLI\)](#). Actualice la sección tags de los datos con formato JSON que utiliza para actualizar el proyecto.

Quitar una etiqueta de un proyecto

Puede quitar una o varias etiquetas asociadas con un proyecto. La eliminación de una etiqueta no la elimina de otros recursos de AWS que están asociados con esa etiqueta.

Important

Quitar las etiquetas de un proyecto puede afectar al acceso a ese proyecto. Antes de quitar una etiqueta de un proyecto, asegúrese de revisar cualquier política de IAM que pueda utilizar la clave o el valor de una etiqueta para controlar el acceso a recursos como los proyectos de compilación. Para ver ejemplos de políticas de acceso basadas en etiquetas, consulte [Uso de etiquetas para controlar el acceso a los recursos de AWS CodeBuild](#).

Quitar una etiqueta de un proyecto (consola)

Puede usar la consola de CodeBuild para eliminar la asociación entre una etiqueta y un proyecto de CodeBuild.

1. Abra la consola de CodeBuild en <https://console.aws.amazon.com/codebuild/>.
2. En Proyectos de compilación, elija el nombre del proyecto en el que desea quitar las etiquetas.
3. En el panel de navegación, seleccione Settings (Configuración). Elija Etiquetas de proyecto de compilación.
4. Elija Edit (Editar).
5. Busque la etiqueta que desea eliminar y, a continuación, seleccione Remove tag (Eliminar etiqueta).
6. Cuando haya terminado de eliminar las etiquetas, seleccione Submit (Enviar).

Quitar una etiqueta de un proyecto (AWS CLI)

Para eliminar una o varias etiquetas de un proyecto de compilación, consulte [Cambiar la configuración de un proyecto de compilación \(AWS CLI\)](#). Actualice la sección `tags` de los datos con formato JSON con una lista actualizada de etiquetas que no contenga las que desea eliminar. Si desea eliminar todas las etiquetas, actualice la sección `tags` para:

```
"tags: []"
```

Note

Si elimina un proyecto de compilación de CodeBuild, todas las asociaciones de etiquetas se quitan del proyecto de compilación eliminado. No es necesario quitar las etiquetas antes de eliminar un proyecto de compilación.

Batch se integra AWS CodeBuild

Se puede utilizar AWS CodeBuild para ejecutar compilaciones simultáneas y coordinadas de un proyecto con compilaciones por lotes.

Temas

- [Rol de seguridad](#)
- [Tipos de compilación por lotes](#)
- [Modo de informe por lotes](#)
- [Más información](#)

Rol de seguridad

Las compilaciones por lotes introducen un nuevo rol de seguridad en la configuración por lotes. Este nuevo rol es obligatorio `StartBuild`, `StopBuild` ya que CodeBuild debes poder ejecutar las compilaciones y `RetryBuild` acciones en tu nombre para ejecutar compilaciones como parte de un lote. Los clientes deberían utilizar un nuevo rol, y no el mismo rol que utilizan para su compilación, por dos motivos:

- Al asignar los permisos de `StartBuild`, `StopBuild` y `RetryBuild` en el rol de compilación, una sola compilación podría iniciar más compilaciones a través de la especificación de compilación.
- CodeBuild Las compilaciones por lotes proporcionan restricciones que restringen la cantidad de compilaciones y tipos de procesamiento que se pueden usar para las compilaciones del lote. Si la función de compilación tiene estos permisos, es posible que las propias compilaciones puedan eludir estas restricciones.

Tipos de compilación por lotes

CodeBuild admite los siguientes tipos de compilación por lotes:

Tipos de compilación por lotes

- [Grafo de compilación](#)
- [Lista de compilación](#)
- [Matriz de compilación](#)

Grafo de compilación

Un grafo de compilación define un conjunto de tareas que dependen de otras tareas del lote.

En el siguiente ejemplo, se define un grafo de compilación que crea una cadena de dependencias.

```
batch:
  fast-fail: false
  build-graph:
    - identifier: build1
      env:
        variables:
          BUILD_ID: build1
      ignore-failure: false
    - identifier: build2
      buildspec: build2.yml
      env:
        variables:
          BUILD_ID: build2
      depend-on:
        - build1
    - identifier: build3
      env:
        variables:
          BUILD_ID: build3
      depend-on:
        - build2
```

En este ejemplo:

- `build1` se ejecuta primero porque no tiene dependencias.
- `build2` depende de `build1`, por lo que `build2` se ejecuta después de completar `build1`.

- build3 depende de build2, por lo que build3 se ejecuta después de completar build2.

Para obtener más información acerca de la sintaxis del grafo de especificación de compilación, consulte [batch/build-graph](#).

Lista de compilación

Una lista de compilación define una serie de tareas que se ejecutan en paralelo.

En el ejemplo siguiente se define una lista de compilación. Las compilaciones build1 y build2 se ejecutarán en paralelo.

```
batch:
  fast-fail: false
  build-list:
    - identifier: build1
      env:
        variables:
          BUILD_ID: build1
      ignore-failure: false
    - identifier: build2
      buildspec: build2.yml
      env:
        variables:
          BUILD_ID: build2
      ignore-failure: true
```

Para obtener más información acerca de la sintaxis de la lista de especificación de compilación, consulte [batch/build-list](#).

Matriz de compilación

Una matriz de compilación define tareas con distintas configuraciones que se ejecutan en paralelo. CodeBuild crea una compilación independiente para cada combinación de configuración posible.

El ejemplo siguiente muestra una matriz de compilación con dos archivos de especificaciones de compilación y tres valores de una variable de entorno.

```
batch:
  build-matrix:
    static:
```

```
ignore-failure: false
dynamic:
  buildspec:
    - matrix1.yml
    - matrix2.yml
  env:
    variables:
      MY_VAR:
        - VALUE1
        - VALUE2
        - VALUE3
```

En este ejemplo, CodeBuild crea seis compilaciones:

- `matrix1.yml` por `$MY_VAR=VALUE1`
- `matrix1.yml` por `$MY_VAR=VALUE2`
- `matrix1.yml` por `$MY_VAR=VALUE3`
- `matrix2.yml` por `$MY_VAR=VALUE1`
- `matrix2.yml` por `$MY_VAR=VALUE2`
- `matrix2.yml` por `$MY_VAR=VALUE3`

Cada compilación tendrá la configuración siguiente:

- `ignore-failure` es `false`.
- `env/type` es `LINUX_CONTAINER`.
- `env/image` es `aws/codebuild/amazonlinux2-x86_64-standard:4.0`.
- `env/privileged-mode` es `true`.

Estas compilaciones se ejecutan en paralelo.

Para obtener más información acerca de la sintaxis de la matriz de especificación de compilación, consulte [batch/build-matrix](#).

Modo de informe por lotes

Si el proveedor de origen de tu proyecto es Bitbucket o GitHub Enterprise y tu proyecto está configurado para informar de los estados de compilación al proveedor de origen, puedes seleccionar cómo quieres que se envíen los estados de compilación por lotes al proveedor de origen. GitHub Es

posible seleccionar que los estados se envíen como un único informe de estado agregado para el lote o que se notifique el estado de cada compilación de lote por separado.

Para obtener más información, consulte los temas siguientes:

- [Configuración por lotes \(crear\)](#)
- [Configuración por lotes \(actualizar\)](#)

Más información

Para obtener más información, consulte los temas siguientes:

- [Referencia de especificaciones de compilación para compilación por lotes](#)
- [Configuración de lotes](#)
- [Ejecutar una compilación por lotes \(AWS CLI\)](#)
- [Detener una compilación por lotes en AWS CodeBuild](#)

GitHub Action Runner en AWS CodeBuild

Una GitHub acción es una acción que se ha desarrollado específicamente para su uso con GitHub flujos de trabajo. Para obtener más información sobre GitHub las acciones, consulte la documentación sobre [GitHub las acciones](#).

Hay dos formas de utilizar GitHub las acciones con CodeBuild:

- Puedes configurar tu proyecto para configurar ejecutores de GitHub Actions autohospedados en CodeBuild contenedores para procesar tus trabajos de flujo de trabajo de GitHub Actions.
- Puedes usar un ejecutor CodeBuild de acciones gestionado para ejecutar GitHub acciones en él. CodeBuild

Puedes elegir configurar GitHub Actions Runners autohospedados en. CodeBuild Esto implica configurar un webhook con tu CodeBuild proyecto y actualizar el flujo de trabajo de GitHub Actions (YAML) para usar ejecutores autohospedados alojados en máquinas. CodeBuild Esto permite que tus trabajos de flujo de trabajo de GitHub Actions se integren de forma nativa. AWS

También puedes optar por utilizar un corredor de acciones CodeBuild gestionado para ejecutar GitHub Actions en CodeBuild él. Esto implica añadir steps a tu especificación de compilación la

sintaxis GitHub Actions, que se ejecuta en una fase diferente a la de los comandos. CodeBuild Esto permite que tus GitHub acciones se integren con CodeBuild funciones, como el almacenamiento en caché de dependencias y las compilaciones por lotes.

Temas

- [Configura ejecutores de GitHub Actions autohospedados en AWS CodeBuild](#)
- [Usa la sintaxis GitHub Actions en una especificación de compilación en AWS CodeBuild](#)

Configura ejecutores de GitHub Actions autohospedados en AWS CodeBuild

Puedes configurar tu proyecto para configurar ejecutores de GitHub Actions autohospedados en CodeBuild contenedores para procesar tus trabajos de flujo de trabajo de GitHub Actions. Para ello, puedes configurar un webhook con tu CodeBuild proyecto y actualizar el flujo de trabajo de GitHub Actions (YAML) para que utilice ejecutores autohospedados y alojados en máquinas. CodeBuild Para obtener más información, consulta [Acerca](#) de los ejecutores autohospedados.

Los pasos generales para configurar un CodeBuild proyecto para ejecutar tareas de GitHub Actions son los siguientes:

1. Si aún no lo has hecho, crea un token de acceso personal o conéctate a una aplicación de OAuth a la que conectar tu proyecto. GitHub
2. Ve a la CodeBuild consola y crea un CodeBuild proyecto con un webhook y configura tus filtros de webhook.
3. Actualiza el flujo de trabajo de GitHub Actions en YAML GitHub para configurar tu entorno de compilación.

Para obtener un procedimiento más detallado, consulte [Tutorial: Configurar un ejecutor de acciones CodeBuild autohospedado GitHub](#) .

Esta función permite que sus trabajos de flujo de trabajo de GitHub Actions se integren de forma nativa AWS, lo que proporciona seguridad y comodidad a través de funciones como IAM AWS CloudTrail, AWS Secrets Manager integración y Amazon VPC. Puede acceder a los tipos de instancias más recientes, incluidas las instancias basadas en ARM.

Temas

- [Tutorial: Configurar un ejecutor de acciones CodeBuild autohospedado GitHub](#)
- [Acerca de CodeBuild -hosted GitHub Actions Runner](#)

Tutorial: Configurar un ejecutor de acciones CodeBuild autohospedado GitHub

Este tutorial le muestra cómo configurar sus CodeBuild proyectos para ejecutar trabajos de GitHub Actions.

Requisitos previos

Para completar este tutorial, primero debe:

- Conéctate con una aplicación OAuth o crea un token de acceso personal. Si quieres conectarte con una aplicación OAuth, debes usar la CodeBuild consola para hacerlo. [Si quieres crear un token de acceso personal, puedes usar la CodeBuild consola o la API. ImportSourceCredentials](#) Para obtener más instrucciones, consulte [GitHub y un token de acceso a GitHub Enterprise Server](#).
- Conéctate CodeBuild a tu GitHub cuenta. Para ello, puede hacer una de las acciones siguientes:
 - Puedes añadirlo GitHub como proveedor de fuentes en la consola. Puedes conectarte con una aplicación OAuth o con un token de acceso personal. Para ver instrucciones, consulte [Conéctese GitHub con un token de acceso \(consola\)](#).
 - [Puedes importar tus GitHub credenciales a través de la ImportSourceCredentials API](#). Esto solo se puede hacer con un token de acceso personal. Si te conectas mediante una aplicación OAuth, debes conectarte mediante la consola en su lugar. Para ver instrucciones, consulte [Conéctese GitHub con un token de acceso \(CLI\)](#).

Note

Esto solo es necesario si no te has conectado a tu GitHub cuenta.

Paso 1: Crea un CodeBuild proyecto con un webhook

En este paso, crearás un CodeBuild proyecto con un webhook y lo revisarás en la GitHub consola.

Para crear un CodeBuild proyecto con un webhook

1. Abre la AWS CodeBuild consola en <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Cree un proyecto de compilación. Para obtener más información, consulte [Creación de un proyecto de compilación \(consola\)](#) y [Ejecutar una compilación \(consola\)](#).
 - En Source (Código fuente):
 - En Proveedor de código fuente, elija GitHub.

- En Repositorio, selecciona Repositorio en mi GitHub cuenta.
- En URL de repositorio, especifique **`https://github.com/user-name/repository-name`**
- En los eventos de webhook de fuente principal:
 - En el caso de Webhook (opcional), selecciona Reconstruir cada vez que se introduzca un cambio de código en este repositorio.
 - En Tipo de evento, selecciona `WORKFLOW_JOB_QUEUED`. Una vez activado, las compilaciones solo se activarán mediante eventos de tareas del flujo de trabajo de Actions.

 Note

CodeBuild solo procesará los eventos de los trabajos del flujo de trabajo de GitHub Actions si un webhook tiene grupos de filtros que contengan el filtro de eventos `WORKFLOW_JOB_QUEUED`.

Filter group 1

Remove filter group

Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

WORKFLOW_JOB_QUEUED X

▶ Start a build under these conditions - *optional*

▶ Don't start a build under these conditions - *optional*

- En Environment (Entorno):
 - Elija una imagen de entorno y un proceso de cómputo compatibles. Ten en cuenta que tienes la opción de anular la configuración de la imagen y la instancia mediante una etiqueta en el flujo de trabajo de GitHub Actions (YAML). Para obtener más información, consulte [Paso 2: actualiza el flujo de trabajo de GitHub Actions \(YAML\)](#).
- En Buildspec:
 - Ten en cuenta que se ignorará tu Buildspec. En su lugar, lo CodeBuild anulará para usar comandos que configurarán el ejecutor autohospedado. La responsabilidad principal de este

proyecto es configurar un ejecutor autohospedado para ejecutar los trabajos del flujo de trabajo CodeBuild de GitHub Actions.

3. Continúe con los valores predeterminados y, a continuación, elija Crear proyecto de compilación.
4. Abre la GitHub consola en <https://github.com/user-name/repository-name/settings/hooks> para comprobar que se ha creado un webhook y que está habilitado para entregar eventos de Workflow Jobs.

Paso 2: actualiza el flujo de trabajo de GitHub Actions (YAML)

En este paso, actualizarás el archivo YAML del flujo de trabajo de GitHub Actions [GitHub](#) para configurar tu entorno de compilación y utilizar los ejecutores autohospedados de GitHub Actions. CodeBuild Para obtener más información, consulta [Cómo usar etiquetas con ejecutores autohospedados](#).

Actualiza el flujo de trabajo de GitHub Actions (YAML)

Ve a la [runs-on](#) configuración de tu flujo de trabajo de GitHub Actions (YAML) [GitHub](#) actualízala para configurar tu entorno de compilación. Para ello, puede hacer una de las acciones siguientes:

- Puedes especificar el nombre del proyecto y el ID de ejecución, en cuyo caso la compilación utilizará la configuración de proyecto existente para el cálculo, la imagen, la versión de la imagen y el tamaño de la instancia. El nombre del proyecto es necesario para vincular la configuración AWS relacionada de tu trabajo de GitHub Actions a un CodeBuild proyecto específico. Al incluir el nombre del proyecto en el YAML, CodeBuild se pueden invocar trabajos con la configuración de proyecto correcta. Al proporcionar el ID de ejecución, CodeBuild asignará tu compilación a ejecuciones de flujo de trabajo específicas y detendrá la compilación cuando se cancele la ejecución del flujo de trabajo. Para obtener más información, consulta [githubel contexto](#).

```
runs-on: codebuild-<project-name>-${{ github.run_id }}-${{ github.run_attempt }}
```

Note

Asegúrese de que <project-name> coincide con el nombre del proyecto que creó en el paso anterior. Si no coincide, no CodeBuild procesará el webhook y es posible que el flujo de trabajo de GitHub Actions se bloquee.

El siguiente es un ejemplo de un flujo de trabajo YAML de GitHub Actions:

```
name: Hello World
on: [push]
jobs:
  Hello-World-Job:
    runs-on: codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}
    steps:
      - run: echo "Hello World!"
```

- También puedes anular la imagen y el tipo de procesamiento en la etiqueta. Esto anulará la configuración del entorno del proyecto. Para anular la configuración del entorno para una compilación informática de Amazon EC2, utilice la siguiente sintaxis:

```
runs-on: codebuild-<project-name>-${{ github.run_id }}-
${{ github.run_attempt }}-<image>-<image-version>-<instance-size>
```

Para anular la configuración de su entorno para una compilación de procesamiento Lambda, utilice la siguiente sintaxis:

```
runs-on: codebuild-<project-name>-${{ github.run_id }}-
${{ github.run_attempt }}-<environment-type>-<runtime-version>-<instance-size>
```

A continuación, se muestra un ejemplo de un flujo de trabajo YAML de GitHub Actions:

```
name: Hello World
on: [push]
jobs:
  Hello-World-Job:
    runs-on: codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}-
    arm-3.0-small
    steps:
      - run: echo "Hello World!"
```

Note

Si una dependencia proporcionada por los GitHub ejecutores alojados no está disponible en el CodeBuild entorno, puedes instalarla mediante GitHub Actions en la ejecución de tu flujo

de trabajo. Por ejemplo, puedes usar la [setup-python](#) acción para instalar Python en tu entorno de compilación.

Imágenes de cómputo compatibles

En la etiqueta, puede anular la configuración del entorno de Amazon EC2 utilizando los valores de las tres primeras columnas. CodeBuild proporciona las siguientes imágenes informáticas de Amazon EC2:

Imagen	Versión de imagen	Tamaño de instancia	Plataforma	Identificador de imagen	Definición
linux	4.0	small medium large xlarge	Amazon Linux 2	aws/codebuild/amazonlinux2-x86_64-standard:4.0	al2/standard/4.0
linux	5.0	2xlarge gpu_small gpu_large	Amazon Linux 2023	aws/codebuild/amazonlinux2-x86_64-standard:5.0	al2/standard/5.0
arm	2.0	small large	Amazon Linux 2	aws/codebuild/amazonlinux2-aarch64-standard:2.0	al2/aarch64/standard/2.0
arm	3.0		Amazon Linux 2023	aws/codebuild/amazonlinux2-aarch64-s	al2/aarch64/standard/3.0

Imagen	Versión de imagen	Tamaño de instancia	Plataforma	Identificador de imagen	Definición
ubuntu	5.0	small medium	Ubuntu 20.04	aws/codebuild/standard:3.0 aws/codebuild/standard:5.0	ubuntu/standard/5.0
ubuntu	6.0	large xlarge	Ubuntu 22.04	aws/codebuild/standard:6.0	ubuntu/standard/6.0
ubuntu	7.0	2xlarge gpu_small gpu_large	Ubuntu 22.04	aws/codebuild/standard:7.0	ubuntu/standard/7.0
windows	1.0	medium large	Windows Server Core 2019	aws/codebuild/windows-base:2019-1.0	N/A
windows	2.0		Windows Server Core 2019	aws/codebuild/windows-base:2019-2.0	N/A
windows	3.0		Windows Server Core 2019	aws/codebuild/windows-base:2019-3.0	N/A

Además, puede anular la configuración del entorno Lambda mediante los siguientes valores. Para obtener más información sobre la computación CodeBuild Lambda, consulte Trabajar con. [Trabajar con AWS Lambda computación en AWS CodeBuild](#) CodeBuild admite las siguientes imágenes de procesamiento Lambda:

Tipo de entorno	Versión de runtime	Tamaño de instancia			
linux-lambda	dotnet6	1GB			
	go1.21	2GB			
arm-lambda	corretto1	4GB			
	1	8GB			
	corretto1	10GB			
	7				
	corretto2				
	1				
	nodejs18				
	nodejs20				
	python3.1				
1					
python3.1					
2					
ruby3.2					

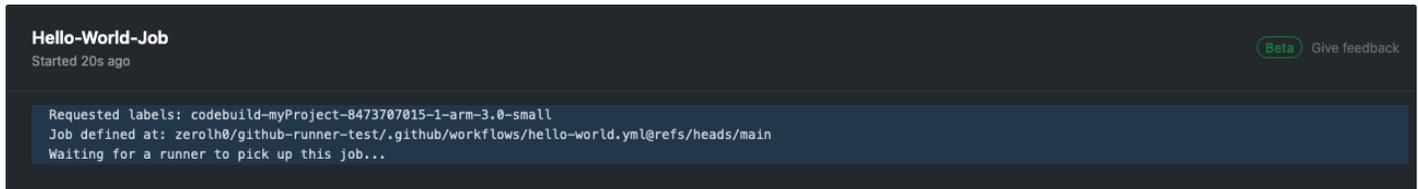
Para obtener más información, consulte [Modos y tipos de computación del entorno de compilación y Imágenes de Docker proporcionadas por CodeBuild](#).

Paso 3: Revise los resultados

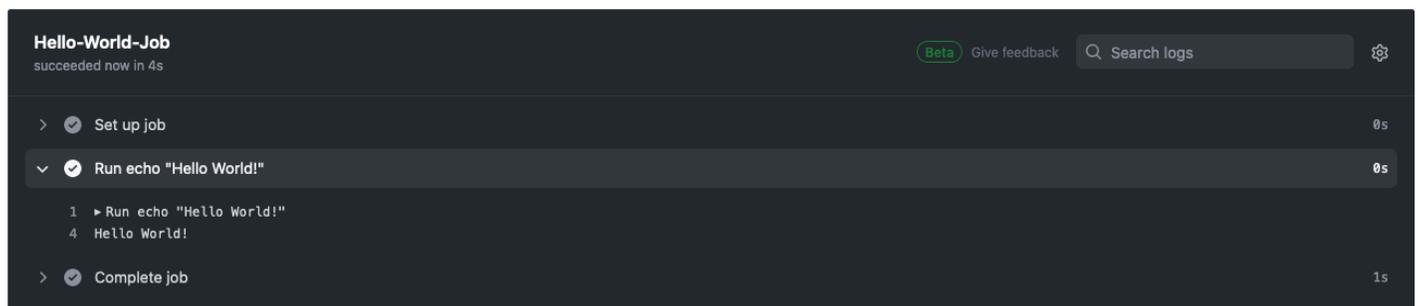
Siempre que se ejecute un flujo de trabajo de GitHub Actions, CodeBuild recibirá los eventos del trabajo del flujo de trabajo a través del webhook. Para cada trabajo del flujo de trabajo, CodeBuild inicia una compilación para ejecutar un ejecutor efímero GitHub de Actions. El ejecutor es responsable de ejecutar un único trabajo de flujo de trabajo. Una vez que se complete el trabajo, el ejecutor y el proceso de creación asociado finalizarán inmediatamente.

Para ver los registros de trabajos de tu flujo de trabajo, navega hasta tu repositorio, selecciona Acciones, elige el flujo de trabajo que desees y, a continuación, elige el trabajo específico para el que quieres revisar los registros. GitHub

Puedes revisar las etiquetas solicitadas en el registro mientras el trabajo está esperando a que lo recoja un agente autohospedado. CodeBuild



Una vez que se haya completado el trabajo, podrá ver el registro del trabajo.



Acerca de CodeBuild -hosted GitHub Actions Runner

¿Cuándo debo incluir las modificaciones de imagen e instancia en la etiqueta?

Puedes incluir las modificaciones de imagen e instancia en la etiqueta para especificar un entorno de compilación diferente para cada uno de tus trabajos del flujo de trabajo de GitHub Actions. Esto se puede hacer sin necesidad de crear varios CodeBuild proyectos o webhooks. Por ejemplo, esto resulta útil cuando necesitas usar una [matriz para tus trabajos de flujo de trabajo](#).

```
name: Hello World
on: [push]
jobs:
  Hello-World-Job:
    runs-on: codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}-
    ${{ matrix.os }}
    strategy:
      matrix:
        os: [arm-3.0-small, a12-5.0-large]
    steps:
      - run: echo "Hello World!"
```

Note

Las comillas pueden ser obligatorias si `runs-on` tiene varias etiquetas que contienen el contexto de GitHub las acciones.

¿Puedo AWS CloudFormation utilizarla para esta función?

Sí, puedes incluir un grupo de filtros en tu AWS CloudFormation plantilla que especifique un filtro de eventos de trabajo del flujo de trabajo de GitHub Actions en el webhook de tu proyecto.

```
Triggers:
  Webhook: true
  FilterGroups:
    - - Type: EVENT
      Pattern: WORKFLOW_JOB_QUEUED
```

Para obtener más información, consulte [Filtra los eventos de GitHub webhook \(AWS CloudFormation\)](#).

Si necesitas ayuda para configurar las credenciales del proyecto en tu AWS CloudFormation plantilla, consulta [AWS::CodeBuild::SourceCredential](#) la Guía del AWS CloudFormation usuario para obtener más información.

¿Qué regiones admiten el uso de un ejecutor CodeBuild de GitHub Actions alojado?

CodeBuildLos ejecutores de GitHub Actions alojados en él son compatibles en todas las CodeBuild regiones. Para obtener más información sobre Regiones de AWS dónde CodeBuild está disponible, consulta [AWS Servicios por región](#).

¿Qué plataformas admiten el uso de un CodeBuild ejecutor de GitHub acciones alojado?

CodeBuild-Los ejecutores de GitHub Actions alojados son compatibles tanto con Amazon EC2 [AWS Lambda](#) como con Compute. Puede utilizar las siguientes plataformas: Amazon Linux 2, Amazon Linux 2023, Ubuntu y Windows Server Core 2019. Para obtener más información, consulte [Imágenes de computación de EC2](#) y [Imágenes de computación de Lambda](#).

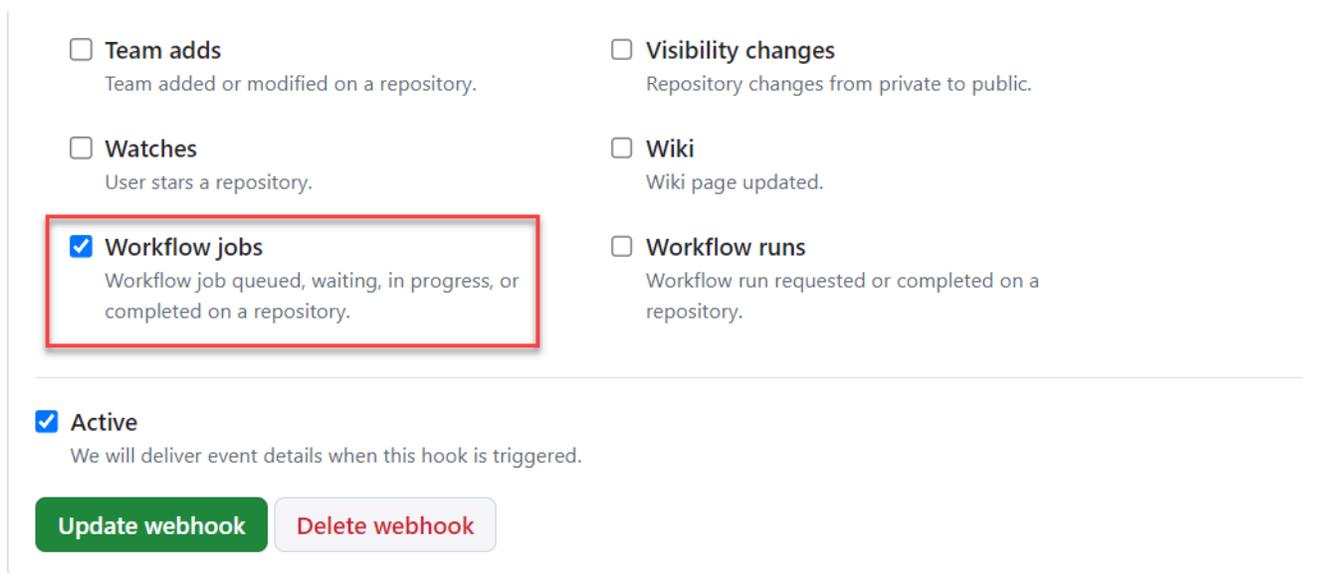
Solución de problemas: ¿Cómo soluciono los problemas si el webhook no funciona?

Problema: tu webhook no funciona o tu flujo de trabajo está pendiente. GitHub

Causa posible: es posible que el evento de trabajos de flujo de trabajo de tu webhook no esté activando una compilación. Revisa los registros de respuestas para ver la respuesta o el mensaje de error.

Solución recomendada: para depurar este error, siga las instrucciones siguientes.

1. Abre la GitHub consola en <https://github.com/user-name/repository-name/settings/hooks> para ver la configuración de webhook de tu repositorio. En esta página, verás un webhook que se creó para tu repositorio.
2. Selecciona Editar y comprueba que el webhook está habilitado para publicar eventos de Workflow Jobs.

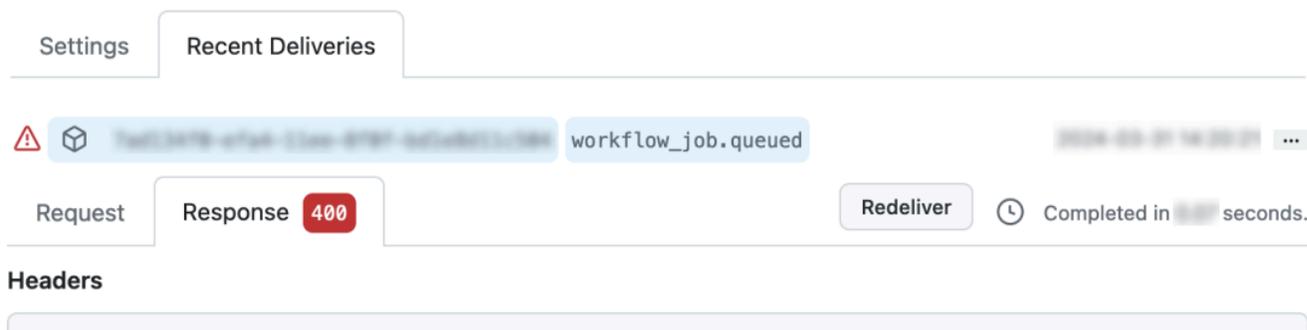


The screenshot shows the GitHub webhook configuration interface. The 'Workflow jobs' checkbox is checked and highlighted with a red box. Below it, the 'Active' checkbox is also checked. At the bottom, there are two buttons: 'Update webhook' (green) and 'Delete webhook' (red).

<input type="checkbox"/> Team adds Team added or modified on a repository.	<input type="checkbox"/> Visibility changes Repository changes from private to public.
<input type="checkbox"/> Watches User stars a repository.	<input type="checkbox"/> Wiki Wiki page updated.
<input checked="" type="checkbox"/> Workflow jobs Workflow job queued, waiting, in progress, or completed on a repository.	<input type="checkbox"/> Workflow runs Workflow run requested or completed on a repository.

Active
We will deliver event details when this hook is triggered.

3. Vaya a la pestaña Entregas recientes, busque el `workflow_job.queued` evento correspondiente y amplíe el evento.
4. Revisa el campo de etiquetas de la carga útil y asegúrate de que está según lo esperado.
5. Por último, revisa la pestaña Respuesta, ya que contiene la respuesta o el mensaje de CodeBuild error devuelto.



The screenshot shows the 'Recent Deliveries' tab in the GitHub webhook settings. The event type is 'workflow_job.queued'. The response status is '400'. There is a 'Redeliver' button and a timer showing 'Completed in 0:00 seconds'.

Settings | Recent Deliveries

Completed in 0:00 seconds.

Headers

Usa la sintaxis GitHub Actions en una especificación de compilación en AWS CodeBuild

Puedes usar un corredor de acciones CodeBuild gestionado para ejecutar acciones en él. GitHub CodeBuild Esto es posible añadiendo `steps` a cualquier [fase](#) del archivo de especificaciones de compilación.

CodeBuild Las especificaciones de compilación admiten una lista de pasos de GitHub acción secuenciales que se ejecutan en una fase separada de los comandos. CodeBuild Estas GitHub acciones se integran con CodeBuild las funciones existentes, que incluyen el almacenamiento en caché de dependencias, las compilaciones por lotes, el acceso a AWS Secrets Manager, etc.

Temas

- [¿Cómo puedo empezar a usar una GitHub acción en mi especificación de compilación?](#)
- [¿Qué GitHub acciones puedo usar en mi especificación de compilación?](#)
- [¿Puedo usar otros proveedores de código fuente además de GitHub usar GitHub Actions en mi especificación de compilación?](#)
- [¿Por qué tengo que conectarme a GitHub un proveedor de código fuente para poder usar GitHub Actions en mi especificación de compilación?](#)
- [¿Cuánto cuesta usar GitHub Actions en mi especificación de compilación?](#)
- [¿Qué regiones admiten el uso de GitHub Actions en mi especificación de compilación?](#)
- [Prácticas recomendadas para usar GitHub Actions en tus especificaciones de compilación](#)
- [Limitaciones del uso de GitHub acciones en tus especificaciones de compilación en CodeBuild](#)
- [GitHub Referencia de especificaciones de compilación de Action Runner](#)
- [GitHub Ejemplos de sintaxis de acciones con AWS CodeBuild](#)

¿Cómo puedo empezar a usar una GitHub acción en mi especificación de compilación?

Los pasos generales para usar una GitHub acción en tu especificación de compilación son los siguientes:

1. Si aún no lo has hecho, conecta tu proyecto a. GitHub

Para ello, puede hacer una de las acciones siguientes:

- Puedes añadirlo GitHub como proveedor de fuentes en la consola. Para obtener más información, consulte [Conéctese GitHub con un token de acceso \(consola\)](#).

- Puede importar sus GitHub credenciales a través de la [CodeBuild API](#). Para obtener más información, consulte [Conéctese GitHub con un token de acceso \(CLI\)](#).

 Note

Esto solo debe hacerse si no te has conectado GitHub a otro proyecto.

2. En las especificaciones de construcción de tu proyecto, puedes añadir `steps`, cada una de las cuales hace referencia a una acción. Esto se puede editar en la CodeBuild consola o en tu repositorio de código fuente. Cada fase de compilación admite una lista de comandos o una lista de pasos, pero no es posible utilizar ambas cosas en la misma fase. Para obtener más información, consulte [Usa la sintaxis GitHub Actions en una especificación de compilación en AWS CodeBuild](#).

¿Qué GitHub acciones puedo usar en mi especificación de compilación?

Puedes usar cualquier acción disponible en [GitHub Marketplace](#) que no entre en conflicto con estas [limitaciones](#).

¿Puedo usar otros proveedores de código fuente además de GitHub usar GitHub Actions en mi especificación de compilación?

Sí, pero GitHub sigue siendo necesario conectarse a para autenticarse y acceder a Actions. Para obtener más información, consulte [GitHub y un token de acceso a GitHub Enterprise Server](#).

¿Por qué tengo que conectarme a GitHub un proveedor de código fuente para poder usar GitHub Actions en mi especificación de compilación?

Para usar GitHub Actions en tu especificación de compilación, la fuente debe descargarse en un proceso de compilación. Las descargas anónimas tendrán una velocidad limitada, por lo que conectarte a ellas GitHub te ayudará a garantizar un acceso uniforme.

¿Cuánto cuesta usar GitHub Actions en mi especificación de compilación?

Se admite el uso de GitHub acciones en tu especificación de compilación sin coste adicional.

¿Qué regiones admiten el uso de GitHub Actions en mi especificación de compilación?

El uso de GitHub acciones en tus especificaciones de compilación está permitido en todas las regiones. CodeBuild Para obtener más información sobre Regiones de AWS dónde CodeBuild está disponible, consulta [AWS Servicios](#) por región.

Prácticas recomendadas para usar GitHub Actions en tus especificaciones de compilación

GitHub Las acciones son de código abierto, creadas y mantenidas por la comunidad. Seguimos el [modelo de responsabilidad compartida](#) y consideramos el código fuente de GitHub Actions como datos de clientes de los que tú eres responsable. GitHub A las acciones se les puede conceder acceso a secretos, fichas de repositorio, código fuente y enlaces de cuentas. Asegúrate de confiar en la fiabilidad y la seguridad de las GitHub acciones que planeas ejecutar.

Directrices más específicas y mejores prácticas de seguridad para Actions: GitHub

- [Refuerzo de la seguridad](#)
- [Prevención de solicitudes propias](#)
- [Entrada que no es de confianza](#)
- [Cómo confiar en los componentes básicos](#)

Limitaciones del uso de GitHub acciones en tus especificaciones de compilación en CodeBuild

- GitHub No se admiten las acciones de tu buildspec que dependan internamente del [githubcontexto](#) o que hagan referencia a recursos GitHub específicos, como las solicitudes de cambios y los problemas. CodeBuild Por ejemplo, las siguientes acciones no funcionarán en: CodeBuild
 - GitHub Acciones que intentan añadir, cambiar o actualizar GitHub recursos, como las acciones que actualizan las solicitudes de incorporación de cambios o que crean problemas en ellos GitHub.

Note

La mayoría de GitHub las acciones oficiales que aparecen en <https://github.com/actions> se basan en `github` el contexto. En su lugar, usa las acciones disponibles en [GitHub Marketplace](#).

- GitHub Las acciones de tu especificación de compilación que sean [acciones de contenedor de Docker](#) funcionarán, pero tu proyecto de compilación debe tener habilitado el [modo privilegiado](#) y ser ejecutado por el usuario predeterminado de Docker (root).
- Las acciones se deben ejecutar como usuario raíz. [Para obtener más información, consulta el tema USUARIO en la sección sobre el soporte de Dockerfile para Actions. GitHub](#)
- GitHub Las acciones de tu especificación de compilación no se admiten en CodeBuild proyectos configurados para ejecutarse en Windows.
- GitHub No se admiten los trabajos de acción (grupos de pasos) ni las propiedades de los trabajos de GitHub acción en tu especificación de compilación.
- GitHub Las acciones de tu buildspec no se admiten en CodeBuild proyectos configurados para ser activados por un webhook para un repositorio Git público. Para obtener más información, consulta [git-credential-helper](#)
- Las compilaciones de VPC sin acceso público a Internet no pueden ejecutar GitHub acciones en tu especificación de compilación.
- Cada fase de compilación admite una lista de comandos o una lista de pasos, pero no es posible utilizar ambas cosas en la misma fase. Por ejemplo, en el siguiente ejemplo, los pasos se utilizan en la fase previa a la compilación para enumerar GitHub las acciones, mientras que los comandos se utilizan en la fase de creación para enumerar los comandos. CodeBuild

```
version: 0.2
phases:
  pre-build:
    steps:
      - name: Lint Code Base
        uses: github/super-linter@v4
        env:
          VALIDATE_ALL_CODEBASE: 'true'
          DEFAULT_BRANCH: main
  build:
    commands:
      - echo "Building..."
      - npm run build
```

GitHub Referencia de especificaciones de compilación de Action Runner

Este tema contiene la referencia de especificaciones de compilación para las propiedades de Action Runner. GitHub

pasos

Secuencia opcional. Los pasos se utilizan para ejecutar comandos y acciones en CodeBuild. Para obtener más información, consulte [Usa la sintaxis GitHub Actions en una especificación de compilación en AWS CodeBuild](#).

Note

Cada fase de compilación admite una lista de `commands` o una lista de `steps`, pero no es posible utilizar ambas cosas en la misma fase.

Cada paso de compilación contiene las propiedades siguientes.

`id`

Opcional. Identificador del paso que se puede usar para hacer referencia al paso en otros [contextos](#).

`if`

Opcional. Declaración condicional que se puede usar para impedir la ejecución de un paso a menos que se cumpla una condición. Esta declaración puede usar cualquier [contexto](#) compatible, como hacer referencia a variables de CodeBuild entorno o [expresiones](#).

`name`

Opcional. Nombre del paso. Si no se especifica el nombre, el nombre predeterminado es el texto especificado en el comando `run`.

`uses`

Acción que se ejecuta en el paso. Para algunas acciones es necesario configurar entradas utilizando `with`. Consulte el README de la acción para determinar qué entradas son necesarias. Para obtener más información, consulte [¿Qué GitHub acciones puedo usar en mi especificación de compilación?](#).

Si `uses` está especificado en la fase de compilación, no se puede usar con `run`.

Note

Se recomienda incluir la versión de la acción que se va a utilizar. Esto se puede hacer especificando una etiqueta Git ref, SHA o Docker. Para obtener más información, consulte la [sintaxis de steps.uses](#).

run

Comando que ejecuta programas de línea de comandos. Pueden ser comandos de una sola línea o de varias líneas. De forma predeterminada, estos comandos se ejecutan mediante intérpretes de comandos sin inicio de sesión. Para elegir un intérprete de comandos distinto, utilice `shell`.

Si `run` está especificado en la fase de compilación, no se puede usar con `uses`.

shell

Opcional. Intérprete de comandos especificado para esta secuencia. Para ver los parámetros de intérprete de comandos compatibles, consulte [steps.shell](#). Si no se especifica, el intérprete de comandos utilizado es `bash`. Si `bash` no está disponible, se usa `sh`.

with

Opcional. Mapa de los parámetros de entrada definidos por la acción. Cada parámetro es un par clave-valor.

with.args

Opcional. Cadena que define las entradas de un contenedor de Docker.

with.entrypoint

Opcional. Punto de entrada de Docker especificado para el Dockerfile.

env

Opcional. Variables especificadas para los pasos que se van a utilizar en el entorno.

continue-on-error

Opcional. Valor booleano que indica si se puede ignorar un error en la secuencia de pasos.

false

Valor predeterminado. Si se produce un error en esta secuencia de pasos, se considerará la compilación como fallida.

`true`

Si se produce un error en esta secuencia de pasos, todavía es posible finalizar la compilación satisfactoriamente.

`timeout-minutes`

Opcional. Número máximo de minutos que puede durar el paso antes de finalizar. De forma predeterminada, no hay tiempo de espera. Si el tiempo de espera del paso supera el tiempo de espera de compilación, el paso se detendrá cuando se alcance el tiempo de espera de compilación.

El siguiente es un ejemplo del uso de la acción [Super-Linter](#) GitHub :

```
version: 0.2
phases:
  build:
    steps:
      - name: Lint Code Base
        uses: github/super-linter@v5
        env:
          VALIDATE_ALL_CODEBASE: true
          USE_FIND_ALGORITHM: true
          FILTER_REGEX_INCLUDE: '/github/workspace/buildspec.yml'
```

GitHub Ejemplos de sintaxis de acciones con AWS CodeBuild

Puedes usar estos grupos de ejemplos para experimentar con GitHub las acciones incluidas en tus especificaciones de construcción. CodeBuild

Temas

- [Ejemplo de Super-Linter Action GitHub](#)
- [Ejemplo de grafo de compilación por lotes](#)
- [Ejemplo de Amazon CodeGuru Reviewer](#)
- [AWS Secrets Manager muestra](#)
- [Archivo de variables de entorno](#)
- [Ejemplo de variable de entorno exportada](#)

Ejemplo de Super-Linter Action GitHub

En este ejemplo se muestra cómo añadir la GitHub acción [Super-Linter](#) a un proyecto. CodeBuild La acción Super-Linter inspecciona el código, encuentra áreas en las que el código contiene errores, problemas de formato y construcciones sospechosas y, a continuación, envía los resultados a la consola. CodeBuild

Puedes añadir la GitHub acción Super-Linter a tu CodeBuild proyecto actualizando la sección de fases del archivo buildspec.

```
version: 0.2
phases:
  build:
    steps:
      - name: Lint Code Base
        uses: github/super-linter@v5
        env:
          VALIDATE_ALL_CODEBASE: true
```

Los registros de Super-Linter tendrán un aspecto similar al siguiente:

```
/github/workspace/hello-world/app.js:3:13: Extra semicolon.
/github/workspace/hello-world/app.js:9:92: Trailing spaces not allowed.
/github/workspace/hello-world/app.js:21:7: Unnecessarily quoted property 'body' found.
/github/workspace/hello-world/app.js:31:1: Expected indentation of 2 spaces but found 4.
/github/workspace/hello-world/app.js:32:2: Newline required at end of file but not found.
```

Ejemplo de grafo de compilación por lotes

En el siguiente ejemplo, se define un grafo de compilación que crea una cadena de dependencias y ejecuta comandos utilizando steps. En este ejemplo, build1 se ejecuta primero porque no tiene dependencias. Puesto que build2 depende de build1, build2 se ejecuta después de completar build1. Para obtener más información, consulte [Grafo de compilación](#).

```
version: 0.2
batch:
  fast-fail: false
  build-graph:
    - identifier: build1
```

```
env:
  variables:
    BUILD_ID: build1
ignore-failure: false
- identifier: build2
env:
  variables:
    BUILD_ID: build2
depend-on:
  - build1

phases:
  build:
    steps:
      - run: echo $BUILD_ID
```

Ejemplo de Amazon CodeGuru Reviewer

Amazon CodeGuru Reviewer encuentra problemas en el código de Java y Python y recomienda cómo solucionarlos. En el siguiente ejemplo, se utiliza CodeGuru Reviewer para proporcionar revisiones completas del código de análisis del repositorio. Estas revisiones de código analizan todo el código de una rama específica. Para obtener más información, consulte [Crear revisiones de código con GitHub acciones](#) en la Guía del usuario de Amazon CodeGuru Reviewer.

```
version: 0.2
phases:
  build:
    steps:
      - name: Amazon CodeGuru Reviewer Scanner
        if: ${{ always() }}
        uses: aws-actions/codeguru-reviewer@v1.1
        with:
          s3_bucket: codeguru-reviewer-user

artifacts:
  files:
    - codeguru-results.sarif.json
```

Note

El bucket de Amazon S3 debe comenzar por el prefijo `codeguru-reviewer-`.

Los registros tendrán un aspecto similar al siguiente:

```
INFO CodeReview created with arn=arn:aws:codeguru-reviewer:region:account-  
id:association:id:code-review:RepositoryAnalysis-job for job=job  
INFO SARIF persisted to /github/workspace/codeguru-results.sarif.json  
INFO Amazon CodeGuru Reviewer job execution completed
```

Una vez finalizada la tarea de Amazon CodeGuru Reviewer, se genera un informe sarif como CodeBuild artefacto. Para obtener más información, consulte [Análisis completo del repositorio](#) en la Guía del usuario de Amazon CodeGuru Reviewer.

AWS Secrets Manager muestra

AWS Secrets Manager le ayuda a gestionar, recuperar y rotar las credenciales de las bases de datos, las credenciales de las aplicaciones, los tokens de OAuth, las claves de API y otros secretos a lo largo de sus ciclos de vida. El ejemplo siguiente define un secreto mediante Secrets Manager y ejecuta comandos mediante steps. [Para obtener más información, consulta ¿Qué es? AWS Secrets Manager](#) en la Guía AWS Secrets Manager del usuario.

```
version: 0.2  
env:  
  secrets-manager:  
    SECRET_VALUE: "arn:aws:secretsmanager:us-east-1:xxxx:secret:/secret-  
13IJg9:my_super_secret_key"  
phases:  
  build:  
    steps:  
      - run: echo $SECRET_VALUE
```

Los registros tendrán un aspecto similar al siguiente:

```
echo $SECRET_VALUE  
env:  
  SECRET_VALUE: ***  
***
```

Archivo de variables de entorno

El siguiente ejemplo define las variables de entorno en la secuencia env. La variable `S3_BUCKET` se define en la especificación de compilación y se le asigna `<bucket-name>` como valor. Se hace

referencia a esta variable en el condicional if como a una variable de entorno normal mediante el signo de dólar (\$) para acceder al contexto ambiental de GitHub Action. Para obtener más información consulte [Secuencia de env](#).

```
version: 0.2
env:
  variables:
    S3_BUCKET: "<bucket-name>"
phases:
  build:
    steps:
      - if: ${{ env.S3_BUCKET == '<bucket-name>' }}
        run: echo "S3 bucket is $S3_BUCKET"
```

Los registros tendrán un aspecto similar al siguiente:

```
echo "S3 bucket is $S3_BUCKET"
env:
  S3_BUCKET: my-s3-bucket
S3 bucket is my-s3-bucket
```

Ejemplo de variable de entorno exportada

Las variables de entorno exportadas se utilizan junto con ellas CodePipeline para exportar variables de entorno desde la fase de creación actual a las etapas siguientes del proceso de creación. En el ejemplo siguiente, se define una variable de entorno exportada en la secuencia env denominada *MY_VARIABLE* y se escribe en el archivo de entorno *GITHUB_ENV*.

```
version: 0.2
env:
  exported-variables:
    - MY_VARIABLE
phases:
  build:
    steps:
      - run: echo "MY_VARIABLE=my-value" >> $GITHUB_ENV
```

Para obtener más información, consulta [ExportedEnvironmentVariable](#) la referencia de la AWS CodeBuild API.

Proyectos de compilación pública en AWS CodeBuild

AWS CodeBuild permite poner a disposición del público en general los resultados de la compilación, los registros y los artefactos de los proyectos de compilación. Esto permite a los colaboradores de los repositorios de código fuente ver los resultados y descargar los artefactos de una compilación, sin necesidad de que accedan a una cuenta de AWS.

Al poner las compilaciones de un proyecto a disposición del público, todos los resultados, registros y artefactos de la compilación del proyecto, incluidas las compilaciones que se hayan ejecutado cuando el proyecto era privado, se ponen a disposición del público. Del mismo modo, cuando se convierte un proyecto de compilación pública en privado, los resultados de la compilación del proyecto dejan de estar disponibles para el público.

Para obtener información sobre cómo cambiar la visibilidad pública de los resultados de compilación de un proyecto, consulte [Habilitar el acceso a compilaciones públicas](#).

CodeBuild proporciona una URL para las compilaciones públicas del proyecto que es exclusiva del mismo. Para obtener la URL pública de un proyecto de compilación, realice el siguiente procedimiento:

1. Abra la consola de AWS CodeBuild en <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. En el panel de navegación, elija Proyectos de compilación.
3. Elija el enlace del proyecto de compilación del que desea obtener la URL pública.
4. La URL pública se muestra en el campo URL pública del proyecto en la sección Configuración. Puede elegir el enlace para abrir la URL o copiar la URL con el botón de copiar.

Warning

A la hora de hacer públicos los resultados de compilación de un proyecto, es necesario tener en cuenta lo siguiente:

- Todos los resultados, registros y artefactos de la compilación del proyecto, incluidas las compilaciones que se hayan ejecutado cuando el proyecto era privado, están disponibles para el público.
- Todos los registros y artefactos de compilación están disponibles para el público. Es posible que las variables de entorno, el código fuente y otra información confidencial estén

presentes en los registros y artefactos de compilación. Es necesario tener cuidado con la información que se envía a los registros de compilación. Algunas de las mejores prácticas son:

- No almacene valores confidenciales, especialmente la clave de acceso de y las claves de acceso secretas de AWS en variables de entorno. Le recomendamos utilizar un almacén de parámetros de Amazon EC2 Systems Manager o AWS Secrets Manager para almacenar los valores confidenciales.
- Siga [Prácticas recomendadas para utilizar webhooks](#) para limitar qué entidades pueden activar una compilación y no almacene las especificaciones de compilación en el propio proyecto para garantizar que los webhooks sean lo más seguros posible.
- Un usuario malintencionado puede utilizar las compilaciones públicas para distribuir artefactos maliciosos. Recomendamos a los administradores del proyecto que revisen todas las solicitudes de extracción para comprobar que se trata de cambios legítimos. También recomendamos validar todos los artefactos con sus sumas de verificación para asegurarse de que se están descargando los artefactos correctos.

Trabajar con compilaciones en AWS CodeBuild

Una compilación representa un conjunto de acciones realizadas por AWS CodeBuild para crear artefactos de salida (por ejemplo, un archivo JAR) en función de una serie de artefactos de entrada (por ejemplo, una colección de archivos de clases de Java).

Las siguientes reglas se aplican cuando ejecuta varias compilaciones:

- Cuando es posible, las compilaciones se ejecutan simultáneamente. El número máximo de compilaciones en ejecución simultánea puede variar. Para obtener más información, consulte [Cuotas para AWS CodeBuild](#).
- Si el proyecto de compilación tiene establecido un límite de compilaciones simultáneas y el número de compilaciones en ejecución alcanza este límite, las compilaciones devuelven un error. Para obtener más información, consulte [Activar el límite de compilaciones simultáneas](#).
- Si el proyecto de compilación no tiene establecido un límite de compilaciones simultáneas y el número de compilaciones en ejecución alcanza el límite de compilaciones simultáneas de la plataforma y el tipo de computación, las compilaciones quedan a la espera en una cola. El número máximo de compilaciones de una cola es de cinco veces el límite de compilaciones simultáneas. Para obtener más información, consulte [Cuotas para AWS CodeBuild](#).

Una compilación en una cola que no se inicia después del número de minutos especificado en su valor de tiempo de espera se elimina de la cola. El valor predeterminado del tiempo de espera es de ocho horas. Puede anular el tiempo de espera de la cola de compilación con un valor comprendido entre cinco minutos y ocho horas cuando ejecuta la compilación. Para obtener más información, consulte [Ejecutar una compilación en AWS CodeBuild](#).

No es posible predecir el orden en que se inician las compilaciones en cola.

Note

Puede acceder al historial de una compilación durante un año.

Puede realizar estas tareas cuando trabaje con compilaciones:

Temas

- [Ejecutar una compilación en AWS CodeBuild](#)
- [Ver detalles de las compilaciones de AWS CodeBuild](#)
- [Ver una lista de identificadores de compilación en AWS CodeBuild](#)
- [Ver una lista de identificadores de compilación de un proyecto de compilación en AWS CodeBuild](#)
- [Detener una compilación en AWS CodeBuild](#)
- [Detener una compilación por lotes en AWS CodeBuild](#)
- [Volver a intentar una compilación AWS CodeBuild](#)
- [Ver una compilación en ejecución en Session Manager](#)
- [Eliminación de compilaciones en AWS CodeBuild](#)

Ejecutar una compilación en AWS CodeBuild

Es posible usar la consola de AWS CodeBuild, la AWS CLI o los SDK de AWS para ejecutar una compilación en CodeBuild.

Temas

- [Ejecutar una compilación \(consola\)](#)
- [Ejecutar una compilación \(AWS CLI\)](#)

- [Ejecutar una compilación por lotes \(AWS CLI\)](#)
- [Iniciar la ejecución de compilaciones automáticamente \(AWS CLI\)](#)
- [Detener la ejecución de compilaciones automáticamente \(AWS CLI\)](#)
- [Ejecutar una compilación \(SDK de AWS\)](#)

Ejecutar una compilación (consola)

Si desea usar AWS CodePipeline para ejecutar una compilación con CodeBuild, omita estos pasos y siga las instrucciones de [Úselo CodePipeline con CodeBuild](#).

1. Abra la consola de AWS CodeBuild en <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. En el panel de navegación, elija Proyectos de compilación.
3. En la lista de proyectos de compilación, seleccione el proyecto de compilación.
4. Es posible ejecutar la compilación con la configuración predeterminada del proyecto de compilación o anular la configuración de compilación solo para esta compilación.
 - a. Para ejecutar la compilación con la configuración predeterminada del proyecto de compilación, seleccione Iniciar compilación. La compilación comienza inmediatamente.
 - b. Para anular la configuración predeterminada de compilación, seleccione Iniciar compilación con anulaciones. En la página Iniciar compilación, es posible anular lo siguiente:
 - Configuración de compilación
 - Fuente
 - Anulaciones de variables de entorno

Si necesita seleccionar anulaciones más avanzadas, seleccione Anulaciones de compilación avanzada. En esta página, es posible anular lo siguiente:

- Configuración de compilación
- Fuente
- Entorno
- Buildspec
- Artefactos
- Registros

Cuando haya hecho sus selecciones de anulación, seleccione Iniciar compilación.

Para obtener información detallada acerca de esta compilación, consulte [Ver detalles de las compilaciones \(consola\)](#).

Ejecutar una compilación (AWS CLI)

Note

Si desea usar CodePipeline para ejecutar una compilación con AWS CodeBuild, omita estos pasos y siga las instrucciones de [Crear una canalización que use CodeBuild \(AWS CLI\)](#).

Para obtener más información sobre cómo usar AWS CLI con CodeBuild, consulte [Referencia de la línea de comandos](#).

1. Ejecute el comando `start-build` de una de las siguientes formas:

```
aws codebuild start-build --project-name <project-name>
```

Utilice esta opción si desea ejecutar una compilación que utilice la última versión del artefacto de entrada de la compilación y la configuración existente del proyecto de compilación.

```
aws codebuild start-build --generate-cli-skeleton
```

Utilice esta opción si desea ejecutar una versión de compilación con una versión anterior del artefacto de entrada de compilación o si desea anular la configuración de los artefactos de salida de compilación, variables de entorno, especificación de compilación o período de tiempo de espera de compilación predeterminado.

2. Si ejecuta el comando `start-build` con la opción `--project-name`, sustituya `<project-name>` por el nombre del proyecto de compilación y vaya al paso 6 de este procedimiento. Para obtener una lista de proyectos de compilación, consulte [Ver una lista de nombres de proyectos de compilación](#).
3. Si ejecuta el comando `start-build` con la opción `--idempotency-token`, se incluye un identificador único en el que se distingue entre mayúsculas y minúsculas, o token, con la solicitud de `start-build`. El token es válido durante 5 minutos después de la solicitud. Si

repite la solicitud `start-build` con el mismo token, pero cambia un parámetro, CodeBuild devuelve un error de coincidencia de parámetros.

4. Si ejecuta el comando `start-build` con la opción `--generate-cli-skeleton`, se mostrarán en el resultado datos con formato JSON. Copie los datos en un archivo (por ejemplo, `start-build.json`) en la ubicación del equipo o instancia local en la que haya instalado la AWS CLI. Modifique los datos copiados para que coincidan con el siguiente formato y guarde los resultados:

```
{
  "projectName": "projectName",
  "sourceVersion": "sourceVersion",
  "artifactsOverride": {
    "type": "type",
    "location": "location",
    "path": "path",
    "namespaceType": "namespaceType",
    "name": "artifactsOverride-name",
    "packaging": "packaging"
  },
  "buildspecOverride": "buildspecOverride",
  "cacheOverride": {
    "location": "cacheOverride-location",
    "type": "cacheOverride-type"
  },
  "certificateOverride": "certificateOverride",
  "computeTypeOverride": "computeTypeOverride",
  "environmentTypeOverride": "environmentTypeOverride",
  "environmentVariablesOverride": {
    "name": "environmentVariablesOverride-name",
    "value": "environmentVariablesValue",
    "type": "environmentVariablesOverride-type"
  },
  "gitCloneDepthOverride": "gitCloneDepthOverride",
  "imageOverride": "imageOverride",
  "idempotencyToken": "idempotencyToken",
  "insecureSslOverride": "insecureSslOverride",
  "privilegedModeOverride": "privilegedModeOverride",
  "queuedTimeoutInMinutesOverride": "queuedTimeoutInMinutesOverride",
  "reportBuildStatusOverride": "reportBuildStatusOverride",
  "timeoutInMinutesOverride": "timeoutInMinutesOverride",
  "sourceAuthOverride": "sourceAuthOverride",
  "sourceLocationOverride": "sourceLocationOverride",
```

```
"serviceRoleOverride": "serviceRoleOverride",  
"sourceTypeOverride": "sourceTypeOverride"  
}
```

Reemplace los siguientes marcadores de posición:

- *projectName*: cadena necesaria. El nombre del proyecto de compilación que desea usar para esta compilación.
- *sourceVersion*: cadena opcional. Una versión del código fuente que se va a compilar, como la siguiente:
 - Para Amazon S3, el ID de versión correspondiente a la versión del archivo ZIP de entrada que desea compilar. Si no se especifica *sourceVersion*, se usa la versión más reciente.
 - Para CodeCommit, el ID de confirmación correspondiente a la versión del código fuente que desea compilar. Si no se especifica *sourceVersion*, se usa el ID de confirmación HEAD de la ramificación personalizada. (No puede especificar un nombre de etiqueta para *sourceVersion*, pero puede especificar el ID de confirmación de la etiqueta).
 - Para GitHub, el ID de confirmación, el ID de solicitud de inserción, el nombre de ramificación o el nombre de etiqueta correspondiente a la versión de código fuente que desea compilar. Si se especifica un ID de solicitud de inserción, este debe tener el formato *pr/pull-request-ID* (por ejemplo, *pr/25*). Si se especifica un nombre de ramificación, se usa el ID de confirmación HEAD de la ramificación. Si no se especifica *sourceVersion*, se usa el ID de confirmación HEAD de la ramificación personalizada.
 - Para Bitbucket, el ID de confirmación, el nombre de ramificación o el nombre de etiqueta correspondiente a la versión de código fuente que desea compilar. Si se especifica un nombre de ramificación, se usa el ID de confirmación HEAD de la ramificación. Si no se especifica *sourceVersion*, se usa el ID de confirmación HEAD de la ramificación personalizada.
- Los siguientes marcadores de posición son para *artifactsOverride*.
 - *type*: opcional. El tipo de artefacto de salida de la compilación que invalida el definido en el proyecto de compilación para esta compilación.
 - *location*: opcional. La ubicación del artefacto de salida de la compilación que invalida la definida en el proyecto de compilación para esta compilación.
 - *path*: opcional. La ruta del artefacto de salida de la compilación que invalida la definida en el proyecto de compilación para esta compilación.

- ***namespaceType***: opcional. El tipo de ruta del artefacto de salida de la compilación que invalida el definido en el proyecto de compilación para esta compilación.
- ***name***: opcional. El nombre del artefacto de salida de la compilación que invalida el definido en el proyecto de compilación para esta compilación.
- ***packaging***: opcional. El tipo de empaquetado del artefacto de salida de la compilación que invalida el definido en el proyecto de compilación para esta compilación.
- ***buildspecOverride***: opcional. Declaración de buildspec que invalida la declaración definida en el proyecto de esta compilación. Si se establece este valor, puede ser una definición de buildspec en línea, la ruta a un archivo de buildspec alternativo en relación con el valor de la variable de entorno CODEBUILD_SRC_DIR incorporada, o la ruta a un bucket de S3. El bucket de S3 debe estar en la misma región de AWS que el proyecto de compilación. Especifique el archivo buildspec utilizando su ARN (por ejemplo, `arn:aws:s3:::<my-codebuild-sample2>/buildspec.yml`). Si este valor no se proporciona o se establece en una cadena vacía, el código fuente debe contener un archivo `buildspec.yml` en su directorio raíz. Para obtener más información, consulte [Nombre de archivo y ubicación de almacenamiento de buildspec](#).
- Los siguientes marcadores de posición son para `cacheOverride`.
 - ***cacheOverride-location***: opcional. La ubicación de un objeto ProjectCache para esta compilación que anula el objeto ProjectCache especificado en el proyecto de compilación. `cacheOverride` es opcional y toma un objeto ProjectCache. `location` es obligatorio en un objeto ProjectCache.
 - ***cacheOverride-type***: opcional. El tipo de un objeto ProjectCache para esta compilación que anula el objeto ProjectCache especificado en el proyecto de compilación. `cacheOverride` es opcional y toma un objeto ProjectCache. `type` es obligatorio en un objeto ProjectCache.
- ***certificateOverride***: opcional. El nombre de un certificado para esta compilación que sustituye el especificado en el proyecto de compilación.
- ***environmentTypeOverride***: opcional. Un tipo de contenedor para esta compilación que sustituye el especificado en el proyecto de compilación. La cadena válida actual es `LINUX_CONTAINER`.
- Los siguientes marcadores de posición son para `environmentVariablesOverride`.
 - ***environmentVariablesOverride-name***: opcional. El nombre de una variable de entorno del proyecto de compilación cuyo valor desea invalidar para esta compilación.

- *environmentVariablesOverride-type*: opcional. El tipo de variable de entorno del proyecto de compilación cuyo valor desea invalidar para esta compilación.
- *environmentVariablesValue*: opcional. El valor de la variable de entorno definida en el proyecto de compilación que desea invalidar para esta compilación.
- *gitCloneDepthOverride*: opcional. El valor de Git clone depth en el proyecto de compilación cuyo valor desea invalidar para esta compilación. Si el tipo de origen es Amazon S3, este valor no se admite.
- *imageOverride*: opcional. El nombre de una imagen para esta compilación que sustituye el especificado en el proyecto de compilación.
- *idempotencyToken*: opcional. Una cadena que sirve de token para especificar que la solicitud de compilación es idempotente. Puede elegir cualquier cadena de 64 caracteres o menos. El token es válido durante 5 minutos después de la solicitud de inicio de la compilación. Si se repite la solicitud de inicio de compilación con el mismo token, pero se cambia un parámetro, CodeBuild devuelve un error de coincidencia de parámetros.
- *insecureSslOverride*: valor booleano opcional que especifica si la configuración TLS no segura especificada se va a invalidar en el proyecto de compilación. La configuración de TLS no segura determina si deben omitirse las advertencias de TLS al conectarse al código fuente del proyecto. Esta sustitución se aplica únicamente si el origen de la compilación es GitHub Enterprise Server.
- *privilegedModeOverride*: valor booleano opcional. Si se establece en true, la compilación sustituye el modo privilegiado en el proyecto de compilación.
- *queuedTimeoutInMinutesOverride*: Número entero opcional que especifica el número de minutos que se permite en cola una compilación antes de que transcurra el tiempo de espera. Su valor mínimo es de cinco minutos y su valor máximo es 480 minutos (ocho horas).
- *reportBuildStatusOverride*: valor booleano opcional que especifica si se debe enviar al proveedor de código fuente el estado de inicio y finalización de una compilación. Si define este valor con un proveedor de código fuente distinto de GitHub, GitHub Enterprise Server o Bitbucket, se genera una excepción `InvalidInputException`.
- *sourceAuthOverride*: cadena opcional. Un tipo de autorización para esta compilación que sustituye la definida en el proyecto de compilación. Esta sustitución se aplica únicamente si el origen del proyecto de compilación es Bitbucket o GitHub.
- *sourceLocationOverride*: cadena opcional. Una ubicación que sustituye la ubicación de código fuente de este proyecto por la definida en el proyecto de compilación.

- *serviceRoleOverride*: cadena opcional. El nombre de un rol de servicio para esta compilación que sustituye el especificado en el proyecto de compilación.
- *sourceTypeOverride*: cadena opcional. Un tipo de entrada de código fuente para esta compilación que sustituye la entrada de código fuente definida en el proyecto de compilación. Las cadenas válidas son NO_SOURCE, CODECOMMIT, CODEPIPELINE, GITHUB, S3, BITBUCKET y GITHUB_ENTERPRISE.
- *timeoutInMinutesOverride*: número opcional. El número de minutos de tiempo de espera de la compilación que invalida el definido en el proyecto de compilación para esta compilación.

Le recomendamos que almacene una variable de entorno con un valor confidencial, como un ID de clave de acceso de AWS, una clave de acceso secreta de AWS o una contraseña, como un parámetro en el almacén de parámetros de Amazon EC2 Systems Manager. CodeBuild puede utilizar un parámetro almacenado en el almacén de parámetros de Amazon EC2 Systems Manager solo si el nombre del parámetro comienza por `/CodeBuild/` (por ejemplo, `/CodeBuild/dockerLoginPassword`). Puede utilizar la consola de CodeBuild para crear un parámetro en Amazon EC2 Systems Manager. Elija Crear un parámetro y siga las instrucciones. (En ese cuadro de diálogo, en Clave de KMS, puede especificar si lo desea el ARN de una clave de AWS KMS de su cuenta. Amazon EC2 Systems Manager utiliza esta clave para cifrar el valor del parámetro durante el almacenamiento y descifrarlo durante la recuperación). Si usa la consola de CodeBuild para crear un parámetro, la consola asigna `/CodeBuild/` al principio del parámetro cuando lo almacena. Sin embargo, si utiliza la consola del almacén de parámetros de Amazon EC2 Systems Manager para crear un parámetro, el nombre del parámetro debe comenzar por `/CodeBuild/` y debe establecer Tipo en Cadena segura. Para obtener más información, consulte [Almacenamiento de parámetros de AWS Systems Manager](#) y [Tutorial : Crear y probar un parámetro de cadena de caracteres \(consola\)](#) en la Guía del usuario de Amazon EC2 Systems Manager.

Si el proyecto de compilación hace referencia a parámetros almacenados en el almacén de parámetros de Amazon EC2 Systems Manager, el rol de servicio del proyecto de compilación debe permitir la acción `ssm:GetParameters`. Si se ha seleccionado Crear un nuevo rol de servicio en su cuenta anteriormente, CodeBuild incluye esta acción en el rol de servicio predeterminado del proyecto de compilación automáticamente. Sin embargo, si se ha seleccionado Seleccionar un rol de servicio existente de su cuenta, debe incluir esta acción en su rol de servicio por separado.

Las variables de entorno que defina reemplazan las variables de entorno existentes. Por ejemplo, si la imagen de Docker ya contiene una variable de entorno denominada MY_VAR con un valor de my_value y establece una variable de entorno denominada MY_VAR con un valor de other_value, my_value se reemplaza por other_value. Asimismo, si la imagen de Docker ya contiene una variable de entorno denominada PATH con un valor de /usr/local/sbin:/usr/local/bin y establece una variable de entorno denominada PATH con un valor de \$PATH:/usr/share/ant/bin, /usr/local/sbin:/usr/local/bin se reemplaza por el valor literal \$PATH:/usr/share/ant/bin.

No establezca variables de entorno con un nombre que empiece por CODEBUILD_. Este prefijo se reserva para uso interno de .

Si se define una variable de entorno con el mismo nombre en varios lugares, el valor de la variable de entorno se determina de la siguiente manera:

- El valor de la llamada a la operación de inicio de la compilación tiene la máxima prioridad.
- El valor de la definición del proyecto de compilación es el siguiente en orden de prioridad.
- El valor en la declaración del archivo buildspec tiene la prioridad más baja.

Para obtener información sobre los valores válidos para estos marcadores de posición, consulte [Crear un proyecto de compilación \(AWS CLI\)](#). Para obtener una lista de las últimas opciones de configuración de un proyecto de compilación, consulte [Ver los detalles de un proyecto de compilación](#).

5. Cambie al directorio que contiene el archivo que acaba de guardar y ejecute de nuevo el comando start-build.

```
aws codebuild start-build --cli-input-json file://start-build.json
```

6. Si el comando se ejecuta correctamente, aparecerán datos similares a los que se describen en el procedimiento [Para ejecutar la compilación](#) en el resultado.

Para trabajar con información detallada acerca de esta compilación, anote el valor de id en la salida y después consulte [Ver detalles de las compilaciones \(AWS CLI\)](#).

Ejecutar una compilación por lotes (AWS CLI)

1. Ejecute el comando start-build-batch de una de las siguientes formas:

```
aws codebuild start-build-batch --project-name <project-name>
```

Utilice esta opción si desea ejecutar una compilación que utilice la última versión del artefacto de entrada de la compilación y la configuración existente del proyecto de compilación.

```
aws codebuild start-build-batch --generate-cli-skeleton > <json-file>
```

Utilice esta opción si desea ejecutar una versión de compilación con una versión anterior del artefacto de entrada de compilación o si desea anular la configuración de los artefactos de salida de compilación, variables de entorno, especificación de compilación o período de tiempo de espera de compilación predeterminado.

2. Si ejecuta el comando `start-build-batch` con la opción `--project-name`, sustituya *<project-name>* por el nombre del proyecto de compilación y vaya al paso 6 de este procedimiento. Para obtener una lista de proyectos de compilación, consulte [Ver una lista de nombres de proyectos de compilación](#).
3. Si ejecuta el comando `start-build-batch` con la opción `--idempotency-token`, se incluye un identificador único en el que se distingue entre mayúsculas y minúsculas, o token, con la solicitud de `start-build-batch`. El token es válido durante 5 minutos después de la solicitud. Si repite la solicitud `start-build-batch` con el mismo token, pero cambia un parámetro, CodeBuild devuelve un error de coincidencia de parámetros.
4. Si ejecuta el comando `start-build-batch` con la opción `--generate-cli-skeleton`, se envían datos en formato JSON al archivo *<json-file>*. Este archivo es similar al esqueleto producido por el comando `start-build`, con la incorporación del objeto siguiente. Para obtener más información acerca de los objetos comunes, consulte [Ejecutar una compilación \(AWS CLI\)](#).

Modifique este archivo para añadir cualquier modificación a la compilación y guarde los resultados.

```
"buildBatchConfigOverride": {  
  "combineArtifacts": combineArtifacts,  
  "restrictions": {  
    "computeTypesAllowed": [  
      allowedComputeTypes  
    ],  
    "maximumBuildsAllowed": maximumBuildsAllowed  
  },  
  "serviceRole": "batchServiceRole",
```

```
"timeoutInMins": batchTimeout
}
```

El objeto `buildBatchConfigOverride` es una estructura [ProjectBuildBatchConfig](#) que contiene las modificaciones a la configuración de compilación por lotes para esta compilación.

combineArtifacts

Especifica si los artefactos de compilación para la compilación por lotes se deben combinar en una única ubicación de artefactos.

allowedComputeTypes

Una matriz de cadenas que especifican los tipos de computación permitidos para la compilación por lotes. Consulte los valores en [Tipos de computación del entorno de compilación](#).

maximumBuildsAllowed

Especifica el número máximo de compilaciones permitidas.

batchServiceRole

Especifica el ARN del rol de servicio para el proyecto de compilación por lotes.

batchTimeout

Especifica la cantidad máxima de tiempo, en minutos, en que se debe completar la compilación por lotes.

5. Cambie al directorio que contiene el archivo que acaba de guardar y ejecute de nuevo el comando `start-build-batch`.

```
aws codebuild start-build-batch --cli-input-json file://start-build.json
```

6. Si se ejecuta correctamente, la salida de la consola muestra la representación en JSON de un objeto [BuildBatch](#). Consulte [Sintaxis de respuesta de StartBuildBatch](#) para ver un ejemplo de estos datos.

Detener la ejecución de compilaciones automáticamente (AWS CLI)

Si el código fuente está almacenado en un repositorio de GitHub o GitHub Enterprise, puede configurar webhooks de GitHub para que AWS CodeBuild vuelva a compilar el código fuente siempre que se inserte un cambio de código en el repositorio. Para obtener más información, consulte [Iniciar la ejecución de compilaciones automáticamente \(AWS CLI\)](#).

Si ha activado este comportamiento, puede desactivarlo ejecutando los comandos de delete-webhook como se indica a continuación:

```
aws codebuild delete-webhook --project-name <project-name>
```

- Donde *<project-name>* es el nombre del proyecto de compilación que contiene el código fuente que se va a volver a compilar.

Si este comando se ejecuta correctamente, no aparecerá información ni errores en el resultado.

Note

Esto elimina el webhook únicamente del proyecto de CodeBuild. También debe eliminar el webhook del repositorio de GitHub o GitHub Enterprise Server.

Ejecutar una compilación (SDK de AWS)

Para usar CodePipeline para ejecutar una compilación con AWS CodeBuild, omita estos pasos y siga las instrucciones de [Usar AWS CodePipeline con AWS CodeBuild para probar el código y ejecutar compilaciones](#) en su lugar.

Para obtener información sobre cómo usar CodeBuild con los SDK de AWS, consulte [Referencia de los SDK y las herramientas de AWS](#).

Ver detalles de las compilaciones de AWS CodeBuild

Es posible usar la consola de AWS CodeBuild, la AWS CLI o los SDK de AWS para ver información detallada sobre las compilaciones administradas por CodeBuild.

Temas

- [Ver detalles de las compilaciones \(consola\)](#)

- [Ver detalles de las compilaciones \(AWS CLI\)](#)
- [Ver detalles de las compilaciones \(SDK de AWS\)](#)
- [Transiciones de fases de compilación](#)

Ver detalles de las compilaciones (consola)

1. Abra la consola de AWS CodeBuild en <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Haga una de las siguientes acciones:
 - En el panel de navegación, elija Build history. En la lista de compilaciones, en la columna Build run (Ejecución de compilación), elija el enlace de la compilación.
 - En el panel de navegación, elija Build projects. En la lista de proyectos de compilación, en la columna Name (Nombre), elija el enlace del nombre del proyecto de compilación. A continuación, en la lista de compilaciones, en la columna Build run (Ejecución de compilación), elija el enlace de la compilación.

Note

De forma predeterminada, solo se muestran las diez compilaciones o proyectos de compilación más recientes. Para ver más compilaciones o proyectos de compilación, elija el icono de rueda dentada y, a continuación, seleccione un valor diferente para Builds per page (Compilaciones por página) o Projects per page (Proyectos por página) o seleccione las flechas atrás y adelante.

Ver detalles de las compilaciones (AWS CLI)

Para obtener más información sobre cómo usar la AWS CLI con AWS CodeBuild, consulte [Referencia de la línea de comandos](#).

Ejecute el comando batch-get-builds:

```
aws codebuild batch-get-builds --ids ids
```

Reemplace el siguiente marcador de posición:

- **ids**: cadena obligatoria. Uno o varios ID de compilación para los que desea ver los detalles. Para especificar más de un ID de compilación, separe cada ID de compilación con un espacio. Puede especificar hasta 100 ID de compilación. Para obtener una lista de identificadores de compilación, consulte los siguientes temas:
 - [Ver una lista de identificadores de compilación \(AWS CLI\)](#)
 - [Ver una lista de identificadores de compilación de un proyecto de compilación \(AWS CLI\)](#)

Por ejemplo, si ejecuta este comando:

```
aws codebuild batch-get-builds --ids codebuild-demo-project:e9c4f4df-3f43-41d2-ab3a-60fe2EXAMPLE codebuild-demo-project:815e755f-bade-4a7e-80f0-efe51EXAMPLE my-other-project:813bb6c6-891b-426a-9dd7-6d8a3EXAMPLE
```

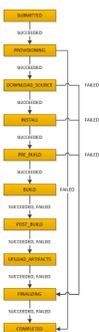
Si el comando tiene éxito, aparecen datos similares a los descritos en [Para ver la información resumida de la compilación](#) en la salida.

Ver detalles de las compilaciones (SDK de AWS)

Para obtener más información sobre cómo usar AWS CodeBuild con los SDK de AWS, consulte [Referencia de los SDK y las herramientas de AWS](#).

Transiciones de fases de compilación

Las compilaciones de AWS CodeBuild avanzan en fases:



⚠ Important

La fase `UPLOAD_ARTIFACTS` se intenta siempre, aunque la fase `BUILD` produzca un error.

Ver una lista de identificadores de compilación en AWS CodeBuild

Es posible usar la consola de AWS CodeBuild, la AWS CLI o los SDK de AWS para ver la lista de identificadores de las compilaciones administradas por CodeBuild.

Temas

- [Ver una lista de identificadores de compilación \(consola\)](#)
- [Ver una lista de identificadores de compilación \(AWS CLI\)](#)
- [Ver una lista de identificadores de compilación por lotes \(AWS CLI\)](#)
- [Ver una lista de identificadores de compilación \(SDK de AWS\)](#)

Ver una lista de identificadores de compilación (consola)

1. Abra la consola de AWS CodeBuild en <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. En el panel de navegación, elija Build history.

Note

De forma predeterminada, solo se muestran las diez compilaciones más recientes. Para ver más compilaciones, elija el icono de rueda dentada y, a continuación, seleccione un valor diferente para Builds per page (Compilaciones por página) o use las flechas atrás y adelante.

Ver una lista de identificadores de compilación (AWS CLI)

Para obtener más información sobre cómo usar AWS CLI con CodeBuild, consulte [Referencia de la línea de comandos](#).

- Ejecute el comando list-builds:

```
aws codebuild list-builds --sort-order sort-order --next-token next-token
```

En el comando anterior, sustituya los siguientes marcadores de posición:

- *sort-order*: cadena opcional que se utiliza para indicar cómo muestran los identificadores de compilación. Los valores válidos son ASCENDING y DESCENDING.
- *next-token*: cadena opcional. Durante una ejecución anterior, si hubiera más de 100 elementos en la lista, solo se devolverían los 100 primeros, junto con una única cadena denominada next token. Para obtener el siguiente lote de elementos de la lista, ejecute de nuevo este comando añadiendo el siguiente token a la llamada. Para obtener todos los elementos de la lista, siga ejecutando el comando con cada uno de los siguientes tokens hasta que no se devuelvan más tokens.

Por ejemplo, si ejecuta este comando:

```
aws codebuild list-builds --sort-order ASCENDING
```

Un resultado similar al siguiente podría aparecer en la salida:

```
{
  "nextToken": "4AEA6u7J...The full token has been omitted for brevity...MzY2OA==",
  "ids": [
    "codebuild-demo-project:815e755f-bade-4a7e-80f0-efe51EXAMPLE"
    "codebuild-demo-project:84a7f3d1-d40e-4956-b4cf-7a9d4EXAMPLE"
    ... The full list of build IDs has been omitted for brevity ...
    "codebuild-demo-project:931d0b72-bf6f-4040-a472-5c707EXAMPLE"
  ]
}
```

Si ejecuta este comando de nuevo:

```
aws codebuild list-builds --sort-order ASCENDING --next-token 4AEA6u7J...The full token has been omitted for brevity...MzY2OA==
```

Un resultado similar al siguiente podría aparecer en la salida:

```
{
  "ids": [
    "codebuild-demo-project:49015049-21cf-4b50-9708-df115EXAMPLE",
    "codebuild-demo-project:543e7206-68a3-46d6-a4da-759abEXAMPLE",
    ... The full list of build IDs has been omitted for brevity ...
    "codebuild-demo-project:c282f198-4582-4b38-bdc0-26f96EXAMPLE"
  ]
}
```

```
]
}
```

Ver una lista de identificadores de compilación por lotes (AWS CLI)

Para obtener más información sobre cómo usar AWS CLI con CodeBuild, consulte [Referencia de la línea de comandos](#).

- Ejecute el comando `list-build-batches`:

```
aws codebuild list-build-batches --sort-order sort-order --next-token next-token
```

En el comando anterior, sustituya los siguientes marcadores de posición:

- *sort-order*: cadena opcional que se utiliza para indicar cómo muestran los identificadores de compilación por lotes. Los valores válidos son ASCENDING y DESCENDING.
- *next-token*: cadena opcional. Durante una ejecución anterior, si hubiera más de 100 elementos en la lista, solo se devolverían los 100 primeros, junto con una única cadena denominada next token. Para obtener el siguiente lote de elementos de la lista, ejecute de nuevo este comando añadiendo el siguiente token a la llamada. Para obtener todos los elementos de la lista, siga ejecutando el comando con cada uno de los siguientes tokens hasta que no se devuelvan más tokens.

Por ejemplo, si ejecuta este comando:

```
aws codebuild list-build-batches --sort-order ASCENDING
```

Un resultado similar al siguiente podría aparecer en la salida:

```
{
  "nextToken": "4AEA6u7J...The full token has been omitted for brevity...MzY20A==",
  "ids": [
    "codebuild-demo-project:815e755f-bade-4a7e-80f0-efe51EXAMPLE"
    "codebuild-demo-project:84a7f3d1-d40e-4956-b4cf-7a9d4EXAMPLE"
    ... The full list of build IDs has been omitted for brevity ...
    "codebuild-demo-project:931d0b72-bf6f-4040-a472-5c707EXAMPLE"
  ]
}
```

Si ejecuta este comando de nuevo:

```
aws codebuild list-build-batches --sort-order ASCENDING --next-token 4AEA6u7J...The full token has been omitted for brevity...MzY2OA==
```

Un resultado similar al siguiente podría aparecer en la salida:

```
{
  "ids": [
    "codebuild-demo-project:49015049-21cf-4b50-9708-df115EXAMPLE",
    "codebuild-demo-project:543e7206-68a3-46d6-a4da-759abEXAMPLE",
    ... The full list of build IDs has been omitted for brevity ...
    "codebuild-demo-project:c282f198-4582-4b38-bdc0-26f96EXAMPLE"
  ]
}
```

Ver una lista de identificadores de compilación (SDK de AWS)

Para obtener más información sobre cómo usar CodeBuild con los SDK de AWS, consulte [Referencia de los SDK y las herramientas de AWS](#).

Ver una lista de identificadores de compilación de un proyecto de compilación en AWS CodeBuild

Puede utilizar la consola de AWS CodeBuild, la AWS CLI o los SDK de AWS para ver la lista de identificadores de compilación de un proyecto de compilación en CodeBuild.

Temas

- [Ver una lista de identificadores de compilación de un proyecto de compilación \(consola\)](#)
- [Ver una lista de identificadores de compilación de un proyecto de compilación \(AWS CLI\)](#)
- [Ver una lista de identificadores de compilación por lotes de un proyecto de compilación \(AWS CLI\)](#)
- [Ver una lista de identificadores de compilación de un proyecto de compilación \(SDK de AWS\)](#)

Ver una lista de identificadores de compilación de un proyecto de compilación (consola)

1. Abra la consola de CodeBuild en <https://console.aws.amazon.com/codebuild/>.
2. En el panel de navegación, elija Build projects. En la lista de proyectos de compilación, en la columna Name (Nombre), elija el proyecto de compilación.

Note

De forma predeterminada, solo se muestran las 100 compilaciones o proyectos de compilación más recientes. Para ver más compilaciones o proyectos de compilación, elija el icono de rueda dentada y, a continuación, seleccione un valor diferente para Builds per page (Compilaciones por página) o Projects per page (Proyectos por página) o seleccione las flechas atrás y adelante.

Ver una lista de identificadores de compilación de un proyecto de compilación (AWS CLI)

Para obtener más información sobre cómo usar la AWS CLI con AWS CodeBuild, consulte [Referencia de la línea de comandos](#).

Ejecute el comando list-builds-for-project de la siguiente manera:

```
aws codebuild list-builds-for-project --project-name project-name --sort-order sort-order --next-token next-token
```

En el comando anterior, sustituya los siguientes marcadores de posición:

- ***project-name***: cadena obligatoria utilizada para indicar el nombre del proyecto de compilación para el que desea mostrar los identificadores de compilación. Para obtener una lista de proyectos de compilación, consulte [Ver una lista de nombres de proyectos de compilación \(AWS CLI\)](#).
- ***sort-order***: cadena opcional que se utiliza para indicar cómo muestran los identificadores de compilación. Los valores válidos son ASCENDING y DESCENDING.
- ***next-token***: cadena opcional. Durante una ejecución anterior, si hubiera más de 100 elementos en la lista, solo se devolverían los 100 primeros, junto con una única cadena denominada next

token. Para obtener el siguiente lote de elementos de la lista, ejecute de nuevo este comando añadiendo el siguiente token a la llamada. Para obtener todos los elementos de la lista, siga ejecutando el comando con cada uno de los siguientes tokens devueltos hasta que no se devuelvan más tokens.

Por ejemplo, si ejecuta un comando similar a este:

```
aws codebuild list-builds-for-project --project-name codebuild-demo-project --sort-order ASCENDING
```

Aparecerá un resultado como el siguiente en la salida:

```
{
  "nextToken": "4AEA6u7J...The full token has been omitted for brevity...MzY2OA==",
  "ids": [
    "codebuild-demo-project:9b175d16-66fd-4e71-93a0-50a08EXAMPLE"
    "codebuild-demo-project:a9d1bd09-18a2-456b-8a36-7d65aEXAMPLE"
    ... The full list of build IDs has been omitted for brevity ...
    "codebuild-demo-project:fe70d102-c04f-421a-9cfa-2dc15EXAMPLE"
  ]
}
```

Si ejecuta este comando de nuevo:

```
aws codebuild list-builds-for-project --project-name codebuild-demo-project --sort-order ASCENDING --next-token 4AEA6u7J...The full token has been omitted for brevity...MzY2OA==
```

Podría aparecer un resultado como el siguiente en la salida:

```
{
  "ids": [
    "codebuild-demo-project:98253670-7a8a-4546-b908-dc890EXAMPLE"
    "codebuild-demo-project:ad5405b2-1ab3-44df-ae2d-fba84EXAMPLE"
    ... The full list of build IDs has been omitted for brevity ...
    "codebuild-demo-project:f721a282-380f-4b08-850a-e0ac1EXAMPLE"
  ]
}
```

Ver una lista de identificadores de compilación por lotes de un proyecto de compilación (AWS CLI)

Para obtener más información sobre cómo usar la AWS CLI con AWS CodeBuild, consulte [Referencia de la línea de comandos](#).

Ejecute el comando `list-build-batches-for-project` de la siguiente manera:

```
aws codebuild list-build-batches-for-project --project-name project-name --sort-order sort-order --next-token next-token
```

En el comando anterior, sustituya los siguientes marcadores de posición:

- ***project-name***: cadena obligatoria utilizada para indicar el nombre del proyecto de compilación para el que desea mostrar los identificadores de compilación. Para obtener una lista de proyectos de compilación, consulte [Ver una lista de nombres de proyectos de compilación \(AWS CLI\)](#).
- ***sort-order***: cadena opcional que se utiliza para indicar cómo muestran los identificadores de compilación. Los valores válidos son ASCENDING y DESCENDING.
- ***next-token***: cadena opcional. Durante una ejecución anterior, si hubiera más de 100 elementos en la lista, solo se devolverían los 100 primeros, junto con una única cadena denominada next token. Para obtener el siguiente lote de elementos de la lista, ejecute de nuevo este comando añadiendo el siguiente token a la llamada. Para obtener todos los elementos de la lista, siga ejecutando el comando con cada uno de los siguientes tokens devueltos hasta que no se devuelvan más tokens.

Por ejemplo, si ejecuta un comando similar a este:

```
aws codebuild list-build-batches-for-project --project-name codebuild-demo-project --sort-order ASCENDING
```

Aparecerá un resultado como el siguiente en la salida:

```
{
  "nextToken": "4AEA6u7J...The full token has been omitted for brevity...MzY20A==",
  "ids": [
    "codebuild-demo-project:9b175d16-66fd-4e71-93a0-50a08EXAMPLE"
    "codebuild-demo-project:a9d1bd09-18a2-456b-8a36-7d65aEXAMPLE"
  ]
}
```

```
... The full list of build IDs has been omitted for brevity ...  
"codebuild-demo-project:fe70d102-c04f-421a-9cfa-2dc15EXAMPLE"  
]  
}
```

Si ejecuta este comando de nuevo:

```
aws codebuild list-build-batches-for-project --project-name codebuild-demo-project  
--sort-order ASCENDING --next-token 4AEA6u7J...The full token has been omitted for  
brevity...MzY2OA==
```

Podría aparecer un resultado como el siguiente en la salida:

```
{  
  "ids": [  
    "codebuild-demo-project:98253670-7a8a-4546-b908-dc890EXAMPLE"  
    "codebuild-demo-project:ad5405b2-1ab3-44df-ae2d-fba84EXAMPLE"  
    ... The full list of build IDs has been omitted for brevity ...  
    "codebuild-demo-project:f721a282-380f-4b08-850a-e0ac1EXAMPLE"  
  ]  
}
```

Ver una lista de identificadores de compilación de un proyecto de compilación (SDK de AWS)

Para obtener más información sobre cómo usar AWS CodeBuild con los SDK de AWS, consulte [Referencia de los SDK y las herramientas de AWS](#).

Detener una compilación en AWS CodeBuild

Puede usar la consola de AWS CodeBuild, la AWS CLI o los SDK de AWS para detener una compilación de AWS CodeBuild.

Temas

- [Detener una compilación \(consola\)](#)
- [Detener una compilación \(AWS CLI\)](#)
- [Detener una compilación \(SDK de AWS\)](#)

Detener una compilación (consola)

1. Abra la consola de AWS CodeBuild en <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Haga una de las siguientes acciones:
 - Si aparece la página **nombre-proyecto-compilación:ID-compilación**, elija Stop build (Detener compilación).
 - En el panel de navegación, elija Build history. En la lista de compilaciones, seleccione la casilla correspondiente a la compilación y después elija Stop build (Detener compilación).
 - En el panel de navegación, elija Build projects. En la lista de proyectos de compilación, en la columna Name (Nombre), elija el enlace correspondiente al nombre del proyecto de compilación. En la lista de compilaciones, seleccione la casilla correspondiente a la compilación y después elija Stop build (Detener compilación).

Note

De forma predeterminada, solo se muestran las 100 compilaciones o proyectos de compilación más recientes. Para ver más compilaciones o proyectos de compilación, elija el icono de rueda dentada y, a continuación, seleccione un valor diferente para Builds per page (Compilaciones por página) o Projects per page (Proyectos por página) o seleccione las flechas atrás y adelante.

Si AWS CodeBuild no puede detener correctamente una compilación (por ejemplo, porque el proceso ya se ha completado), el botón Stop (Detener) estará desactivado o podría no aparecer.

Detener una compilación (AWS CLI)

- Ejecute el comando stop-build:

```
aws codebuild stop-build --id id
```

En el comando anterior, sustituya el siguiente marcador de posición:

- ***id***: cadena obligatoria. El ID de la compilación que desea detener. Para obtener una lista de identificadores de compilación, consulte los siguientes temas:

- [Ver una lista de identificadores de compilación \(AWS CLI\)](#)
- [Ver una lista de identificadores de compilación de un proyecto de compilación \(AWS CLI\)](#)

Si AWS CodeBuild detiene correctamente la compilación, el valor de `buildStatus` del objeto `build` de la salida será `STOPPED`.

Si CodeBuild no puede detener correctamente la compilación (por ejemplo, porque la compilación ya se ha completado), el valor de `buildStatus` del objeto `build` de la salida es el estado final de la compilación (por ejemplo, `SUCCEEDED`).

Detener una compilación (SDK de AWS)

Para obtener más información sobre cómo usar AWS CodeBuild con los SDK de AWS, consulte [Referencia de los SDK y las herramientas de AWS](#).

Detener una compilación por lotes en AWS CodeBuild

Puede usar la consola de AWS CodeBuild, la AWS CLI o los SDK de AWS para detener una compilación por lotes de AWS CodeBuild.

Temas

- [Detener una compilación por lotes \(consola\)](#)
- [Detener una compilación por lotes \(AWS CLI\)](#)
- [Detener una compilación por lotes \(SDK de AWS\)](#)

Detener una compilación por lotes (consola)

1. Abra la consola de AWS CodeBuild en <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Haga una de las siguientes acciones:
 - Si aparece la página ***nombre-proyecto-compilación:ID-compilación***, elija Stop build (Detener compilación).
 - En el panel de navegación, elija Build history. En la lista de compilaciones, seleccione la casilla correspondiente a la compilación y después elija Stop build (Detener compilación).

- En el panel de navegación, elija Build projects. En la lista de proyectos de compilación, en la columna Name (Nombre), elija el enlace correspondiente al nombre del proyecto de compilación. En la lista de compilaciones, seleccione la casilla correspondiente a la compilación y después elija Stop build (Detener compilación).

Note

De forma predeterminada, solo se muestran las 100 compilaciones o proyectos de compilación más recientes. Para ver más compilaciones o proyectos de compilación, elija el icono de rueda dentada y, a continuación, seleccione un valor diferente para Builds per page (Compilaciones por página) o Projects per page (Proyectos por página) o seleccione las flechas atrás y adelante.

Si AWS CodeBuild no puede detener correctamente una compilación por lotes (por ejemplo, porque el proceso ya se ha completado), el botón Detener está desactivado.

Detener una compilación por lotes (AWS CLI)

- Ejecute el comando [stop-build-batch](#):

```
aws codebuild stop-build-batch --id <batch-build-id>
```

En el comando anterior, sustituya el siguiente marcador de posición:

- *<batch-build-id>*: cadena obligatoria. Identificador del lote que se desea detener. Para obtener una lista de identificadores de compilación por lotes, consulte los temas siguientes:
 - [Ver una lista de identificadores de compilación por lotes \(AWS CLI\)](#)
 - [Ver una lista de identificadores de compilación por lotes de un proyecto de compilación \(AWS CLI\)](#)

Si AWS CodeBuild detiene correctamente la compilación por lotes, el valor de `buildBatchStatus` del objeto `buildBatch` de la salida será STOPPED.

Si CodeBuild no puede detener correctamente la compilación por lotes (por ejemplo, porque la compilación ya se ha completado), el valor de `buildBatchStatus` del objeto `buildBatch` de la salida es el estado final de la compilación (por ejemplo, SUCCEEDED).

Detener una compilación por lotes (SDK de AWS)

Para obtener más información sobre cómo usar AWS CodeBuild con los SDK de AWS, consulte [Referencia de los SDK y las herramientas de AWS](#).

Volver a intentar una compilación AWS CodeBuild

Es posible utilizar la consola de AWS CodeBuild, AWS CLI o los SDK de AWS para volver a intentar una compilación o una compilación por lotes en AWS CodeBuild.

Temas

- [Volver a intentar una compilación \(consola\)](#)
- [Reintentar una compilación \(AWS CLI\)](#)
- [Volver a intentar una compilación \(SDK de AWS\)](#)

Volver a intentar una compilación (consola)

1. Abra la consola de AWS CodeBuild en <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Haga una de las acciones siguientes:
 - Si aparece la página **build-project-name:build-ID**, seleccione Reintentar compilación.
 - En el panel de navegación, elija Build history. En la lista de compilaciones, seleccione la casilla correspondiente a la compilación y después elija Reintentar compilación.
 - En el panel de navegación, elija Build projects. En la lista de proyectos de compilación, en la columna Name (Nombre), elija el enlace correspondiente al nombre del proyecto de compilación. En la lista de compilaciones, seleccione la casilla correspondiente a la compilación y después elija Reintentar compilación.

Note

De forma predeterminada, solo se muestran las 100 compilaciones o proyectos de compilación más recientes. Para ver más compilaciones o proyectos de compilación, elija el icono de rueda dentada y, a continuación, seleccione un valor diferente para Builds per page (Compilaciones por página) o Projects per page (Proyectos por página) o seleccione las flechas atrás y adelante.

Reintentar una compilación (AWS CLI)

- Ejecute el comando `retry-build`:

```
aws codebuild retry-build --id <build-id> --idempotency-token <idempotencyToken>
```

En el comando anterior, sustituya el siguiente marcador de posición:

- **<build-id>**: cadena obligatoria. ID de la compilación o compilación por lotes que se desea volver a intentar. Para obtener una lista de identificadores de compilación, consulte los siguientes temas:
 - [Ver una lista de identificadores de compilación \(AWS CLI\)](#)
 - [Ver una lista de identificadores de compilación por lotes \(AWS CLI\)](#)
 - [Ver una lista de identificadores de compilación de un proyecto de compilación \(AWS CLI\)](#)
 - [Ver una lista de identificadores de compilación por lotes de un proyecto de compilación \(AWS CLI\)](#)
- **--idempotency-token**: opcional. Si ejecuta el comando `retry-build` con la opción, se incluye un identificador único en el que se distingue entre mayúsculas y minúsculas, o token, con la solicitud de `retry-build`. El token es válido durante 5 minutos después de la solicitud. Si repite la solicitud `retry-build` con el mismo token, pero cambia un parámetro, CodeBuild devuelve un error de coincidencia de parámetros.

Volver a intentar una compilación (SDK de AWS)

Para obtener más información sobre cómo usar AWS CodeBuild con los SDK de AWS, consulte [Referencia de los SDK y las herramientas de AWS](#).

Ver una compilación en ejecución en Session Manager

En AWS CodeBuild, puedes pausar una compilación en ejecución y, a continuación, usar el Administrador de AWS Systems Manager sesiones para conectarte al contenedor de compilación y ver su estado.

Note

Esta característica no está disponible en entornos Windows.

Temas

- [Requisitos previos](#)
- [Pausar la compilación](#)
- [Iniciar la compilación](#)
- [Conectarse al contenedor de compilación](#)
- [Reanudar la compilación](#)

Requisitos previos

Para permitir el uso de Session Manager con la sesión de compilación, es necesario habilitar la conexión de sesión para la compilación. Hay dos prerequisites:

- CodeBuild Las imágenes seleccionadas estándar de Linux ya tienen el agente SSM instalado y el agente ContainerMode SSM activado.

Si va a utilizar una imagen personalizada para la compilación, haga lo siguiente:

1. Instalar el SSM Agent de . Para obtener más información, consulte [Instalación manual de SSM Agent en instancias EC2 de Linux](#) en la Guía del usuario de AWS Systems Manager . El agente SSM debe ser 3.0.1295.0 o posterior.
2. Copia el archivo <https://github.com/aws/aws-codebuild-docker-images/blob/master/ubuntu/standard/5.0/amazon-ssm-agent> .json en el directorio de la imagen. /etc/amazon/ssm/ Esto habilita el modo de contenedor en el agente SSM.

Note

Las imágenes personalizadas requieren el agente SSM más actualizado para que esta función funcione según lo esperado.

- CodeBuild El rol de servicio debe tener la siguiente política de SSM:

```
{
  "Effect": "Allow",
  "Action": [
    "ssmmessages:CreateControlChannel",
    "ssmmessages:CreateDataChannel",
    "ssmmessages:OpenControlChannel",
```

```

    "ssmmessages:OpenDataChannel"
  ],
  "Resource": "*"
}

```

Puede hacer que la CodeBuild consola asocie automáticamente esta política a su función de servicio al iniciar la compilación. Como alternativa, puede adjuntar la política al rol de servicio manualmente.

- Si tiene habilitada la auditoría y el registro de la actividad de la sesión en las preferencias de Systems Manager, la función de CodeBuild servicio también debe tener permisos adicionales. Los permisos varían según el lugar donde se almacenen los registros.

CloudWatch Registros

Si utiliza CloudWatch los registros para almacenar los registros, añada el siguiente permiso a la función CodeBuild de servicio:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "logs:DescribeLogGroups",
      "Resource": "arn:aws:logs:<region-id>:<account-id>:log-group:*:*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "arn:aws:logs:<region-id>:<account-id>:log-group:<log-group-
name>:*"
    }
  ]
}

```

Amazon S3

Si utiliza Amazon S3 para almacenar sus registros, añada el siguiente permiso al rol CodeBuild de servicio:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetEncryptionConfiguration",
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::<bucket-name>",
        "arn:aws:s3:::<bucket-name>/*"
      ]
    }
  ]
}
```

Para obtener más información, consulte [Actividad de auditoría y registro de sesión](#) en la Guía del usuario de AWS Systems Manager .

Pausar la compilación

Para pausar la compilación, inserte el comando `codebuild-breakpoint` en cualquiera de las fases de compilación del archivo de especificación de compilación. La compilación se pausará en este punto, lo que permite conectar con el contenedor de compilación y ver el contenedor en su estado actual.

Por ejemplo, añada lo siguiente a las fases de compilación del archivo de especificación de compilación.

```
phases:
  pre_build:
    commands:
      - echo Entered the pre_build phase...
      - echo "Hello World" > /tmp/hello-world
      - codebuild-breakpoint
```

Este código crea el archivo `/tmp/hello-world` y, a continuación, detiene la compilación en este punto.

Iniciar la compilación

Para permitir el uso de Session Manager con la sesión de compilación, es necesario habilitar las conexiones de sesión para la compilación. Para hacerlo, al iniciar la compilación, siga estos pasos:

1. Abra la AWS CodeBuild consola en <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. En el panel de navegación, elija Proyectos de compilación. Elija el proyecto de compilación y, a continuación, elija Iniciar compilación con anulaciones.
3. Seleccione Advanced build overrides (Sustitución avanzada de compilaciones).
4. En la sección Entorno, elija la opción Habilitar la conexión de sesión. Si no se selecciona esta opción, se ignoran todos los comandos de codebuild-breakpoint y codebuild-resume.
5. Realice los demás cambios que desee y elija Iniciar compilación.
6. Monitoriza el estado de la compilación en la consola. Cuando la sesión esté disponible, el enlace al AWS Session Manager aparecerá en la sección Estado de compilación.

Conectarse al contenedor de compilación

Es posible conectarse al contenedor de compilación de una de las dos formas siguientes:

CodeBuild consola

En un navegador web, abra el enlace de AWS Session Manager para conectarse al contenedor de compilación. Se abre una sesión de terminal que le permite explorar y controlar el contenedor de compilación.

AWS CLI

Note

Su máquina local debe tener instalado el complemento de Session Manager para realizar este procedimiento. Para obtener más información, consulte [Instalación del complemento Session Manager para la AWS CLI](#) en la Guía del AWS Systems Manager usuario.

1. Llame a la API de batch-get-builds con el ID de compilación para obtener información sobre la compilación, incluido el identificador de destino de la sesión. El nombre de la propiedad del identificador de destino de la sesión varía según el tipo de salida del comando aws. Por eso se añade `--output json` al comando.

```
aws codebuild batch-get-builds --ids <buildID> --region <region> --output json
```

2. Copie el valor de la propiedad `sessionTarget`. El nombre de la propiedad `sessionTarget` puede variar en función del tipo de salida del comando `aws`. Por eso se añade `--output json` al comando en el paso anterior.
3. Utilice el comando siguiente para conectarse al contenedor de compilación.

```
aws ssm start-session --target <sessionTarget> --region <region>
```

Para este ejemplo, comprueba que el archivo `/tmp/hello-world` exista y contiene el texto `Hello World`.

Reanudar la compilación

Después de examinar el contenedor de compilación, ejecute el comando `codebuild-resume` desde el intérprete de comandos del contenedor.

```
$ codebuild-resume
```

Eliminación de compilaciones en AWS CodeBuild

Puede utilizar AWS CLI o los SDK de AWS para eliminar compilaciones en AWS CodeBuild.

Eliminación de compilaciones (AWS CLI)

Ejecute el comando `batch-delete-builds`:

```
aws codebuild batch-delete-builds --ids ids
```

En el comando anterior, sustituya el siguiente marcador de posición:

- *ids*: cadena obligatoria. Los ID de las compilaciones que se van a eliminar. Para especificar varias compilaciones, separe cada ID de compilación con un espacio. Para obtener una lista de identificadores de compilación, consulte los siguientes temas:
 - [Ver una lista de identificadores de compilación \(AWS CLI\)](#)
 - [Ver una lista de identificadores de compilación de un proyecto de compilación \(AWS CLI\)](#)

En caso de éxito, aparece una matriz `buildsDeleted` en la salida, que contiene el Nombre de recurso de Amazon (ARN) de cada compilación eliminada correctamente. La información sobre las compilaciones que no se eliminaron correctamente aparece en la salida dentro de la matriz `buildsNotDeleted`.

Por ejemplo, si ejecuta este comando:

```
aws codebuild batch-delete-builds --ids my-demo-build-project:f8b888d2-5e1e-4032-8645-b115195648EX my-other-demo-build-project:a18bc6ee-e499-4887-b36a-8c90349c7eEX
```

En la salida se muestra información similar a la siguiente:

```
{
  "buildsNotDeleted": [
    {
      "id": "arn:aws:codebuild:us-west-2:123456789012:build/my-demo-build-project:f8b888d2-5e1e-4032-8645-b115195648EX",
      "statusCode": "BUILD_IN_PROGRESS"
    }
  ],
  "buildsDeleted": [
    "arn:aws:codebuild:us-west-2:123456789012:build/my-other-demo-build-project:a18bc6ee-e499-4887-b36a-8c90349c7eEX"
  ]
}
```

Eliminación de compilaciones (SDK de AWS)

Para obtener información sobre cómo usar AWS CodeBuild con los SDK de AWS, consulte [Referencia de los SDK y las herramientas de AWS](#).

Trabajar con AWS Lambda computación en AWS CodeBuild

AWS Lambda Compute ofrece velocidades de inicio optimizadas para sus compilaciones. AWS Lambda admite compilaciones más rápidas debido a una latencia de inicio más baja. AWS Lambda también se escala automáticamente, por lo que las compilaciones no tienen que hacer cola para ejecutarse. Sin embargo, hay algunos casos de uso que AWS Lambda no son compatibles y, si le afectan, utilice el procesamiento de EC2. Para obtener más información, consulte [Limitaciones de AWS Lambda cómputo](#).

Temas

- [¿Qué herramientas y entornos de tiempo de ejecución se incluyen en las imágenes de Docker del entorno de tiempo de ejecución seleccionado que se ejecuta en AWS Lambda?](#)
- [¿Qué pasa si la imagen seleccionada no incluye las herramientas que necesito?](#)
- [¿En qué regiones se admite la AWS Lambda computación CodeBuild?](#)
- [Limitaciones de AWS Lambda cómputo](#)
- [AWS Lambda calcule muestras con AWS CodeBuild](#)

¿Qué herramientas y entornos de tiempo de ejecución se incluyen en las imágenes de Docker del entorno de tiempo de ejecución seleccionado que se ejecuta en AWS Lambda?

AWS Lambda admite las siguientes herramientas: AWS CLI v2, AWS SAM CLI, git, go, Java, Node.js, Python, pip, Ruby y .NET.

¿Qué pasa si la imagen seleccionada no incluye las herramientas que necesito?

Si la imagen seleccionada no incluye las herramientas que necesitas, puedes proporcionar una imagen de Docker de entorno personalizado que incluya las herramientas necesarias.

Tenga en cuenta que necesita los siguientes permisos de Amazon ECR para utilizar imágenes personalizadas para la computación Lambda:

```
{
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "ecr:GetAuthorizationToken"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "ecr:BatchCheckLayerAvailability",
      "ecr:GetDownloadUrlForLayer",
      "ecr:BatchGetImage"
    ],
    "Resource": "arn:aws:ecr:image-region:image-account-id:repository/image-repo"
  }
]
```

Tenga en cuenta también que `curl` o `wget` debe estar instalado para poder utilizar imágenes personalizadas.

¿En qué regiones se admite la AWS Lambda computación CodeBuild?

En CodeBuild, la AWS Lambda computación se admite en las siguientes regiones Regiones de AWS: EE.UU. Este (Norte de Virginia), EE.UU. Este (Ohio), EE.UU. Oeste (Oregón), Asia Pacífico (Bombay), Asia Pacífico (Singapur), Asia Pacífico (Sídney), Asia Pacífico (Tokio), Europa (Fráncfort), Europa (Irlanda) y Sudamérica (São Paulo). Para obtener más información sobre Regiones de AWS dónde CodeBuild está disponible, consulta [AWS Servicios por región](#).

Limitaciones de AWS Lambda cómputo

Hay algunos casos de uso que AWS Lambda no son compatibles y, si le afectan, utilice el procesamiento de EC2:

- AWS Lambda no admite herramientas que requieren permisos de root. En el caso de herramientas como yum o rpm, utilice el tipo de computación EC2 u otras herramientas que no requieran permisos de root.
- AWS Lambda no admite compilaciones ni ejecuciones de Docker. Puede utilizar alternativas que no requieran permisos de root, como Podman.
- AWS Lambda no admite la escritura en archivos externos/tmp. Los administradores de paquetes incluidos están configurados para usar el directorio /tmp de forma predeterminada para descargar paquetes y hacer referencia a ellos.
- AWS Lambda no admite este tipo de entorno LINUX_GPU_CONTAINER y no es compatible con Windows Server Core 2019.
- AWS Lambda no admite el almacenamiento en caché, las compilaciones por lotes, los tiempos de espera de compilación personalizados, el tiempo de espera de las colas, las insignias de compilación, el modo privilegiado, los entornos de ejecución personalizados ni los tiempos de ejecución superiores a 15 minutos.
- AWS Lambda no admite la conectividad de VPC, un rango fijo de direcciones IP de CodeBuild origen, EFS, el control de versiones semántico, la instalación de certificados ni el acceso SSH con Session Manager.

AWS Lambda calcule muestras con AWS CodeBuild

Estos grupos de muestras se pueden usar para experimentar con la AWS Lambda computación interna CodeBuild.

Temas

- [Implemente una función Lambda con AWS SAM CodeBuild Lambda Java](#)
- [Cree una aplicación React de una sola página con CodeBuild Lambda Node.js](#)
- [Actualizar la configuración de una función Lambda con CodeBuild Lambda Python](#)

Implemente una función Lambda con AWS SAM CodeBuild Lambda Java

AWS Serverless Application Model (AWS SAM) es un marco de trabajo de código abierto para crear aplicaciones sin servidor. Para obtener más información, consulte el [AWS Serverless Application Model repositorio](#) en GitHub. En el ejemplo de Java siguiente, se usa Gradle para compilar y probar una función de AWS Lambda. Después de lo cual, la AWS SAM CLI se usa para implementar la

AWS CloudFormation plantilla y el paquete de implementación. Al usar CodeBuild Lambda, los pasos de compilación, prueba e implementación se gestionan automáticamente, lo que permite que la infraestructura se actualice rápidamente sin intervención manual en una sola compilación.

Configure su repositorio AWS SAM

Cree un AWS SAM Hello World proyecto mediante la AWS SAM CLI.

Para crear su AWS SAM proyecto

1. Siga las instrucciones de la Guía del AWS Serverless Application Model desarrollador para [instalar la AWS SAM CLI](#) en su máquina local.
2. Ejecute `sam init` y seleccione la siguiente configuración de proyecto.

```
Which template source would you like to use?: 1 - AWS Quick Start Templates
Choose an AWS Quick Start application template: 1 - Hello World Example
Use the most popular runtime and package type? (Python and zip) [y/N]: N
Which runtime would you like to use?: 8 - java21
What package type would you like to use?: 1 - Zip
Which dependency manager would you like to use?: 1 - gradle
Would you like to enable X-Ray tracing on the function(s) in your application? [y/N]: N
Would you like to enable monitoring using CloudWatch Application Insights? [y/N]: N
Would you like to set Structured Logging in JSON format on your Lambda functions? [y/N]: N
Project name [sam-app]: <insert project name>
```

3. Cargue la carpeta AWS SAM del proyecto en un repositorio de origen compatible. Para obtener una lista de los tipos de fuentes compatibles, consulte [ProjectSource](#).

Creación de un CodeBuild proyecto Java Lambda

Cree un proyecto Java AWS CodeBuild Lambda y configure los permisos de IAM necesarios para la compilación.

Para crear su proyecto CodeBuild Lambda Java

1. Abra la consola de AWS CodeBuild en <https://console.aws.amazon.com/codesuite/codebuild/home>.

2. Si aparece una página de CodeBuild información, elija Crear proyecto de compilación. De lo contrario, en el panel de navegación, expanda Compilar, elija Proyectos de compilación y, a continuación, elija Crear proyecto de compilación.
3. En Project name (Nombre de proyecto), escriba un nombre para este proyecto de compilación. Los nombres de los proyectos de compilación debe ser únicos en cada cuenta de AWS. También puede introducir una descripción opcional del proyecto de compilación para ayudar a otros usuarios a entender para qué se utiliza el proyecto.
4. En Fuente, seleccione el repositorio de origen en el que se encuentra el AWS SAM proyecto.
5. En Environment (Entorno):
 - Para Compute, selecciona Lambda.
 - Para Runtime (s), selecciona Java.
 - En Imagen, seleccione aws/codebuild/amazonlinux-x86_64-lambda-standard:corretto21.
 - En Función de servicio, deje seleccionada Nueva función de servicio. Anote el nombre del rol. Será necesario cuando actualice los permisos de IAM del proyecto más adelante en este ejemplo.
6. Elija Crear el proyecto de compilación.
7. Abra la consola de IAM en <https://console.aws.amazon.com/iam/>.
8. En el panel de navegación, elija Funciones y seleccione la función de servicio asociada a su proyecto. Para encontrar su función en el proyecto, seleccione CodeBuild el proyecto de construcción, elija Edición, Entorno y, a continuación, la función de servicio.
9. Elija la pestaña Relaciones de confianza y, a continuación, Editar política de confianza.
10. Añada la siguiente política en línea a su función de IAM. Esto se utilizará para implementar su AWS SAM infraestructura más adelante. Para más información, consulte [Adición y eliminación de permisos de identidad de IAM](#) en la Guía del usuario de IAM de .

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Action": [
        "cloudformation:*",
        "lambda:*",
        "iam:*",
```

```

        "apigateway:*",
        "s3:*"
    ],
    "Resource": [
        "*"
    ]
}
]
}

```

Configure el proyecto: buildspec

Para crear, probar e implementar la función Lambda, CodeBuild lee y ejecuta los comandos de compilación desde una especificación de compilación.

Para configurar su proyecto, buildspec

1. En la CodeBuild consola, selecciona tu proyecto de compilación y, a continuación, selecciona Editar y Buildspec.
2. En Buildspec, selecciona Insertar comandos de compilación y, a continuación, Cambiar al editor.
3. Elimine los comandos de compilación preconfigurados y pegue la siguiente especificación de compilación.

```

version: 0.2
env:
  variables:
    GRADLE_DIR: "HelloWorldFunction"
phases:
  build:
    commands:
      - echo "Running unit tests..."
      - cd $GRADLE_DIR; gradle test; cd ..
      - echo "Running build..."
      - sam build --template-file template.yaml
      - echo "Running deploy..."
      - sam package --output-template-file packaged.yaml --resolve-s3 --template-
file template.yaml
      - yes | sam deploy

```

4. Seleccione Update buildspec (Actualizar buildspec).

Implemente su AWS SAM infraestructura Lambda

Utilice CodeBuild Lambda para implementar automáticamente su infraestructura Lambda

Para implementar su infraestructura Lambda

1. Seleccione Iniciar la compilación. Esto creará, probará e implementará automáticamente la AWS SAM aplicación para AWS Lambda AWS CloudFormation utilizarla.
2. Una vez finalizada la compilación, diríjase a la AWS Lambda consola y busque la nueva función Lambda en el nombre del AWS SAM proyecto.
3. Pruebe su función Lambda seleccionando API Gateway en la descripción general de la función y, a continuación, haciendo clic en la URL del punto de enlace de la API. Debería ver una página abierta con el mensaje "message": "hello world".

Limpie su infraestructura

Para evitar cargos adicionales por los recursos que utilizó durante este tutorial, elimine los recursos creados por su AWS SAM plantilla y CodeBuild.

Para limpiar su infraestructura

1. Navegue hasta la AWS CloudFormation consola y seleccione `aws-sam-cli-managed-default`.
2. En Recursos, vacíe el depósito de despliegue `SamCliSourceBucket`.
3. Elimine la pila de `aws-sam-cli-managed-default`.
4. Elimina la AWS CloudFormation pila asociada a tu AWS SAM proyecto. Esta pila debe tener el mismo nombre que tu AWS SAM proyecto.
5. Ve a la CloudWatch consola y elimina los grupos de CloudWatch registros asociados a tu CodeBuild proyecto.
6. Ve a la CodeBuild consola y elimina tu CodeBuild proyecto seleccionando Eliminar proyecto de compilación.

Cree una aplicación React de una sola página con CodeBuild Lambda Node.js

[Create React App](#) es una forma de crear aplicaciones React de una sola página. El siguiente ejemplo de Node.js usa Node.js para crear los artefactos de origen a partir de la aplicación Create React y devuelve los artefactos de compilación.

Configura tu repositorio de fuentes y tu depósito de artefactos

Crea un repositorio de código fuente para tu proyecto con yarn and Create React App.

Para configurar el repositorio de fuentes y el depósito de artefactos

1. En tu máquina local, ejecuta `yarn create react-app <app-name>` para crear una aplicación React sencilla.
2. Sube la carpeta del proyecto de la aplicación React a un repositorio fuente compatible. Para obtener una lista de los tipos de fuentes compatibles, consulte [ProjectSource](#).

Crear un proyecto CodeBuild Lambda Node.js

Cree un AWS CodeBuild proyecto Lambda Node.js.

Para crear su proyecto CodeBuild Lambda Node.js

1. Abra la consola de AWS CodeBuild en <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Si aparece una página de CodeBuild información, elija Crear proyecto de compilación. De lo contrario, en el panel de navegación, expanda Compilar, elija Proyectos de compilación y, a continuación, elija Crear proyecto de compilación.
3. En Project name (Nombre de proyecto), escriba un nombre para este proyecto de compilación. Los nombres de los proyectos de compilación debe ser únicos en cada cuenta de AWS. También puede introducir una descripción opcional del proyecto de compilación para ayudar a otros usuarios a entender para qué se utiliza el proyecto.
4. En Fuente, seleccione el repositorio de origen en el que se encuentra el AWS SAM proyecto.
5. En Environment (Entorno):
 - Para Compute, selecciona Lambda.

- Para Runtime (s), seleccione Node.js.
 - En Imagen, seleccione aws/codebuild/amazonlinux-x86_64-lambda-standard:nodejs20.
6. En Artifacts (Artefactos):
 - En Tipo, seleccione Amazon S3.
 - En el nombre del bucket, seleccione el bucket de artefactos del proyecto que creó anteriormente.
 - Para empaquetar artefactos, selecciona Zip.
 7. Elija Crear el proyecto de compilación.

Configura el proyecto (buildspec)

Para crear tu aplicación React, CodeBuild lee y ejecuta los comandos de compilación desde un archivo buildspec.

Para configurar tu proyecto, buildspec

1. En la CodeBuild consola, selecciona tu proyecto de compilación y, a continuación, selecciona Editar y Buildspec.
2. En Buildspec, selecciona Insertar comandos de compilación y, a continuación, Cambiar al editor.
3. Elimine los comandos de compilación preconfigurados y pegue la siguiente especificación de compilación.

```
version: 0.2
phases:
  build:
    commands:
      - yarn
      - yarn add --dev jest-junit @babel/plugin-proposal-private-property-in-object
      - yarn run build
      - yarn run test -- --coverage --watchAll=false --testResultsProcessor="jest-junit" --detectOpenHandles
artifacts:
  name: "build-output"
  files:
    - "**/*"
reports:
  test-report:
```

```
files:
  - 'junit.xml'
file-format: 'JUNITXML'
coverage-report:
  files:
    - 'coverage/clover.xml'
  file-format: 'CLOVERXML'
```

4. Seleccione Update buildspec (Actualizar buildspec).

Crea y ejecuta tu aplicación React

Cree la aplicación React en CodeBuild Lambda, descargue los artefactos de compilación y ejecute la aplicación React localmente.

Para compilar y ejecutar tu aplicación React

1. Seleccione Iniciar la compilación.
2. Una vez finalizada la compilación, navegue hasta el depósito de artefactos del proyecto de Amazon S3 y descargue el artefacto de la aplicación React.
3. Descomprime el artefacto de compilación de React y colócalo `run npm install -g serve && serve -s build` en la carpeta del proyecto.
4. El `serve` comando mostrará el sitio estático en un puerto local e imprimirá el resultado en su terminal. Puedes visitar la URL de localhost que aparece `Local`: en la salida del terminal para ver tu aplicación React.

Para obtener más información sobre cómo gestionar el despliegue de un servidor basado en React, consulta [Crear el despliegue de una aplicación React](#).

Limpie su infraestructura

Para evitar cargos adicionales por los recursos que utilizó durante este tutorial, elimine los recursos creados para su CodeBuild proyecto.

Para limpiar su infraestructura

1. Elimine los artefactos de su proyecto (bucket de Amazon S3)
2. Diríjase a la CloudWatch consola y elimine los grupos de CloudWatch registros asociados a su CodeBuild proyecto.

3. Ve a la CodeBuild consola y elimina tu CodeBuild proyecto seleccionando Eliminar proyecto de compilación.

Actualizar la configuración de una función Lambda con CodeBuild Lambda Python

En el siguiente ejemplo de Python, se utilizan [Boto3](#) y Lambda CodeBuild Python para actualizar la configuración de una función Lambda. Este ejemplo se puede ampliar para gestionar otros recursos mediante programación. AWS Para obtener más información, consulte la documentación de [Boto3](#).

Requisitos previos

Cree o busque una función Lambda en su cuenta.

En este ejemplo se supone que ya ha creado una función Lambda en su cuenta y que la utilizará CodeBuild para actualizar las variables de entorno de la función Lambda. Para obtener más información sobre cómo configurar una función Lambda mediante CodeBuild, consulte el [Implemente una función Lambda con AWS SAM CodeBuild Lambda Java](#) ejemplo o visite. [AWS Lambda](#)

Configura tu repositorio de código fuente

Crea un repositorio de código fuente para almacenar tu script de Python de Boto3.

Para configurar el repositorio de código fuente

1. Copie el siguiente script de Python en un nuevo archivo llamado `update_lambda_environment_variables.py`.

```
import boto3
from os import environ

def update_lambda_env_variable(lambda_client):
    lambda_function_name = environ['LAMBDA_FUNC_NAME']
    lambda_env_variable = environ['LAMBDA_ENV_VARIABLE']
    lambda_env_variable_value = environ['LAMBDA_ENV_VARIABLE_VALUE']
    print("Updating lambda function " + lambda_function_name + " environment
variable "
        + lambda_env_variable + " to " + lambda_env_variable_value)
    lambda_client.update_function_configuration(
```

```
        FunctionName=lambda_function_name,
        Environment={
            'Variables': {
                lambda_env_variable: lambda_env_variable_value
            }
        },
    )

if __name__ == "__main__":
    region = environ['AWS_REGION']
    client = boto3.client('lambda', region)
    update_lambda_env_variable(client)
```

2. Cargue el archivo python en un repositorio de código fuente compatible. Para obtener una lista de los tipos de fuentes compatibles, consulte [ProjectSource](#).

Crear un proyecto de CodeBuild Python Lambda

Cree un proyecto de CodeBuild Python Lambda.

Para crear su proyecto CodeBuild Lambda Java

1. Abra la consola de AWS CodeBuild en <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Si aparece una página de CodeBuild información, elija Crear proyecto de compilación. De lo contrario, en el panel de navegación, expanda Compilar, elija Proyectos de compilación y, a continuación, elija Crear proyecto de compilación.
3. En Project name (Nombre de proyecto), escriba un nombre para este proyecto de compilación. Los nombres de los proyectos de compilación debe ser únicos en cada cuenta de AWS. También puede introducir una descripción opcional del proyecto de compilación para ayudar a otros usuarios a entender para qué se utiliza el proyecto.
4. En Fuente, seleccione el repositorio de origen en el que se encuentra el AWS SAM proyecto.
5. En Environment (Entorno):
 - Para Compute, selecciona Lambda.
 - Para Runtime (s), selecciona Python.
 - En Imagen, seleccione `aws/codebuild/amazonlinux-x86_64-lambda-standard:python3.12`.

- En Función de servicio, deje seleccionada Nueva función de servicio. Anote el nombre del rol. Esto será necesario cuando actualice los permisos de IAM del proyecto más adelante en este ejemplo.
6. Elija Crear el proyecto de compilación.
 7. Abra la consola de IAM en <https://console.aws.amazon.com/iam/>.
 8. En el panel de navegación, elija Funciones y seleccione la función de servicio asociada a su proyecto. Para encontrar su función en el proyecto, seleccione CodeBuild el proyecto de construcción, elija Edición, Entorno y, a continuación, la función de servicio.
 9. Elija la pestaña Relaciones de confianza y, a continuación, Editar política de confianza.
 10. Añada la siguiente política en línea a su función de IAM. Esto se utilizará para implementar su AWS SAM infraestructura más adelante. Para más información, consulte [Adición y eliminación de permisos de identidad de IAM](#) en la Guía del usuario de IAM de .

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "UpdateLambdaPermissions",
      "Effect": "Allow",
      "Action": [
        "lambda:UpdateFunctionConfiguration"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

Configure el proyecto: buildspec

Para actualizar la función Lambda, el script lee las variables de entorno de la especificación de compilación para buscar el nombre de la función Lambda, el nombre de la variable de entorno y el valor de la variable de entorno.

Para configurar su proyecto, buildspec

1. En la CodeBuild consola, selecciona tu proyecto de compilación y, a continuación, selecciona Editar y Buildspec.
2. En Buildspec, selecciona Insertar comandos de compilación y, a continuación, Cambiar al editor.
3. Elimine los comandos de compilación preconfigurados y pegue la siguiente especificación de compilación.

```
version: 0.2
env:
  variables:
    LAMBDA_FUNC_NAME: "<lambda-function-name>"
    LAMBDA_ENV_VARIABLE: "FEATURE_ENABLED"
    LAMBDA_ENV_VARIABLE_VALUE: "true"
phases:
  install:
    commands:
      - pip3 install boto3
  build:
    commands:
      - python3 update_lambda_environment_variables.py
```

4. Seleccione Update buildspec (Actualizar buildspec).

Actualice la configuración de Lambda

Utilice CodeBuild Lambda Python para actualizar automáticamente la configuración de la función Lambda.

Para actualizar la configuración de la función Lambda

1. Seleccione Iniciar la compilación.
2. Una vez finalizada la compilación, navegue hasta la función Lambda.
3. Seleccione Configuración y, a continuación, Variables de entorno. Debería ver una nueva variable de entorno con la clave FEATURE_ENABLED y el valor true.

Limpie su infraestructura

Para evitar cargos adicionales por los recursos que utilizó durante este tutorial, elimine los recursos creados para su CodeBuild proyecto.

Para limpiar su infraestructura

1. Ve a la CloudWatch consola y elimina los grupos de CloudWatch registros asociados a tu CodeBuild proyecto.
2. Ve a la CodeBuild consola y elimina tu CodeBuild proyecto seleccionando Eliminar proyecto de compilación.
3. Si ha creado una función Lambda para los fines de este ejemplo, elija Acciones y Eliminar función para limpiar la función Lambda.

Extensiones

Si desea ampliar este ejemplo para gestionar otros AWS recursos mediante AWS CodeBuild Lambda Python:

- Actualice el script de Python para modificar los nuevos recursos con Boto3.
- Actualice el rol de IAM asociado a su CodeBuild proyecto para tener permisos para los nuevos recursos.
- Agrega cualquier variable de entorno nueva asociada a los nuevos recursos a tu especificación de compilación.

Trabajando con capacidad reservada en AWS CodeBuild

CodeBuild ofrece las siguientes flotas informáticas:

- Flotas bajo demanda
- Flotas de capacidad reservada

Con flotas bajo demanda, CodeBuild proporciona procesamiento para sus compilaciones. Las máquinas se destruyen cuando finaliza la compilación. Las flotas bajo demanda están completamente administradas e incluyen capacidades de escalado automático para gestionar los picos de demanda.

Note

Las flotas bajo demanda no son compatibles con Windows Server 2022.

CodeBuild también ofrece flotas de capacidad reservada que contienen instancias impulsadas por Amazon EC2 que son mantenidas por CodeBuild. Con las flotas de capacidad reservada, configura un conjunto de instancias dedicadas para su entorno de compilación. Estas máquinas permanecen inactivas, listas para procesar compilaciones o pruebas de forma inmediata y reducen la duración de las compilaciones. Con flotas de capacidad reservada, sus máquinas estarán siempre en funcionamiento y seguirán incurriendo en costos mientras estén aprovisionadas.

Important

Independientemente del tiempo durante el que ejecute una instancia, las flotas de capacidad reservada incurren en un cargo inicial por instancia, después del cual puede haber costos adicionales asociados. Para obtener más información, consulte <https://aws.amazon.com/codebuild/pricing/>.

Temas

- [¿Cómo comienzo a utilizar las flotas de capacidad reservada?](#)
- [Prácticas recomendadas](#)

- [¿Puedo compartir una flota de capacidad reservada en varios CodeBuild proyectos?](#)
- [¿Qué regiones admiten flotas con capacidad reservada?](#)
- [Propiedades de flota con capacidad reservada](#)
- [Muestras de capacidad reservadas con AWS CodeBuild](#)
- [Limitaciones de las flotas de capacidad reservada](#)

¿Cómo comienzo a utilizar las flotas de capacidad reservada?

Para crear una flota de capacidad reservada

1. [Inicie sesión en la AWS CodeBuild consola AWS Management Console y ábrala en https://console.aws.amazon.com/codesuite/codebuild/home.](https://console.aws.amazon.com/codesuite/codebuild/home)
2. En el panel de navegación, elija Flotas de computación y, a continuación, Crear flota de computación.
3. En el campo de texto Nombre de la flota de computación, introduzca el nombre de su flota.
4. En el menú desplegable Sistema operativo, elija el sistema operativo.
5. En el menú desplegable Arquitectura, elija la arquitectura.
6. En el menú desplegable Computación, elija el tipo de máquina de computación para su máquina.
7. En el campo de texto Capacidad, introduzca el número mínimo de instancias de la flota.
8. En el campo Comportamiento de desbordamiento, selecciona el comportamiento cuando la demanda supere la capacidad de la flota. Para obtener más información sobre estas opciones, consulte [Propiedades de flota con capacidad reservada](#).
9. Elija Crear flota de computación.
10. Una vez creada la flota de procesamiento, cree un CodeBuild proyecto nuevo o edite uno existente. En Entorno, elija Capacidad reservada en el Modelo de aprovisionamiento y, a continuación, elija la flota especificada en Nombre de flota.

Prácticas recomendadas

Cuando utilice flotas de capacidad reservada, recomendamos que siga estas prácticas recomendadas.

- Recomendamos utilizar el modo de caché de origen para ayudar a mejorar el rendimiento de la compilación mediante el almacenamiento en caché del origen.
- Recomendamos utilizar el almacenamiento en caché de capas de Docker para mejorar el rendimiento de la compilación mediante el almacenamiento en caché de las capas de Docker existentes.

¿Puedo compartir una flota de capacidad reservada en varios CodeBuild proyectos?

Sí, puede maximizar la utilización de la capacidad de una flota utilizándola en varios proyectos.

¿Qué regiones admiten flotas con capacidad reservada?

Se admiten flotas con capacidad reservada en las siguientes regiones de AWS: EE.UU. Este (Norte de Virginia), EE.UU. Este (Ohio), EE.UU. Oeste (Oregón), Asia Pacífico (Bombay), Asia Pacífico (Singapur), Asia Pacífico (Sídney), Asia Pacífico (Tokio), Europa (Fráncfort), Europa (Irlanda) y Sudamérica (São Paulo). Para obtener más información sobre Regiones de AWS donde CodeBuild está disponible, consulte [AWS Servicios por región](#).

Propiedades de flota con capacidad reservada

Una flota de capacidad reservada contiene las siguientes propiedades:

Sistema operativo

El sistema operativo. A continuación, se muestran los sistemas operativos disponibles:

- Amazon Linux
- Windows Server 2019
- Windows Server 2022

Arquitectura

Arquitectura del procesador. Están disponibles las siguientes arquitecturas:

- x86_64
- Arm64

Cálculo

El tipo de máquina de computación de cada instancia. Dispone de los siguientes tipos de máquina:

Tipo de computación	Valor de computeType del entorno	Valor de tipo del entorno	Memoria	vCPU	Espacio en disco
ARM Small	BUILD_GENERAL1_SMALL	ARM_CONTAINER	4 GB	2	50 GB
ARM Large	BUILD_GENERAL1_LARGE	ARM_CONTAINER	16 GB	8	50 GB
Linux Small ¹	BUILD_GENERAL1_SMALL	LINUX_CONTAINER	3 GB	2	64 GB
Linux Medium ¹	BUILD_GENERAL1_MEDIUM	LINUX_CONTAINER	7 GB	4	128 GB
Linux Large ¹	BUILD_GENERAL1_LARGE	LINUX_CONTAINER	15 GB	8	128 GB
Linux XLarge	BUILD_GENERAL1_XLARGE	LINUX_CONTAINER	70 GB	36	256 GB
Linux 2XLarge	BUILD_GENERAL1_2XLARGE	LINUX_CONTAINER	145 GB	72	824 GB (SSD)

Tipo de computación	Valor de computeType del entorno	Valor de tipo del entorno	Memoria	vCPU	Espacio en disco
Linux GPU Small	BUILD_GENERAL1_SMALL	LINUX_GPU_CONTAINER	16 GB	4	220 GB
Linux GPU Large	BUILD_GENERAL1_LARGE	LINUX_GPU_CONTAINER	255 GB	32	50 GB
Windows Medium	BUILD_GENERAL1_MEDIUM	WINDOWS_SERVER_2019_CONTAINER	7 GB	4	128 GB
Windows Medium	BUILD_GENERAL1_MEDIUM	WINDOWS_SERVER_2022_CONTAINER	7 GB	4	128 GB
Windows Large	BUILD_GENERAL1_LARGE	WINDOWS_SERVER_2019_CONTAINER	15 GB	8	128 GB
Windows Large	BUILD_GENERAL1_LARGE	WINDOWS_SERVER_2022_CONTAINER	15 GB	8	128 GB

Capacidad

El número inicial de máquinas asignadas a la flota, que define el número de compilaciones que pueden funcionar en paralelo.

Comportamiento de desbordamiento

Define el comportamiento cuando el número de compilaciones supera la capacidad de la flota.

Bajo demanda

Las compilaciones de Overflow se ejecutan bajo CodeBuild demanda.

Important

Si decide configurar su comportamiento de desbordamiento como bajo demanda, tenga en cuenta que las compilaciones de desbordamiento se facturarán por separado, de forma similar a Amazon EC2 bajo demanda. Para obtener más información, consulte <https://aws.amazon.com/codebuild/pricing/>.

Cola

Las ejecuciones de compilaciones se colocan en una cola hasta que haya una máquina disponible. Esto limita los costos adicionales, ya que no se asignan máquinas adicionales.

Muestras de capacidad reservadas con AWS CodeBuild

Estos ejemplos se pueden utilizar para experimentar con flotas de capacidad reservada. CodeBuild

Temas

- [Ejemplo de almacenamiento en caché con capacidad reservada](#)

Ejemplo de almacenamiento en caché con capacidad reservada

Una caché puede almacenar fragmentos reutilizables del entorno de compilación y utilizarlos en diferentes compilaciones. En este ejemplo se muestra cómo habilitar el almacenamiento en caché dentro de su proyecto de compilación utilizando la capacidad reservada. Para obtener más información, consulte [Almacenamiento en caché de compilaciones de AWS CodeBuild](#).

Puede empezar por especificar uno o varios modos de caché en la configuración de su proyecto:

Cache:

Type: LOCAL

Modes:

- LOCAL_CUSTOM_CACHE
- LOCAL_DOCKER_LAYER_CACHE
- LOCAL_SOURCE_CACHE

 Note

Asegúrese de habilitar el modo con privilegios para utilizar la caché de capas de Docker.

La configuración de la especificación de compilación de su proyecto debe tener el aspecto siguiente:

```
version: 0.2
  phases:
    build:
      commands:
        - echo testing local source cache
        - touch /codebuild/cache/workspace/foobar.txt
        - git checkout -b cached_branch
        - echo testing local docker layer cache
        - docker run alpine:3.14 2>&1 | grep 'Pulling from' || exit 1
        - echo testing local custom cache
        - touch foo
        - mkdir bar && ln -s foo bar/foo2
        - mkdir bar/bar && touch bar/bar/foo3 && touch bar/bar/foo4
        - "[ -f foo ] || exit 1"
        - "[ -L bar/foo2 ] || exit 1"
        - "[ -f bar/bar/foo3 ] || exit 1"
        - "[ -f bar/bar/foo4 ] || exit 1"
      cache:
        paths:
          - './foo'
          - './bar/**/*'
          - './bar/bar/foo3'
```

Puede empezar por ejecutar una compilación con el nuevo proyecto para propagar la caché. Una vez hecho esto, debe empezar otra compilación con una especificación de compilación de anulación, similar a la siguiente:

```
version: 0.2
  phases:
```

```
build:
  commands:
    - echo testing local source cache
    - git branch | if grep 'cached_branch'; then (exit 0); else (exit 1); fi
    - ls /codebuild/cache/workspace | if grep 'foobar.txt'; then (exit 0); else
(exit 1); fi
    - echo testing local docker layer cache
    - docker run alpine:3.14 2>&1 | if grep 'Pulling from'; then (exit 1); else
(exit 0); fi
    - echo testing local custom cache
    - "[ -f foo ] || exit 1"
    - "[ -L bar/foo2 ] || exit 1"
    - "[ -f bar/bar/foo3 ] || exit 1"
    - "[ -f bar/bar/foo4 ] || exit 1"
  cache:
    paths:
      - './foo'
      - './bar/**/*'
      - './bar/bar/foo3'
```

Limitaciones de las flotas de capacidad reservada

Hay algunos casos de uso que las flotas de capacidad reservada no admiten. En caso de que le afectan, utilice flotas bajo demanda:

- Las flotas de capacidad reservada no admiten compilaciones por lotes, métricas de utilización de compilaciones ni control de versiones semántico.
- Las flotas de capacidad reservada no admiten la conectividad de VPC.

Para obtener más información sobre los límites y las cuotas, consulte [Flotas de computación](#).

Trabajar con informes de pruebas en AWS CodeBuild

Puede crear informes CodeBuild que contengan detalles sobre las pruebas que se ejecutan durante las compilaciones. Puede crear pruebas como pruebas unitarias, pruebas de configuración y pruebas funcionales.

Se admiten los siguientes formatos de archivo de informe de prueba:

- Cucumber JSON (.json)
- JUnit XML (.xml)
- NUnit XML (.xml)
- NUnit3 XML (.xml)
- TestNG XML (.xml)
- Visual Studio TRX (.trx)
- XML de Visual Studio TRX (.xml)

Note

La última versión compatible de `cucumber-js` es la 7.3.2.

Cree sus casos de prueba con cualquier marco de pruebas que pueda crear archivos de informes en uno de esos formatos (por ejemplo, el complemento Surefire JUnit, TestNG o Cucumber).

Para crear un informe de pruebas, se añade un nombre de grupo de informes al archivo buildspec de un proyecto de compilación con información sobre los casos de prueba. Cuando ejecuta el proyecto de compilación, se ejecutan los casos de prueba y se crea un informe de pruebas. No es necesario crear un grupo de informes antes de ejecutar las pruebas. Si especifica un nombre de grupo de informes, CodeBuild crea un grupo de informes para usted cuando ejecute los informes. Si desea utilizar un grupo de informes que ya exista, especifique su ARN en el archivo buildspec.

Puede utilizar un informe de pruebas para solucionar un problema durante una ejecución de compilación. Si tiene muchos informes de pruebas de varias compilaciones de un proyecto de compilación, puede utilizar los informes de pruebas para ver tendencias y tasas de pruebas y errores como ayuda para optimizar las compilaciones.

Un informe caduca 30 días después de su creación. No puede ver un informe de pruebas caducado. Si desea conservar los informes de pruebas durante más de 30 días, puede exportar los archivos de datos sin procesar de los resultados de prueba a un bucket de Amazon S3. Los archivos de prueba exportados no caducan. La información sobre el bucket de S3 se especifica al crear el grupo de informes.

Note

El rol CodeBuild de servicio especificado en el proyecto se usa para obtener permisos de carga en el bucket de S3.

Temas

- [Crear un informe de pruebas](#)
- [Uso de grupos de informes](#)
- [Uso de informes](#)
- [Uso de permisos de informes de pruebas](#)
- [Consultar los informes de pruebas](#)
- [Generación de informes de pruebas con marcos de trabajo de pruebas](#)
- [Informes de cobertura de código](#)
- [Detección automática de informes](#)

Crear un informe de pruebas

Para crear un informe de pruebas, se ejecuta un proyecto de compilación que se configura con uno a cinco grupos de informes en su archivo buildspec. Se crea un informe de pruebas durante la ejecución. Contiene los resultados de los casos de prueba que se especifican para los grupos de informes. Se genera un nuevo informe de pruebas para cada compilación posterior que utilice el mismo archivo buildspec.

Para crear un informe de pruebas

1. Cree un proyecto de compilación. Para obtener más información, consulte [Creación de un proyecto de compilación en AWS CodeBuild](#).
2. Configure el archivo buildspec del proyecto con información de informe de pruebas

- a. Añada una sección `reports:` y especifique el ARN de un grupo de informes existente o el nombre de un nuevo grupo de informes.

Si especifica un ARN, CodeBuild utiliza ese grupo de informes.

Si especifica un nombre, CodeBuild crea un grupo de informes para usted con el nombre de su proyecto y el nombre que especificó, en el formato `<project-name>- <report-group-name >`. Si el grupo de informes nombrado ya existe, CodeBuild utiliza ese grupo de informes.

- b. En el grupo de informes, especifique la ubicación de los archivos que contienen los resultados de las pruebas. Si utiliza más de un grupo de informes, especifique las ubicaciones de los archivos de resultados de prueba para cada uno de ellos. Cada vez que se ejecuta el proyecto de compilación, se crea un nuevo informe de pruebas. Para obtener más información, consulte [Especificar archivos de prueba](#).
- c. En la sección `commands` de la secuencia `build` o `post_build`, especifique los comandos que ejecutan los casos de prueba que ha especificado para sus grupos de informes. Para obtener más información, consulte [Especificar comandos de prueba](#).

El siguiente es un ejemplo de una sección `reports` de la especificación de compilación:

```
reports:
  php-reports:
    files:
      - "reports/php/*.xml"
    file-format: "JUNITXML"
  nunit-reports:
    files:
      - "reports/nunit/*.xml"
    file-format: "NUNITXML"
```

3. Ejecute una compilación del proyecto de compilación. Para obtener más información, consulte [Ejecutar una compilación en AWS CodeBuild](#).
4. Cuando la compilación se haya completado, elija la nueva ejecución de la compilación en Build history (Historial de compilación) en la página del proyecto. Seleccione Reports (Informes) para ver el informe de pruebas. Para obtener más información, consulte [Consultar los informes de pruebas de una compilación](#).

Uso de grupos de informes

Un grupo de informes contiene informes de pruebas y especifica la configuración compartida. Utilice el archivo `buildspec` para especificar los casos de prueba que se van a ejecutar y los comandos para ejecutarlos cuando se compila. Para cada grupo de informes configurado en un proyecto de compilación, una ejecución del proyecto de compilación crea un informe de pruebas. Varias ejecuciones de un proyecto de compilación configurado con un grupo de informes crean varios informes de pruebas en ese grupo de informes, cada uno con resultados de los mismos casos de prueba especificados para dicho grupo de informes.

Los casos de prueba se especifican para un grupo de informes en el archivo `buildspec` de un proyecto de compilación. Puede especificar hasta cinco grupos de informes en un proyecto de compilación. Cuando ejecuta una compilación, se ejecutan todos los casos de prueba. Se crea un nuevo informe de pruebas con los resultados de cada caso de prueba especificado para un grupo de informes. Cada vez que se ejecuta una nueva compilación, se ejecutan los casos de prueba y se crea un nuevo informe de pruebas con los nuevos resultados de prueba.

Los grupos de informes se pueden utilizar en varios proyectos de compilación. Todos los informes de pruebas creados con un grupo de informes comparten la misma configuración, como su opción de exportación y permisos, incluso si los informes de prueba se crean utilizando diferentes proyectos de compilación. Los informes de pruebas creados con un grupo de informes en varios proyectos de compilación pueden contener los resultados de la ejecución de diferentes conjuntos de casos de prueba (un conjunto de casos de prueba para cada proyecto de compilación). Esto se debe a que puede especificar diferentes archivos de casos de prueba para el grupo de informes en el archivo `buildspec` de cada proyecto. También puede cambiar los archivos de casos de prueba de un grupo de informes en un proyecto de compilación si edita su archivo `buildspec`. Las ejecuciones de compilación posteriores crean nuevos informes de pruebas que contienen los resultados de los archivos de casos de prueba en el archivo `buildspec` actualizado.

Temas

- [Crear un grupo de informes](#)
- [Actualizar un grupo de informes](#)
- [Especificar archivos de prueba](#)
- [Especificar comandos de prueba](#)
- [Nomenclatura de grupos de informes](#)
- [Etiquetado de grupos de informes en AWS CodeBuild](#)

- [Usar grupos de informes compartidos](#)

Crear un grupo de informes

Puede usar la CodeBuild consola AWS CLI, el archivo Buildspec o un archivo buildspec para crear un grupo de informes. Su rol de IAM debe tener los permisos necesarios para crear un grupo de informes. Para obtener más información, consulte [Uso de permisos de informes de pruebas](#).

Temas

- [Crear un grupo de informes \(buildspec\)](#)
- [Crear un grupo de informes \(consola\)](#)
- [Crear un grupo de informes \(CLI\)](#)
- [Crear un grupo de informes \(AWS CloudFormation\)](#)

Crear un grupo de informes (buildspec)

Un grupo de informes creado con buildspec no exporta archivos de resultados de prueba sin procesar. Puede ver el grupo de informes y especificar las opciones de exportación. Para obtener más información, consulte [Actualizar un grupo de informes](#).

Para crear un grupo de informes mediante un archivo buildspec

1. Elija un nombre de grupo de informes que no esté asociado a ningún grupo de informes de su cuenta. AWS
2. Configure la sección `reports` del archivo buildspec con este nombre. En este ejemplo, el nombre del grupo de informes es `new-report-group` y los casos de prueba de uso se crean con el marco JUnit:

```
reports:
  new-report-group: #surefire junit reports
    files:
      - '**/*'
    base-directory: 'surefire/target/surefire-reports'
```

El nombre del grupo de informes también se puede especificar mediante variables de entorno en la especificación de compilación:

```
version: 0.2
env:
  variables:
    REPORT_GROUP_NAME: "new-report-group"
phases:
  build:
    commands:
      - ...
...
reports:
  $REPORT_GROUP_NAME:
    files:
      - '**/*'
    base-directory: 'surefire/target/surefire-reports'
```

Para obtener más información, consulte [Especificar archivos de prueba](#) y [Reports syntax in the buildspec file](#).

3. En la sección `commands`, especifique el comando para ejecutar las pruebas. Para obtener más información, consulte [Especificar comandos de prueba](#).
4. Ejecute la compilación. Cuando se completa la compilación, se crea un nuevo grupo de informes con un nombre que utiliza el formato `project-name-report-group-name`. Para obtener más información, consulte [Nomenclatura de grupos de informes](#).

Crear un grupo de informes (consola)

Para crear un informe de pruebas

1. Abra la consola de AWS CodeBuild en <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. En el panel de navegación, elija Report groups (Grupos de informes).
3. Elija Create report group (Crear grupo de informes).
4. En Report group name (Nombre del grupo de informes), escriba un nombre para el grupo de informes.
5. (Opcional) En Tags (Etiquetas), escriba el nombre y el valor de las etiquetas que desea que usen los servicios compatibles de AWS. Use Add row para añadir una etiqueta. Puede añadir hasta 50 etiquetas.

6. Si desea cargar los datos sin procesar de los resultados del informe de pruebas a un bucket de Amazon S3:
 - a. Seleccione Exportar a Amazon S3.
 - b. En S3 bucket name (Nombre de bucket de S3), escriba el nombre del bucket de S3.
 - c. (Opcional) Para el propietario del bucket de S3, introduzca el identificador de cuenta de AWS correspondiente a la cuenta propietaria del bucket de S3. Esto permite exportar los datos de los informes a un bucket de Amazon S3 que es propiedad de una cuenta distinta de la cuenta que ejecuta la compilación.
 - d. En Path prefix (Prefijo de ruta), introduzca la ruta en el bucket de S3 donde desea cargar los resultados de prueba.
 - e. Seleccione Compress test result data in a zip file (Comprimir los datos de los resultados de prueba en un archivo zip) para comprimir los archivos de datos de resultados de prueba sin procesar.
 - f. Expanda Additional configuration (Configuración adicional) para mostrar las opciones de cifrado. Elija una de las siguientes opciones:
 - Clave de AWS administrada predeterminada para usar una Clave administrada de AWS para Amazon S3. Para obtener más información, consulte [CMK administradas por el cliente](#) en la Guía del usuario de AWS Key Management Service. Esta es la opción de cifrado predeterminada.
 - Seleccione una clave personalizada para usar una clave administrada por el cliente creada y configurada por usted. Como clave de cifrado de AWS KMS, introduzca el ARN de la clave de cifrado. Su formato es el siguiente: `arn:aws:kms:<region-id>:<aws-account-id>:key/<key-id>` . Para obtener más información, consulte [Creación de claves de KMS](#) en la Guía del usuario de AWS Key Management Service.
 - Disable artifact encryption (Desactivar el cifrado de artefactos) para desactivar el cifrado. Puede elegir esta opción si desea compartir los resultados de sus pruebas o publicarlos en un sitio web estático. (Un sitio web dinámico puede ejecutar código para descifrar los resultados de prueba).

Para obtener más información sobre el cifrado de datos en reposo, consulte [Cifrado de datos](#).

Note

El rol de servicio de CodeBuild especificado en el proyecto se utiliza para los permisos para cargar en el bucket de S3.

7. Elija Create report group (Crear grupo de informes).

Crear un grupo de informes (CLI)

Para crear un grupo de informes

1. Cree un archivo denominado `CreateReportGroup.json`.
2. Según sus requisitos, copie uno de los siguientes fragmentos de código JSON en `CreateReportGroup.json`:
 - Utilice el código JSON siguiente para especificar que su grupo de informes de pruebas exporta archivos de resultados de prueba sin procesar a un bucket de Amazon S3.

```
{
  "name": "<report-name>",
  "type": "TEST",
  "exportConfig": {
    "exportConfigType": "S3",
    "s3Destination": {
      "bucket": "<bucket-name>",
      "bucketOwner": "<bucket-owner>",
      "path": "<path>",
      "packaging": "NONE | ZIP",
      "encryptionDisabled": "false",
      "encryptionKey": "<your-key>"
    },
    "tags": [
      {
        "key": "tag-key",
        "value": "tag-value"
      }
    ]
  }
}
```

- Reemplace `<bucket-name>` por el nombre del bucket de Amazon S3 y `<path>` por la ruta de acceso del bucket al lugar donde desea exportar los archivos.
- Si desea comprimir los archivos exportados, para `packaging`, especifique ZIP. De lo contrario, especifique NONE.
- `bucketOwner` es opcional y solo es obligatorio si el bucket de Amazon S3 es propiedad de una cuenta distinta de la cuenta que ejecuta la compilación.
- Se utiliza `encryptionDisabled` para especificar si se deben cifrar los archivos exportados. Si cifra los archivos exportados, introduzca la clave administrada por el cliente. Para obtener más información, consulte [Actualizar un grupo de informes](#).
- Utilice el siguiente código JSON para especificar que el informe de pruebas no exporta archivos de prueba sin procesar:

```
{
  "name": "<report-name>",
  "type": "TEST",
  "exportConfig": {
    "exportConfigType": "NO_EXPORT"
  }
}
```

Note

El rol de servicio de CodeBuild especificado en el proyecto se utiliza para los permisos para cargar en el bucket de S3.

3. Ejecute el siguiente comando:

```
aws codebuild create-report-group --cli-input-json file://
CreateReportGroupInput.json
```

Crear un grupo de informes (AWS CloudFormation)

Para crear un informe de pruebas con la plantilla de AWS CloudFormation

Puede usar un archivo de AWS CloudFormation plantilla para crear y aprovisionar un grupo de informes. Si quiere obtener más información, consulte la [Guía del usuario de AWS CloudFormation](#).

La siguiente plantilla AWS CloudFormation YAML crea un grupo de informes que no exporta archivos de resultados de pruebas sin procesar.

```
Resources:
  CodeBuildReportGroup:
    Type: AWS::CodeBuild::ReportGroup
    Properties:
      Name: my-report-group-name
      Type: TEST
      ExportConfig:
        ExportConfigType: NO_EXPORT
```

La siguiente plantilla AWS CloudFormation YAML crea un grupo de informes que exporta los archivos de resultados de pruebas sin procesar a un bucket de Amazon S3.

```
Resources:
  CodeBuildReportGroup:
    Type: AWS::CodeBuild::ReportGroup
    Properties:
      Name: my-report-group-name
      Type: TEST
      ExportConfig:
        ExportConfigType: S3
        S3Destination:
          Bucket: my-s3-bucket-name
          Path: path-to-folder-for-exported-files
          Packaging: ZIP
          EncryptionKey: my-KMS-encryption-key
          EncryptionDisabled: false
```

Note

El rol CodeBuild de servicio especificado en el proyecto se usa para obtener permisos de carga en el bucket de S3.

Actualizar un grupo de informes

Cuando actualice un grupo de informes, puede especificar información acerca de si se deben exportar los datos de resultados de prueba sin procesar a archivos en un bucket de Amazon S3. Si decide exportar a un bucket de S3, puede especificar lo siguiente para su grupo de informes:

- Si los archivos de resultados de prueba sin procesar están comprimidos en un archivo ZIP.
- Si los archivos de resultados de prueba sin procesar están cifrados. Puede especificar el cifrado con una de las siguientes opciones:
 - Y Clave administrada de AWS para Amazon S3.
 - Una clave administrada por el cliente creada y configurada por usted.

Para obtener más información, consulte [Cifrado de datos](#).

Si utilizas el AWS CLI para actualizar un grupo de informes, también puedes actualizar o añadir etiquetas. Para obtener más información, consulte [Etiquetado de grupos de informes en AWS CodeBuild](#).

Note

El rol CodeBuild de servicio especificado en el proyecto se usa para obtener permisos de carga en el bucket de S3.

Temas

- [Actualizar un grupo de informes \(consola\)](#)
- [Actualizar un grupo de informes \(CLI\)](#)

Actualizar un grupo de informes (consola)

Para actualizar un grupo de informes

1. Abra la AWS CodeBuild consola en <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. En el panel de navegación, elija Report groups (Grupos de informes).
3. Elija el grupo de informes que desea actualizar.
4. Elija Editar.
5. Seleccione o desactive Copia de seguridad en Amazon S3. Si ha seleccionado esta opción, especifique la configuración de exportación:
 - a. En S3 bucket name (Nombre de bucket de S3), escriba el nombre del bucket de S3.
 - b. En Path prefix (Prefijo de ruta), introduzca la ruta en el bucket de S3 donde desea cargar los resultados de prueba.

- c. Seleccione Compress test result data in a zip file (Comprimir los datos de los resultados de prueba en un archivo zip) para comprimir los archivos de datos de resultados de prueba sin procesar.
- d. Expanda Additional configuration (Configuración adicional) para mostrar las opciones de cifrado. Seleccione una de las siguientes opciones:
 - Clave AWS administrada predeterminada para usar a Clave administrada de AWS para Amazon S3. Para obtener más información, consulte [CMK administradas por el cliente](#) en la Guía del usuario de AWS Key Management Service . Esta es la opción de cifrado predeterminada.
 - Seleccione una clave personalizada para usar una clave administrada por el cliente creada y configurada por usted. Como clave de cifrado de AWS KMS , introduzca el ARN de la clave de cifrado. Su formato es el siguiente: `arn:aws:kms:<region-id>:<aws-account-id>:key/<key-id>` . Para obtener más información, consulte [Creación de claves de KMS](#) en la Guía del usuario de AWS Key Management Service .
 - Disable artifact encryption (Desactivar el cifrado de artefactos) para desactivar el cifrado. Puede elegir esta opción si desea compartir los resultados de sus pruebas o publicarlos en un sitio web estático. (Un sitio web dinámico puede ejecutar código para descifrar los resultados de prueba).

Actualizar un grupo de informes (CLI)

Para actualizar un grupo de informes

1. Cree un archivo denominado `UpdateReportGroupInput.json`.
2. Copie lo siguiente en `UpdateReportGroupInput.json`:

```
{
  "arn": "",
  "exportConfig": {
    "exportConfigType": "S3",
    "s3Destination": {
      "bucket": "bucket-name",
      "path": "path",
      "packaging": "NONE | ZIP",
      "encryptionDisabled": "false",
      "encryptionKey": "your-key"
    }
  }
}
```

```
    },  
    "tags": [  
      {  
        "key": "tag-key",  
        "value": "tag-value"  
      }  
    ]  
  }  
}
```

3. Escriba el ARN de su grupo de informes en la línea de `arn` (por ejemplo, `"arn": "arn:aws:codebuild:region:123456789012:report-group/report-group-1"`).
4. Aplique a `UpdateReportGroupInput.json` las actualizaciones que desee implementar al grupo de informes.
 - Si desea actualizar el grupo de informes para exportar archivos de resultados de prueba sin procesar a un bucket de S3, actualice la sección `exportConfig`. Reemplace `bucket-name` por el nombre del bucket de S3 y `path` por la ruta de acceso del bucket de S3 a la que desea exportar los archivos. Si desea comprimir los archivos exportados, para `packaging`, especifique `ZIP`. De lo contrario, especifique `NONE`. Se utiliza `encryptionDisabled` para especificar si se deben cifrar los archivos exportados. Si cifra los archivos exportados, introduzca la clave administrada por el cliente.
 - Si desea actualizar su grupo de informes para que no exporte archivos de resultados de prueba sin procesar a un bucket de S3, actualice la sección `exportConfig` con el siguiente JSON:

```
{  
  "exportConfig": {  
    "exportConfigType": "NO_EXPORT"  
  }  
}
```

- Si desea actualizar las etiquetas del grupo de informes, actualice la sección `tags`. Puede cambiar, agregar o quitar etiquetas. Si desea quitar todas las etiquetas, actualícelo con el siguiente JSON:

```
"tags": []
```

5. Ejecute el siguiente comando:

```
aws codebuild update-report-group \  
--cli-input-json file://UpdateReportGroupInput.json
```

Especificar archivos de prueba

Los archivos de resultados de prueba y su ubicación para cada grupo de informes se especifican en la sección `reports` del archivo `buildspec` del proyecto de compilación. Para obtener más información, consulte [Reports syntax in the buildspec file](#).

A continuación se muestra una sección `reports` de ejemplo que especifica dos grupos de informes para un proyecto de compilación. Uno se especifica con su ARN y el otro con un nombre. La sección `files` especifica los archivos que contienen los resultados del caso de prueba. La sección `base-directory` opcional especifica el directorio donde se encuentran los archivos de casos de prueba. La sección `discard-paths` opcional especifica si se descartan las rutas de los archivos de resultados de prueba cargados en un bucket de Amazon S3.

```
reports:  
  arn:aws:codebuild:your-region:your-aws-account-id:report-group/report-group-name-1:  
  #surefire junit reports  
  files:  
    - '**/*'  
  base-directory: 'surefire/target/surefire-reports'  
  discard-paths: false  
  
sampleReportGroup: #Cucumber reports from json plugin  
  files:  
    - 'cucumber-json/target/cucumber-json-report.json'  
  file-format: CUCUMBERJSON #Type of the report, defaults to JUNITXML
```

Especificar comandos de prueba

Los comandos que ejecutan los casos de prueba se especifican en la sección `commands` del archivo `buildspec`. Estos comandos ejecutan los casos de prueba especificados para sus grupos de informes en la sección `reports` de su archivo `buildspec`. A continuación se muestra una sección `commands` de ejemplo que incluye comandos para ejecutar las pruebas en archivos de prueba:

```
commands:  
  - echo Running tests for surefire junit
```

```
- mvn test -f surefire/pom.xml -fn
- echo
- echo Running tests for cucumber with json plugin
- mvn test -Dcucumber.options="--plugin json:target/cucumber-json-report.json" -f
cucumber-json/pom.xml -fn
```

Para obtener más información, consulte [Sintaxis de buildspec](#).

Nomenclatura de grupos de informes

Cuando usa la consola AWS CLI o la AWS CodeBuild consola para crear un grupo de informes, especifica un nombre para el grupo de informes. Si utiliza el archivo de especificación de compilación para crear un nuevo grupo de informes, se le asigna un nombre con el formato *project-name-report-group-name-specified-in-buildspec*. Todos los informes creados mediante la ejecución de compilaciones de ese proyecto de compilación pertenecen al nuevo grupo de informes que tiene el nuevo nombre.

Si no desea CodeBuild crear un nuevo grupo de informes, especifique el ARN del grupo de informes en el archivo buildspec de un proyecto de compilación. Puede especificar el ARN de un grupo de informes en varios proyectos de compilación. Después de que se ejecute cada proyecto de compilación, el grupo de informes contiene informes de pruebas creados por cada proyecto de compilación.

Por ejemplo, si crea un grupo de informes con el nombre `my-report-group` y, a continuación, utiliza su nombre en dos proyectos de compilación diferentes denominados `my-project-1` y `my-project-2`, y crea una compilación de ambos proyectos, se crean dos nuevos grupos de informes. El resultado son tres grupos de informes con los siguientes nombres:

- `my-report-group`: no tiene informes de pruebas.
- `my-project-1-my-report-group`: contiene informes con resultados de pruebas ejecutadas por el proyecto de compilación denominado `my-project-1`.
- `my-project-2-my-report-group`: contiene informes con resultados de pruebas ejecutadas por el proyecto de compilación denominado `my-project-2`.

Si utiliza el ARN del grupo de informes denominado `my-report-group` en ambos proyectos y, a continuación, ejecuta compilaciones de cada proyecto, seguirá teniendo un grupo de informes (`my-report-group`). Ese grupo de informes contiene informes de pruebas con resultados de pruebas ejecutadas por ambos proyectos de compilación.

Si elige un nombre de grupo de informes que no pertenece a un grupo de informes de su cuenta de AWS y, a continuación, utiliza ese nombre para un grupo de informes en un archivo buildspec y ejecuta una compilación de su proyecto de compilación, se creará un nuevo grupo de informes. El formato del nombre del nuevo grupo de informes es *project-name-new-group-name*. Por ejemplo, si no hay un grupo de informes en su AWS cuenta con ese nombre `new-report-group` y lo especifica en un proyecto de compilación denominado `test-project`, una ejecución de compilación crea un nuevo grupo de informes con ese nombre. `test-project-new-report-group`

Etiquetado de grupos de informes en AWS CodeBuild

Una etiqueta es un atributo personalizado que usted o AWS asignan a un recurso de AWS. Cada etiqueta de AWS tiene dos partes:

- Una clave de etiqueta (por ejemplo, `CostCenter`, `Environment`, `Project` o `Secret`). Las claves de etiqueta distinguen entre mayúsculas y minúsculas.
- Un campo opcional que se denomina valor de etiqueta (por ejemplo, `111122223333` o `Production` o el nombre de un equipo). Omitir el valor de etiqueta es lo mismo que utilizar una cadena vacía. Al igual que las claves de etiqueta, los valores de etiqueta distinguen entre mayúsculas y minúsculas.

En conjunto, se conocen como pares clave-valor. Para conocer los límites del número de etiquetas que puede tener en un grupo de informes y las restricciones sobre las claves y valores de las etiquetas, consulte [Etiquetas](#).

Las etiquetas le ayudan a identificar y organizar los recursos de AWS. Muchos servicios de AWS admiten el etiquetado, por lo que puede asignar la misma etiqueta a los recursos de diferentes servicios para indicar que los recursos están relacionados. Por ejemplo, puede asignar la misma etiqueta a un grupo de informes de CodeBuild asignados a un bucket de Amazon S3. Para obtener más información sobre el uso de etiquetas, consulte el documento técnico [Prácticas recomendadas de etiquetado](#).

En CodeBuild, los recursos principales son el grupo de informes y el proyecto. Puede utilizar la consola de CodeBuild, la AWS CLI, las API de CodeBuild o los SDK de AWS para agregar, administrar y quitar etiquetas de un grupo de informes. Además de utilizar etiquetas para identificar, organizar y realizar el seguimiento del grupo de informes, puede utilizarlas en las políticas de IAM para ayudar a controlar quién puede consultar el grupo de informes o interactuar con él. Para ver

ejemplos de políticas de acceso basadas en etiquetas, consulte [Uso de etiquetas para controlar el acceso a los recursos de AWS CodeBuild](#).

Temas

- [Agregar una etiqueta a un grupo de informes](#)
- [Consultar etiquetas de un grupo de informes](#)
- [Editar etiquetas de un grupo de informes](#)
- [Quitar una etiqueta de un grupo de informes](#)

Agregar una etiqueta a un grupo de informes

Agregar etiquetas a un grupo de informes puede ayudarle a identificar y organizar sus recursos de AWS y a administrar el acceso a ellos. En primer lugar, agregue una o varias etiquetas (pares de clave-valor) a un grupo de informes. Tenga en cuenta que hay límites en el número de etiquetas que puede tener en un grupo de informes. Existen restricciones sobre los caracteres que se pueden utilizar en los campos de clave y valor. Para obtener más información, consulte [Etiquetas](#). Cuando tenga las etiquetas, puede crear políticas de IAM para administrar el acceso al grupo de informes en función de estas etiquetas. Puede usar la consola de CodeBuild o la AWS CLI para agregar etiquetas a un grupo de informes.

Important

Agregar etiquetas a un grupo de informes puede afectar al acceso a ese grupo de informes. Antes de agregar una etiqueta a un grupo de informes, asegúrese de revisar las políticas de IAM que es posible que utilicen etiquetas para controlar el acceso a recursos como los grupos de informes. Para ver ejemplos de políticas de acceso basadas en etiquetas, consulte [Uso de etiquetas para controlar el acceso a los recursos de AWS CodeBuild](#).

Para obtener más información sobre cómo agregar etiquetas a un grupo de informes al crearlo, consulte [Crear un grupo de informes \(consola\)](#).

Temas

- [Agregar una etiqueta a un grupo de informes \(consola\)](#)
- [Agregar una etiqueta a un grupo de informes \(AWS CLI\)](#)

Agregar una etiqueta a un grupo de informes (consola)

Puede usar la consola de CodeBuild para agregar una o varias etiquetas a un grupo de informes de CodeBuild.

1. Abra la consola de CodeBuild en <https://console.aws.amazon.com/codebuild/>.
2. En Grupos de informes, elija el nombre del grupo de informes al que desea agregar etiquetas.
3. En el panel de navegación, seleccione Settings (Configuración).
4. Si las etiquetas no se han agregado al grupo de informes, elija Agregar etiqueta. También puede elegir Editar y, a continuación, elegir Agregar etiqueta.
5. En Key (Clave), escriba un nombre para la etiqueta. Puede añadir un valor opcional para la etiqueta en Value (Valor).
6. (Opcional) Para añadir otra etiqueta, vuelva a elegir Add tag (Añadir etiqueta).
7. Cuando haya terminado de añadir etiquetas, seleccione Submit (Enviar).

Agregar una etiqueta a un grupo de informes (AWS CLI)

Para agregar una etiqueta a un grupo de informes al crearlo, consulte [Crear un grupo de informes \(CLI\)](#). En `CreateReportGroup.json`, agregue sus etiquetas.

Para agregar etiquetas a un grupo de informes existente, consulte [Actualizar un grupo de informes \(CLI\)](#) y agregue las etiquetas en `UpdateReportGroupInput.json`.

En estos pasos, se presupone que ya ha instalado una versión reciente de la AWS CLI o que la ha actualizado a la versión actual. Para obtener más información, consulte [Instalación de AWS Command Line Interface](#).

Consultar etiquetas de un grupo de informes

Las etiquetas pueden ayudarle a identificar y organizar sus recursos de AWS y administrar el acceso a ellos. Para obtener más información sobre el uso de etiquetas, consulte el documento técnico [Prácticas recomendadas de etiquetado](#). Para ver ejemplos de políticas de acceso basadas en etiquetas, consulte [Deny or allow actions on report groups based on resource tags](#).

Consultar etiquetas de un grupo de informes (consola)

Puede usar la consola de CodeBuild para consultar las etiquetas asociadas con un grupo de informes de CodeBuild.

1. Abra la consola de CodeBuild en <https://console.aws.amazon.com/codebuild/>.
2. En Grupos de informes, elija el nombre del grupo de informes en el que desea consultar las etiquetas.
3. En el panel de navegación, seleccione Settings (Configuración).

Consultar etiquetas de un grupo de informes (AWS CLI)

Siga estos pasos para utilizar la AWS CLI para consultar las etiquetas de AWS de un grupo de informes. Si no se han agregado etiquetas, la lista de etiquetas devueltas está vacía.

1. Utilice la consola o la AWS CLI para localizar el ARN del grupo de informes. Anótelos.

AWS CLI

Ejecute el siguiente comando.

```
aws list-report-groups
```

Este comando devuelve información con formato JSON similar a la siguiente:

```
{
  "reportGroups": [
    "arn:aws:codebuild:region:123456789012:report-group/report-group-1",
    "arn:aws:codebuild:region:123456789012:report-group/report-group-2",
    "arn:aws:codebuild:region:123456789012:report-group/report-group-3"
  ]
}
```

El ARN del grupo de informes termina con su nombre, que puede utilizar para identificar el ARN del grupo de informes.

Console

1. Abra la consola de CodeBuild en <https://console.aws.amazon.com/codebuild/>.
 2. En Grupos de informes, elija el nombre del grupo de informes con las etiquetas que desea consultar.
 3. En Configuración localice el ARN del grupo de informes.
2. Ejecute el siguiente comando. Utilice el ARN del que tomó nota para el parámetro `--report-group-arns`.

```
aws codebuild batch-get-report-groups --report-group-arns
arn:aws:codebuild:region:123456789012:report-group/report-group-name
```

Si tiene éxito, este comando devuelve información con formato JSON que contiene una sección `tags` similar a la siguiente:

```
{
  ...
  "tags": {
    "Status": "Secret",
    "Project": "TestBuild"
  }
  ...
}
```

Editar etiquetas de un grupo de informes

Puede cambiar el valor de una etiqueta asociada con un grupo de informes. También puede cambiar el nombre de la clave, lo que equivale a eliminar la etiqueta actual y añadir otra distinta con el nuevo nombre y el mismo valor que la otra clave. Tenga en cuenta que hay restricciones en los caracteres que puede utilizar en los campos de clave y valor. Para obtener más información, consulte [Etiquetas](#).

Important

La edición de etiquetas de un grupo de informes puede afectar al acceso a ese grupo de informes. Antes de editar el nombre (clave) o valor de una etiqueta de un grupo de informes, asegúrese de revisar cualquier política de IAM que es posible que utilice la clave o el valor de una etiqueta para controlar el acceso a recursos como los grupos de informes. Para ver ejemplos de políticas de acceso basadas en etiquetas, consulte [Deny or allow actions on report groups based on resource tags](#).

Editar una etiqueta de un grupo de informes (consola)

Puede usar la consola de CodeBuild para editar las etiquetas asociadas con un grupo de informes de CodeBuild.

1. Abra la consola de CodeBuild en <https://console.aws.amazon.com/codebuild/>.

2. En Grupos de informes, elija el nombre del grupo de informes en el que desea editar las etiquetas.
3. En el panel de navegación, seleccione Settings (Configuración).
4. Elija Edit (Editar).
5. Haga una de las siguientes acciones:
 - Para cambiar la etiqueta, escriba un nuevo nombre en Key (Clave). Cambiar el nombre de la etiqueta equivale a quitar una etiqueta y añadir otra nueva con el nuevo nombre de clave.
 - Para cambiar el valor de una etiqueta, escriba un nuevo valor. Si desea cambiar el valor por nada, elimine el valor actual y deje el campo en blanco.
6. Cuando haya terminado de editar las etiquetas, seleccione Submit (Enviar).

Editar etiquetas de un grupo de informes (AWS CLI)

Para agregar, cambiar o eliminar etiquetas de un grupo de informes, consulte [Actualizar un grupo de informes \(CLI\)](#). Actualice las etiquetas en `UpdateReportGroupInput.json`.

Quitar una etiqueta de un grupo de informes

Puede quitar una o varias etiquetas asociadas con un grupo de informes. La eliminación de una etiqueta no la elimina de otros recursos de AWS que están asociados con esa etiqueta.

Important

Quitar etiquetas de un grupo de informes puede afectar al acceso a ese grupo de informes. Antes de quitar una etiqueta de un grupo de informes, asegúrese de revisar cualquier política de política de IAM que es posible que utilice la clave o el valor de una etiqueta para controlar el acceso a recursos como los grupos de informes. Para ver ejemplos de políticas de acceso basadas en etiquetas, consulte [Uso de etiquetas para controlar el acceso a los recursos de AWS CodeBuild](#).

Quitar una etiqueta de un grupo de informes (consola)

Puede usar la consola de CodeBuild para eliminar la asociación entre una etiqueta y un grupo de informes de CodeBuild.

1. Abra la consola de CodeBuild en <https://console.aws.amazon.com/codebuild/>.

2. En Grupos de informes, elija el nombre del grupo de informes en el que desea quitar etiquetas.
3. En el panel de navegación, seleccione Settings (Configuración).
4. Elija Edit (Editar).
5. Busque la etiqueta que desea eliminar y, a continuación, seleccione Remove tag (Eliminar etiqueta).
6. Cuando haya terminado de eliminar las etiquetas, seleccione Submit (Enviar).

Quitar una etiqueta de un grupo de informes (AWS CLI)

Siga estos pasos para utilizar la AWS CLI para quitar una etiqueta de un grupo de informes de CodeBuild. Al quitar una etiqueta no la elimina, sino que simplemente elimina la asociación entre la etiqueta y el grupo de informes.

Note

Si elimina un grupo de informes de CodeBuild, se eliminan todas las asociaciones de etiquetas del grupo de informes eliminado. No es necesario quitar las etiquetas antes de eliminar un grupo de informes.

Para eliminar una o varias etiquetas de un grupo de informes, consulte [Editar etiquetas de un grupo de informes \(AWS CLI\)](#). Actualice la sección tags de los datos con formato JSON con una lista actualizada de etiquetas que no contenga las que desea eliminar. Si desea eliminar todas las etiquetas, actualice la sección tags para:

```
"tags: []"
```

Usar grupos de informes compartidos

El uso compartido de grupos de informes permite que varias cuentas o usuarios de AWS vean un grupo de informes, sus informes no caducados y los resultados de las pruebas de sus informes. En este modelo, la cuenta que posee el grupo de informes (propietario) comparte un grupo de informes con otras cuentas (consumidores). Un consumidor no puede editar un grupo de informes. Un informe caduca 30 días después de su creación.

Contenido

- [Requisitos previos para compartir grupos de informes](#)
- [Requisitos previos para acceder a grupos de informes compartidos con usted](#)
- [Servicios relacionados](#)
- [Uso compartido de un grupo de informes](#)
- [Dejar de compartir un grupo de informes compartidos](#)
- [Identificación de un grupo de informes compartido](#)
- [Permisos de un grupo de informes compartido](#)

Requisitos previos para compartir grupos de informes

Para compartir un grupo de informes, su cuenta de AWS debe ser la propietaria. No puede compartir un grupo de informes que se haya compartido con usted.

Requisitos previos para acceder a grupos de informes compartidos con usted

Para tener acceso a un grupo de informes compartido, el rol de IAM de un consumidor requiere el permiso `BatchGetReportGroups`. Puede adjuntar la siguiente política a su rol de IAM:

```
{
  "Effect": "Allow",
  "Resource": [
    "*"
  ],
  "Action": [
    "codebuild:BatchGetReportGroups"
  ]
}
```

Para obtener más información, consulte [Uso de políticas basadas en la identidad para AWS CodeBuild](#).

Servicios relacionados

El uso compartido de grupos de informes se integra con AWS Resource Access Manager (AWS RAM), un servicio que le permite compartir sus recursos de AWS con cualquier cuenta de AWS o a través de AWS Organizations. Con AWS RAM, comparte los recursos que posee mediante la creación de un recurso compartido que especifica los recursos y los consumidores con los que los

comparte. Los consumidores pueden ser cuentas de AWS individuales, unidades organizativas en AWS Organizations o toda una organización de AWS Organizations.

Para obtener más información, consulte la Guía del usuario de [AWS RAM](#).

Uso compartido de un grupo de informes

Cuando comparte un grupo de informes, al consumidor se le concede acceso de solo lectura al grupo de informes y a sus informes. El consumidor puede usar la AWS CLI para ver el grupo de informes, sus informes y los resultados del caso de prueba para cada informe. El consumidor no puede:

- Ver un grupo de informes compartido o sus informes en la consola de CodeBuild.
- Editar un grupo de informes compartido.
- Utilice el ARN del grupo de informes compartido en un proyecto para ejecutar un informe. Se produce un error en una compilación de proyecto que especifica un grupo de informes compartido.

Puede utilizar la consola de CodeBuild para añadir un grupo de informes a un recurso compartido existente. Si desea añadir el grupo de informes a un nuevo recurso compartido, primero debe crearlo en la [consola de AWS RAM](#).

Para compartir un grupo de informes con unidades organizativas o con toda la organización, debe permitir el uso compartido con AWS Organizations. Para obtener más información, consulte [Habilitar el uso compartido con AWS Organizations](#) en la Guía del usuario de AWS RAM.

Puede usar la consola de CodeBuild, la consola de AWS RAM o AWS CLI para compartir grupos de informes de su propiedad.

Para compartir un grupo de informes de su propiedad (consola de CodeBuild)

1. Abra la consola de AWS CodeBuild en <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. En el panel de navegación, elija Report groups (Grupos de informes).
3. Elija el proyecto que desea compartir y, a continuación, elija Share (Compartir). Para obtener más información, consulte [Crear un recurso compartido](#) en la Guía del usuario de AWS RAM.

Para compartir grupos de informes que posee (consola de AWS RAM)

Consulte [Crear un recurso compartido](#) en la Guía del usuario de AWS RAM.

Para compartir grupos de informes que posee (comando AWS RAM)

Utilice el comando [create-resource-share](#).

Para compartir un grupo de informes que posee (comando de CodeBuild)

Utilice el comando [put-resource-policy](#):

1. Cree un archivo llamado `policy.json` y copie lo siguiente en él.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Principal": {
      "AWS": "consumer-aws-account-id-or-user"
    },
    "Action": [
      "codebuild:BatchGetReportGroups",
      "codebuild:BatchGetReports",
      "codebuild:ListReportsForReportGroup",
      "codebuild:DescribeTestCases"],
    "Resource": "arn-of-report-group-to-share"
  ]
}
```

2. Actualice `policy.json` con el ARN del grupo de informes y los identificadores para compartirlo. El ejemplo siguiente concede acceso de solo lectura al grupo de informes con el ARN `arn:aws:codebuild:us-west-2:123456789012:report-group/my-report-group` a Alice y al usuario raíz para la cuenta de AWS identificada por 123456789012.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Principal": {
      "AWS": [
        "arn:aws:iam::123456789012:user/Alice",
        "123456789012"
      ]
    },
  ],
}
```

```
"Action":[
  "codebuild:BatchGetReportGroups",
  "codebuild:BatchGetReports",
  "codebuild:ListReportsForReportGroup",
  "codebuild:DescribeTestCases"],
"Resource":"arn:aws:codebuild:us-west-2:123456789012:report-group/my-report-
group"
}]
}
```

3. Ejecute el siguiente comando.

```
aws codebuild put-resource-policy --resource-arn report-group-arn --policy file://
policy.json
```

Dejar de compartir un grupo de informes compartidos

Solo el propietario puede obtener acceso a un grupo de informes no compartido, incluidos sus informes y los resultados de los casos de prueba. Si deja de compartir un grupo de informes, cualquier cuenta de AWS o usuario con el que lo haya compartido anteriormente no podrá obtener acceso al grupo de informes, a sus informes ni a los resultados de los casos de prueba de los informes.

Para dejar de compartir un grupo de informes compartido que posee, debe quitarlo del recurso compartido. Puede usar la consola de AWS RAM o AWS CLI para realizar esta operación.

Para dejar de compartir un grupo de informes compartido que posee (consola de AWS RAM)

Consulte [Actualización de un recurso compartido](#) en la Guía del usuario de AWS RAM.

Para dejar de compartir un grupo de informes compartido que posee (comando AWS RAM)

Utilice el comando [disassociate-resource-share](#).

Para dejar de compartir el grupo de informes que posee (comando de CodeBuild)

Ejecute el comando [delete-resource-policy](#) y especifique el ARN del grupo de informes que desea dejar de compartir:

```
aws codebuild delete-resource-policy --resource-arn report-group-arn
```

Identificación de un grupo de informes compartido

Los propietarios y los consumidores pueden utilizar la AWS CLI para identificar grupos de informes compartidos.

Para identificar y obtener información acerca de un grupo de informes compartido y sus informes, utilice los siguientes comandos:

- Para ver los ARN de los grupos de informes compartidos con usted, ejecute [list-shared-report-groups](#):

```
aws codebuild list-shared-report-groups
```

- Para ver los ARN de los informes de un grupo de informes, ejecute [list-reports-for-report-group](#) con el ARN del grupo de informes:

```
aws codebuild list-reports-for-report-group --report-group-arn report-group-arn
```

- Para ver la información sobre los casos de prueba en un informe, ejecute [describe-test-cases](#) con el ARN de informe:

```
aws codebuild describe-test-cases --report-arn report-arn
```

El resultado es similar al siguiente:

```
{
  "testCases": [
    {
      "status": "FAILED",
      "name": "Test case 1",
      "expired": 1575916770.0,
      "reportArn": "report-arn",
      "prefix": "Cucumber tests for agent",
      "message": "A test message",
      "durationInNanoSeconds": 1540540,
      "testRawDataPath": "path-to-output-report-files"
    },
    {
      "status": "SUCCEEDED",
      "name": "Test case 2",
      "expired": 1575916770.0,

```

```
        "reportArn": "report-arn",
        "prefix": "Cucumber tests for agent",
        "message": "A test message",
        "durationInNanoSeconds": 1540540,
        "testRawDataPath": "path-to-output-report-files"
    }
]
}
```

Permisos de un grupo de informes compartido

Permisos de los propietarios

Un propietario de un grupo de informes puede editarlo y especificarlo en un proyecto para ejecutar informes.

Permisos de los consumidores

Un consumidor de grupo de informes puede ver un grupo de informes, sus informes y los resultados de los casos de prueba de sus informes. Un consumidor no puede editar un grupo de informes ni sus informes y no puede usarlo para crear informes.

Uso de informes

Un informe contiene los resultados de los casos de prueba especificados para un grupo de informes. Se crea un informe de pruebas durante la ejecución de un proyecto de compilación. Se especifican un grupo de informes, los archivos de casos de prueba y los comandos para ejecutar los casos de prueba en su archivo buildspec. Cada vez que se ejecutan los casos de prueba, se crea un nuevo informe de pruebas en el grupo de informes.

Un informe de pruebas caduca 30 días después de su creación. No puede ver un informe de pruebas caducado, pero puede exportar los resultados de pruebas a archivos de resultados de pruebas sin procesar a un bucket de S3. Los archivos de pruebas sin procesar exportados no caducan. Para obtener más información, consulte [Actualizar un grupo de informes](#).

El estado de un informe de pruebas puede ser uno de los siguientes:

- **GENERATING**: la ejecución de los casos de prueba todavía está en curso.

- **DELETING:** se está eliminando el informe de pruebas. Cuando se elimina un informe de pruebas, también se eliminan sus casos de prueba. Los archivos de datos de resultados de pruebas sin procesar exportados a un bucket de S3 no se eliminan.
- **INCOMPLETE:** el informe de pruebas no se ha completado. Este estado se puede devolver por uno de los siguientes motivos:
 - Un problema con la configuración del grupo de informes que especifica los casos de prueba de este informe. Por ejemplo, la ruta de los casos de prueba en el grupo de informes del archivo buildspec puede ser incorrecta.
 - El usuario de IAM que ha ejecutado la compilación no tiene permisos para ejecutar pruebas. Para obtener más información, consulte [Uso de permisos de informes de pruebas](#).
 - La compilación no se ha completado debido a un error que no está relacionado con las pruebas.
- **SUCCEEDED:** todos los casos de prueba se realizaron correctamente.
- **FAILED:** algunos de los casos de prueba no se realizaron correctamente.

Cada caso de prueba devuelve un estado. El estado de un caso de prueba puede ser uno de los siguientes:

- **SUCCEEDED:** se ha superado el caso de prueba.
- **FAILED:** se ha producido un error en el caso de prueba.
- **ERROR:** el caso de prueba ha producido un error inesperado.
- **SKIPPED:** el caso de prueba no se ha ejecutado.
- **UNKNOWN:** el caso de prueba ha devuelto un estado distinto de SUCCEEDED, FAILED, ERROR o SKIPPED.

Un informe de pruebas puede tener un máximo de 500 resultados de casos de prueba. Si se ejecutan más de 500 casos de prueba, CodeBuild prioriza las pruebas con el estado FAILED y trunca los resultados de los casos de prueba.

Uso de permisos de informes de pruebas

En este tema se describe información importante acerca de los permisos relacionados con los informes de pruebas.

Temas

- [Crear un rol para los informes de pruebas](#)
- [Permisos para las operaciones de informes de pruebas](#)
- [Ejemplos de permisos de informes de pruebas](#)

Crear un rol para los informes de pruebas

Para ejecutar un informe de pruebas y actualizar un proyecto para incluir informes de pruebas, el rol de IAM requiere los siguientes permisos. Estos permisos se incluyen en las políticas gestionadas predefinidas. AWS Si desea añadir un informe de pruebas a un proyecto de compilación existente, debe añadir estos permisos.

- `CreateReportGroup`
- `CreateReport`
- `UpdateReport`
- `BatchPutTestCases`

Para ejecutar un informe de cobertura de código, su rol de IAM también debe incluir el permiso `BatchPutCodeCoverages`.

Note

`BatchPutTestCases`, `CreateReport`, `UpdateReport` y `BatchPutCodeCoverages` no son permisos públicos. No puede llamar al AWS CLI comando o método SDK correspondiente para obtener estos permisos.

Para asegurarse de que tiene estos permisos, puede asociar la siguiente política a su rol de IAM:

```
{
  "Effect": "Allow",
  "Resource": [
    "*"
  ],
  "Action": [
    "codebuild:CreateReportGroup",
    "codebuild:CreateReport",
    "codebuild:UpdateReport",
  ]
}
```

```
    "codebuild:BatchPutTestCases",
    "codebuild:BatchPutCodeCoverages"
  ]
}
```

Le recomendamos que restrinja esta política únicamente a los grupos de informes que debe utilizar. Lo siguiente restringe los permisos solo a los grupos de informes con los dos ARN en la política:

```
{
  "Effect": "Allow",
  "Resource": [
    "arn:aws:codebuild:your-region:your-aws-account-id:report-group/report-group-name-1",
    "arn:aws:codebuild:your-region:your-aws-account-id:report-group/report-group-name-2"
  ],
  "Action": [
    "codebuild:CreateReportGroup",
    "codebuild:CreateReport",
    "codebuild:UpdateReport",
    "codebuild:BatchPutTestCases",
    "codebuild:BatchPutCodeCoverages"
  ]
}
```

Lo siguiente restringe los permisos solo a los grupos de informes creados por la ejecución de compilaciones de un proyecto llamado `my-project`:

```
{
  "Effect": "Allow",
  "Resource": [
    "arn:aws:codebuild:your-region:your-aws-account-id:report-group/my-project-*"
  ],
  "Action": [
    "codebuild:CreateReportGroup",
    "codebuild:CreateReport",
    "codebuild:UpdateReport",
    "codebuild:BatchPutTestCases",
    "codebuild:BatchPutCodeCoverages"
  ]
}
```

Note

El rol CodeBuild de servicio especificado en el proyecto se usa para obtener permisos de carga en el bucket de S3.

Permisos para las operaciones de informes de pruebas

Puede especificar los permisos para las siguientes operaciones de la CodeBuild API de informes de pruebas:

- BatchGetReportGroups
- BatchGetReports
- CreateReportGroup
- DeleteReportGroup
- DeleteReport
- DescribeTestCases
- ListReportGroups
- ListReports
- ListReportsForReportGroup
- UpdateReportGroup

Para obtener más información, consulte [AWS CodeBuild referencia de permisos](#).

Ejemplos de permisos de informes de pruebas

Para obtener información acerca de las directivas de ejemplo relacionadas con los informes de pruebas, consulte lo siguiente:

- [Permitir que un usuario pueda cambiar un grupo de informes](#)
- [Permitir que un usuario pueda crear un grupo de informes](#)
- [Permitir que un usuario pueda eliminar un informe](#)
- [Permitir que un usuario pueda eliminar un grupo de informes](#)
- [Permitir que un usuario obtenga información sobre los grupos de informes](#)
- [Permitir que un usuario obtenga información sobre los informes](#)

- [Permitir que a un usuario pueda obtener una lista de grupos de informes](#)
- [Permitir que un usuario pueda obtener una lista de informes](#)
- [Permitir que un usuario pueda obtener una lista con los informes de un grupo de informes](#)
- [Permitir que un usuario pueda obtener una lista con los casos de prueba de un informe](#)

Consultar los informes de pruebas

Puede ver detalles sobre un informe de pruebas, como información sobre sus casos de prueba, los números de realizaciones correctas e incorrectas, y el tiempo que tardó en ejecutarse. Puedes ver los informes de las pruebas agrupados por compilación, ejecución, grupo de informes o tu AWS cuenta. Elija un informe de pruebas en la consola para ver los detalles y los resultados de sus casos de prueba.

Puede ver los informes de pruebas que no están caducados. Los informes de pruebas caducan 30 días después de su creación. No puede ver un informe caducado en CodeBuild.

Temas

- [Consultar los informes de pruebas de una compilación](#)
- [Consultar los informes de pruebas de un grupo de informes](#)
- [Consultar los informes de pruebas de una cuenta de AWS](#)

Consultar los informes de pruebas de una compilación

Para ver los informes de pruebas de una compilación

1. Abra la AWS CodeBuild consola en <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Localice la compilación que desea ver. Si conoce el proyecto que ejecutó la compilación que creó el informe de pruebas:
 1. En el panel de navegación, elija Build projects (Proyectos de compilación) y, a continuación, elija el proyecto con la compilación que ejecutó el informe de pruebas que desea ver.
 2. Elija Build history (Historial de compilación) y, a continuación, elija la compilación ejecutada que creó los informes que desea ver.

También puede localizar la compilación en el historial de compilación de su cuenta de AWS :

1. En el panel de navegación, elija Build history (Historial de compilación) y, a continuación, elija la compilación que creó los informes de pruebas que desea ver.
3. En la página de compilación, elija Reports (Informes) y, a continuación, elija un informe de pruebas para ver sus detalles.

Consultar los informes de pruebas de un grupo de informes

Para ver informes de pruebas de un grupo de informes

1. Abra la AWS CodeBuild consola en <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. En el panel de navegación, elija Report groups (Grupos de informes).
3. Elija el grupo de informes que contiene los informes de pruebas que desea ver.
4. Elija un informe de pruebas para ver sus detalles.

Consultar los informes de pruebas de una cuenta de AWS

Para ver los informes de las pruebas en su AWS cuenta

1. Abra la AWS CodeBuild consola en <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. En el panel de navegación, elija Report history (Historial de informes).
3. Elija un informe de pruebas para ver sus detalles.

Generación de informes de pruebas con marcos de trabajo de pruebas

En los temas de esta sección se muestra cómo configurar los informes de pruebas AWS CodeBuild para varios marcos de pruebas.

Temas

- [Configurar generación de informes de pruebas con Jasmine](#)
- [Configurar generación de informes de pruebas con Jest](#)
- [Configurar generación de informes de pruebas con pytest](#)
- [Configurar generación de informes de pruebas con RSpec](#)

Configurar generación de informes de pruebas con Jasmine

El siguiente procedimiento muestra cómo configurar generación de informes de prueba en AWS CodeBuild con el [marco de pruebas JasmineBDD](#).

El procedimiento requiere los siguientes requisitos previos:

- Debe tener un proyecto de CodeBuild existente.
- Su proyecto es un proyecto Node.js que está configurado para usar el marco de pruebas de Jasmine.

Agregue el paquete [jasmine-reporters](#) a la sección `devDependencies` del archivo `package.json` de su proyecto. Este paquete tiene una colección de clases de generador de informes de JavaScript que se pueden usar con Jasmine.

```
npm install --save-dev jasmine-reporters
```

Si todavía no está presente, agregue el script `test` al archivo `package.json` de su proyecto. El script `test` asegura que se llame a Jasmine cuando se ejecuta `npm test`.

```
{
  "scripts": {
    "test": "npx jasmine"
  }
}
```

CodeBuild es compatible con los siguientes generadores de informes de pruebas de Jasmine:

JUnitXmlReporter

Se utiliza para generar informes en el formato `JUnitXml`.

NUnitXmlReporter

Se utiliza para generar informes en el formato `NunitXml`.

Un proyecto Node.js con Jasmine tendrá, de forma predeterminada, un subdirectorio `spec`, que contiene los scripts de configuración y prueba de Jasmine.

Para configurar Jasmine para generar informes en el formato JUnitXML, cree una instancia para el generador de informes JUnitXmlReporter agregando el siguiente código a las pruebas.

```
var reporters = require('jasmine-reporters');

var junitReporter = new reporters.JUnitXmlReporter({
  savePath: <test report directory>,
  filePrefix: <report filename>,
  consolidateAll: true
});

jasmine.getEnv().addReporter(junitReporter);
```

Para configurar Jasmine para generar informes en el formato NUnitXML, cree una instancia para el generador de informes NUnitXmlReporter agregando el siguiente código a las pruebas.

```
var reporters = require('jasmine-reporters');

var nunitReporter = new reporters.NUnitXmlReporter({
  savePath: <test report directory>,
  filePrefix: <report filename>,
  consolidateAll: true
});

jasmine.getEnv().addReporter(nunitReporter)
```

Los informes de pruebas se exportan al archivo especificado por *<directorio de informes de pruebas>/<nombre de archivo de pruebas>*.

En su archivo `buildspec.yml`, agregue y actualice las siguientes secciones.

```
version: 0.2

phases:
  pre_build:
    commands:
      - npm install
  build:
    commands:
      - npm build
      - npm test
```

```
reports:
  jasmine_reports:
    files:
      - <report filename>
    file-format: JUNITXML
    base-directory: <test report directory>
```

Si utiliza el formato de informe NunitXml, cambie el valor `file-format` al siguiente.

```
file-format: NUNITXML
```

Configurar generación de informes de pruebas con Jest

El siguiente procedimiento muestra cómo configurar generación de informes de pruebas en AWS CodeBuild con el [marco de pruebas de Jest](#).

El procedimiento requiere los siguientes requisitos previos:

- Debe tener un proyecto de CodeBuild existente.
- Su proyecto es un proyecto Node.js que está configurado para usar el marco de pruebas de Jest.

Agregue el paquete [jest-junit](#) a la sección `devDependencies` del archivo `package.json` de su proyecto. CodeBuild utiliza este paquete para generar informes en el formato `JunitXml`.

```
npm install --save-dev jest-junit
```

Si todavía no está presente, agregue el script `test` al archivo `package.json` de su proyecto. El script `test` asegura que se llama a Jest al ejecutar `npm test`.

```
{
  "scripts": {
    "test": "jest"
  }
}
```

Configure Jest para usar el generador de informes `JunitXml` agregando lo siguiente al archivo de configuración de Jest. Si su proyecto no tiene un archivo de configuración de Jest, cree

un archivo denominado `jest.config.js` en la raíz del proyecto y agregue lo siguiente. Los informes de pruebas se exportan al archivo especificado por *<directorio de informes de pruebas>/<nombre de archivo de pruebas>*.

```
module.exports = {
  reporters: [
    'default',
    [ 'jest-junit', {
      outputDirectory: <test report directory>,
      outputName: <report filename>,
    } ]
  ]
};
```

En su archivo `buildspec.yml`, agregue y actualice las siguientes secciones.

```
version: 0.2

phases:
  pre_build:
    commands:
      - npm install
  build:
    commands:
      - npm build
      - npm test

reports:
  jest_reports:
    files:
      - <report filename>
    file-format: JUNITXML
    base-directory: <test report directory>
```

Configurar generación de informes de pruebas con pytest

El siguiente procedimiento muestra cómo configurar informes de pruebas en AWS CodeBuild con el [marco de pruebas de pytest](#).

El procedimiento requiere los siguientes requisitos previos:

- Debe tener un proyecto de CodeBuild existente.

- Su proyecto es un proyecto de Python que está configurado para usar el marco de pruebas de pytest.

Agregue la siguiente entrada a la fase `build` o `post_build` del archivo `buildspec.yml`. Este código detecta automáticamente las pruebas en el directorio actual y exporta los informes de pruebas al archivo especificado por `<directorio de informes de pruebas>/<nombre de archivo de informes>`. El informe utiliza el formato JunitXml.

```
- python -m pytest --junitxml=<test report directory>/<report filename>
```

En su archivo `buildspec.yml`, agregue y actualice las siguientes secciones.

```
version: 0.2

phases:
  install:
    runtime-versions:
      python: 3.7
    commands:
      - pip3 install pytest
  build:
    commands:
      - python -m pytest --junitxml=<test report directory>/<report filename>

reports:
  pytest_reports:
    files:
      - <report filename>
    base-directory: <test report directory>
    file-format: JUNITXML
```

Configurar generación de informes de pruebas con RSpec

El siguiente procedimiento muestra cómo configurar generación de informes de pruebas en AWS CodeBuild con el [marco de pruebas de RSpec](#).

El procedimiento requiere los siguientes requisitos previos:

- Debe tener un proyecto de CodeBuild existente.
- Su proyecto es un proyecto Ruby que está configurado para usar el marco de pruebas de RSpec.

Agregue y actualice lo siguiente en su archivo `buildspec.yml`. Este código ejecuta las pruebas en el directorio `<directorio de fuente de pruebas>` y exporta los informes de pruebas al archivo especificado por `<directorio de informes de pruebas>/<nombre de archivo del informe>`. El informe utiliza el formato `JUnitXml`.

```
version: 0.2

phases:
  install:
    runtime-versions:
      ruby: 2.6
  pre_build:
    commands:
      - gem install rspec
      - gem install rspec_junit_formatter
  build:
    commands:
      - rspec <test source directory>/* --format RspecJUnitFormatter --out <test report
        <report filename>
  reports:
    rspec_reports:
      files:
        - <report filename>
      base-directory: <test report directory>
      file-format: JUNITXML
```

Informes de cobertura de código

CodeBuild le permite generar informes de cobertura de código para sus pruebas. Se proporcionan los siguientes informes de cobertura de código:

Cobertura de línea

La cobertura de línea mide el número de declaraciones cubiertas por las pruebas. Una declaración es una sola instrucción, in contar comentarios ni condicionales.

$$\text{line coverage} = (\text{total lines covered}) / (\text{total number of lines})$$

Cobertura de ramificaciones

La cobertura de ramificaciones mide cuántas ramificaciones cubren las pruebas de cada ramificación posible de una estructura de control, como una declaración `if` o `case`.

```
branch coverage = (total branches covered)/(total number of branches)
```

Se admiten los siguientes formatos de archivo de informe de cobertura de código:

- JaCoCo XML
- SimpleCov JSON¹
- Clover XML
- Cobertura XML
- INFORMACIÓN LCOV

¹ [CodeBuild acepta informes de cobertura de código JSON generados por simplecov, no por simplecov-json.](#)

Crear un informe de cobertura de código

Para crear un informe de cobertura de código, ejecuta un proyecto de compilación que esté configurado con al menos un grupo de informes de cobertura de código en su archivo buildspec. CodeBuild interpretará los resultados de la cobertura de código y proporcionará un informe de cobertura de código para la ejecución. Se genera un nuevo informe de pruebas para cada compilación posterior que utilice el mismo archivo buildspec.

Para crear un informe de pruebas

1. Cree un proyecto de compilación. Para obtener más información, consulte [Creación de un proyecto de compilación en AWS CodeBuild](#).
2. Configure el archivo de especificación de la compilación del proyecto con información de informe de pruebas:
 - a. Agregue una `reports:` sección y especifique el nombre del grupo de informes. CodeBuild crea un grupo de informes para usted con el nombre de su proyecto y el nombre que especificó en el formato `project-name-report-group-name-in-buildspec`. Si ya tiene un grupo de informes que desee utilizar, especifique su ARN. Si utiliza el nombre en lugar del ARN, CodeBuild crea un nuevo grupo de informes. Para obtener más información, consulte [Reports syntax in the buildspec file](#).
 - b. En el grupo de informes, especifique la ubicación de los archivos que contienen los resultados de la cobertura de código. Si utiliza más de un grupo de informes, especifique las ubicaciones de los archivos de resultados para cada uno de ellos. Cada vez que se ejecuta

el proyecto de compilación, se crea un nuevo informe de cobertura de código. Para obtener más información, consulte [Especificar archivos de prueba](#).

Este es un ejemplo que genera un informe de cobertura de código para un archivo de resultados JaCoCo XML ubicado en `test-results/jacoco-coverage-report.xml`.

```
reports:
  jacoco-report:
    files:
      - 'test-results/jacoco-coverage-report.xml'
    file-format: 'JACOCOXML'
```

- c. En la sección `commands` de la secuencia `build` o `post_build`, especifique los comandos que ejecutan el análisis de cobertura del código. Para obtener más información, consulte [Especificar comandos de prueba](#).
3. Ejecute una compilación del proyecto de compilación. Para obtener más información, consulte [Ejecutar una compilación en AWS CodeBuild](#).
4. Cuando la compilación se haya completado, elija la nueva ejecución de la compilación en Build history (Historial de compilación) en la página del proyecto. Seleccione Informes para ver el informe de cobertura de código. Para obtener más información, consulte [Consultar los informes de pruebas de una compilación](#).

Detección automática de informes

Con la detección automática, CodeBuild busca en todos los archivos de compilación una vez finalizada la fase de creación, busca cualquier tipo de archivo de informe compatible y crea automáticamente nuevos informes y grupos de informes de pruebas y cobertura de código. Para cualquier tipo de informe descubierto, CodeBuild crea nuevos grupos de informes con el siguiente patrón:

```
<project-name>-<report-file-format>-AutoDiscovered
```

Note

Si los archivos de informe descubiertos tienen el mismo tipo de formato, se colocarán en el mismo grupo de informes o informe.

La detección automática de informes está configurada por las variables de entorno del proyecto:

CODEBUILD_CONFIG_AUTO_DISCOVER

Esta variable determina si la detección automática de informes está deshabilitada durante la compilación. De forma predeterminada, la detección automática de informes está habilitada para todas las compilaciones. Para deshabilitar esta función, `CODEBUILD_CONFIG_AUTO_DISCOVER` configúrela `false` en.

CODEBUILD_CONFIG_AUTO_DISCOVER_DIR

(Opcional) Esta variable determina dónde se CodeBuild buscan los posibles archivos de informes. Tenga en cuenta que, de forma `**/*` predeterminada, CodeBuild busca en

Estas variables de entorno se pueden modificar durante la fase de creación. Por ejemplo, si solo quieres habilitar la detección automática de informes para las compilaciones de la rama `main` git, puedes comprobar la rama git durante el proceso de compilación y `CODEBUILD_CONFIG_AUTO_DISCOVER` establecerla en `false` si la compilación no está en la `main` rama. La detección automática de informes se puede desactivar mediante la consola o mediante variables de entorno del proyecto.

Temas

- [Configure la detección automática de informes mediante la consola](#)
- [Configurar la detección automática de informes mediante variables de entorno del proyecto](#)

Configure la detección automática de informes mediante la consola

Para configurar la detección automática de informes mediante la consola

1. Cree un proyecto de compilación o elija un proyecto de compilación para editarlo. Para obtener información, consulte [Creación de un proyecto de compilación en AWS CodeBuild](#) o [Cambiar la configuración de un proyecto de compilación en AWS CodeBuild](#).
2. En Entorno, seleccione Configuración adicional.
3. Para deshabilitar la detección automática de informes, en la detección automática de informes, seleccione Desactivar la detección automática de informes.

4. (Opcional) En el directorio de detección automática (opcional), introduzca un patrón de directorio CodeBuild para buscar archivos con formato de informe compatibles. Tenga en cuenta que CodeBuild las búsquedas se realizan de forma `**/*` predeterminada.

Configurar la detección automática de informes mediante variables de entorno del proyecto

Para configurar la detección automática de informes mediante variables de entorno del proyecto

1. Cree un proyecto de compilación o elija un proyecto de compilación para editarlo. Para obtener información, consulte [Creación de un proyecto de compilación en AWS CodeBuild](#) o [Cambiar la configuración de un proyecto de compilación en AWS CodeBuild](#).
2. En Variables de entorno, haga lo siguiente:
 - a. Para deshabilitar la detección automática de informes, para Nombre, ingrese **CODEBUILD_CONFIG_AUTO_DISCOVER** y para Valor, ingrese **false**. Esto desactiva la detección automática de informes.
 - b. (Opcional) En Nombre, introduzca **CODEBUILD_CONFIG_AUTO_DISCOVER_DIR** y, en Valor, introduzca el directorio en el que se CodeBuild deben buscar los archivos con formato de informe compatibles. Por ejemplo, `output/*.xml` busca `.xml` archivos en el `output` directorio

Registro y monitorización en AWS CodeBuild

La monitorización es una parte importante del mantenimiento de la fiabilidad, la disponibilidad y el rendimiento de AWS CodeBuild y sus soluciones de AWS. Debería recopilar los datos de monitorización de todas las partes de la solución de AWS para poder depurar más fácilmente un error multipunto, si se produce. AWS proporciona las siguientes herramientas para monitorizar los recursos y compilaciones y responder a posibles incidencias.

Temas

- [Registrar llamadas a la API de AWS CodeBuild con AWS CloudTrail](#)
- [Monitorización de AWS CodeBuild](#)

Registrar llamadas a la API de AWS CodeBuild con AWS CloudTrail

AWS CodeBuild está integrado con AWS CloudTrail, un servicio que proporciona un registro de las acciones hechas por un usuario, un rol o un servicio de AWS en CodeBuild. CloudTrail captura todas las llamadas a la API de CodeBuild como eventos, incluidas las llamadas procedentes de la consola de CodeBuild y las llamadas de código a las API de CodeBuild. Si crea un registro de seguimiento, puede habilitar la entrega continua de eventos de CloudTrail a un bucket de S3, incluidos los eventos de CodeBuild. Si no configura un registro de seguimiento, puede ver los eventos más recientes de la consola de CloudTrail en el Historial de eventos. Mediante la información que recopila CloudTrail, puede determinar la solicitud que se hizo a CodeBuild, la dirección IP desde la que se hizo la solicitud, quién la hizo y cuándo e información adicional.

Para obtener más información acerca de CloudTrail, consulte la [Guía del usuario de AWS CloudTrail](#).

Información de AWS CodeBuild en CloudTrail

CloudTrail se habilita en su cuenta de AWS cuando la crea. Cuando se produce una actividad en CodeBuild, esa actividad se registra en un evento de CloudTrail junto con otros eventos de servicio de AWS en Historial de eventos. Puede ver, buscar y descargar los últimos eventos de la cuenta de AWS. Para obtener más información, consulte [Ver eventos con el historial de eventos de CloudTrail](#) en la Guía del usuario de AWS CloudTrail.

Para un registro continuo de los eventos de la cuenta de AWS, incluidos los eventos de CodeBuild, cree un registro de seguimiento. Un registro de seguimiento permite a CloudTrail que pueda enviar archivos de registro a un bucket de S3. De forma predeterminada, cuando se crea un registro de seguimiento en la consola, el registro de seguimiento se aplica a todas las regiones. El registro de seguimiento registra los eventos de todas las regiones de la partición de AWS y envía los archivos de registros al bucket de S3 especificado. También puede configurar otros servicios de AWS para analizar y actuar en función de los datos de eventos recopilados en los registros de CloudTrail. Para obtener más información, consulte:

- [Introducción a la creación de registros de seguimiento](#)
- [Servicios e integraciones compatibles con CloudTrail](#)
- [Configuración de notificaciones de Amazon SNS para CloudTrail](#)
- [Recibir archivos de registro de CloudTrail de varias regiones](#) y [Recibir archivos de registro de CloudTrail de varias cuentas](#)

Todas las acciones de CodeBuild las registra CloudTrail y están documentadas en la [Referencia de la API de CodeBuild](#). Por ejemplo, las llamadas a las acciones de CreateProject (en la AWS CLI, create-project), StartBuild (en la AWS CLI, start-project) y UpdateProject (en la AWS CLI, update-project) generan entradas en los archivos de registro de CloudTrail.

Cada entrada de registro o evento contiene información sobre quién generó la solicitud. La información de identidad del usuario le ayuda a determinar lo siguiente:

- Si la solicitud se realizó con las credenciales raíz o del usuario de .
- Si la solicitud se realizó con credenciales de seguridad temporales de un rol o fue un usuario federado.
- Si la solicitud la realizó otro servicio de AWS.

Para obtener más información, consulte el [Elemento userIdentity de CloudTrail](#) en la Guía del usuario de AWS CloudTrail.

Descripción de las entradas de los archivos de registro de AWS CodeBuild

Un registro de seguimiento es una configuración que permite la entrega de eventos como archivos de log al bucket de S3 que se especifique. Los archivos log de CloudTrail pueden contener una o varias entradas de log. Un evento representa una única solicitud de cualquier origen e incluye información acerca de la acción solicitada, la fecha y la hora de la acción, los parámetros de la solicitud, etcétera.

Los archivos de registro de CloudTrail no son un rastro de pila ordenado de las llamadas a las API públicas, por lo que no aparecen en ningún orden específico.

Note

Para proteger la información confidencial, lo siguiente está oculto en los registros de CodeBuild:

- ID de clave de acceso de AWS. Para obtener más información, consulte [Administración de claves de acceso para usuarios de IAM](#) en la Guía del usuario de AWS Identity and Access Management.
- Cadenas especificadas mediante el almacén de parámetros. Para obtener más información, consulte [Almacén de parámetros de Systems Manager](#) y [Tutorial de la consola del almacén de parámetros de Systems Manager](#) en la Guía del usuario de Amazon EC2 Systems Manager.
- Cadenas especificadas mediante AWS Secrets Manager. Para obtener más información, consulte [Administración de claves](#).

En el ejemplo siguiente, se muestra una entrada de registro de CloudTrail que ilustra la creación de un proyecto de compilación en CodeBuild.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "FederatedUser",
    "principalId": "account-ID:user-name",
    "arn": "arn:aws:sts::account-ID:federated-user/user-name",
    "accountId": "account-ID",
    "accessKeyId": "access-key-ID",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2016-09-06T17:59:10Z"
      }
    },
    "sessionIssuer": {
      "type": "IAMUser",
      "principalId": "access-key-ID",
      "arn": "arn:aws:iam::account-ID:user/user-name",
      "accountId": "account-ID",
```

```

    "userName": "user-name"
  }
}
},
"eventTime": "2016-09-06T17:59:11Z",
"eventSource": "codebuild.amazonaws.com",
"eventName": "CreateProject",
"awsRegion": "region-ID",
"sourceIPAddress": "127.0.0.1",
"userAgent": "user-agent",
"requestParameters": {
  "awsActId": "account-ID"
},
"responseElements": {
  "project": {
    "environment": {
      "image": "image-ID",
      "computeType": "BUILD_GENERAL1_SMALL",
      "type": "LINUX_CONTAINER",
      "environmentVariables": []
    },
    "name": "codebuild-demo-project",
    "description": "This is my demo project",
    "arn": "arn:aws:codebuild:region-ID:account-ID:project/codebuild-demo-
project:project-ID",
    "encryptionKey": "arn:aws:kms:region-ID:key-ID",
    "timeoutInMinutes": 10,
    "artifacts": {
      "location": "arn:aws:s3:::codebuild-region-ID-account-ID-output-bucket",
      "type": "S3",
      "packaging": "ZIP",
      "outputName": "MyOutputArtifact.zip"
    },
    "serviceRole": "arn:aws:iam::account-ID:role/CodeBuildServiceRole",
    "lastModified": "Sep 6, 2016 10:59:11 AM",
    "source": {
      "type": "GITHUB",
      "location": "https://github.com/my-repo.git"
    },
    "created": "Sep 6, 2016 10:59:11 AM"
  }
},
"requestID": "9d32b228-745b-11e6-98bb-23b67EXAMPLE",
"eventID": "581f7dd1-8d2e-40b0-aeee-0dbf7EXAMPLE",

```

```
"eventType": "AwsApiCall",  
"recipientAccountId": "account-ID"  
}
```

Monitorización de AWS CodeBuild

Es posible utilizar Amazon CloudWatch para observar las compilaciones, informar cuando algo no funcione y realizar acciones automáticas cuando proceda. Puede monitorizar las compilaciones en dos niveles:

Nivel del proyecto

Nivel del proyecto: estas métricas son para todas las compilaciones solo en el proyecto especificado. Para ver las métricas de un proyecto, especifique `ProjectName` para la dimensión en CloudWatch.

Nivel de cuenta de AWS

Estas métricas son para todas las compilaciones de una cuenta. Para ver las métricas en el nivel de la cuenta de AWS, no especifique ninguna dimensión en CloudWatch. Las métricas de uso de los recursos de compilación no están disponibles en el nivel de cuenta de AWS.

Las métricas de CloudWatch muestran el comportamiento de las compilaciones a lo largo del tiempo. Por ejemplo, es posible monitorizar:

- Cuántas compilaciones se intentaron en un proyecto de compilación o en una cuenta de AWS con el paso del tiempo.
- Cuántas compilaciones se intentaron con éxito en un proyecto de compilación o en una cuenta de AWS con el paso del tiempo.
- Cuántas compilaciones mostraron errores en un proyecto de compilación o en una cuenta de AWS con el paso del tiempo.
- Cantidad de tiempo invertido por CodeBuild en ejecutar las compilaciones de un proyecto de compilación o una cuenta de AWS a lo largo del tiempo.
- Utilización de los recursos de compilación para una compilación o un proyecto de compilación completo. Las métricas de utilización de los recursos de compilación incluyen métricas como la utilización de la CPU, la memoria y el almacenamiento.

Para obtener más información, consulte [Monitorización de métricas de CodeBuild](#).

Métricas de CloudWatch de CodeBuild

Es posible realizar el seguimiento de las métricas siguientes por cuenta de AWS o por proyecto de compilación.

BuildDuration

Mide la duración de la fase BUILD de la compilación.

Unidades: segundos

Estadísticas de CloudWatch válidas: Media (recomendado), Máximo, Mínimo

Compilaciones

Mide el número de compilaciones activadas.

Unidades: recuento

Estadísticas de CloudWatch válidas: Suma

DownloadSourceDuration

Mide la duración de la fase DOWNLOAD_SOURCE de la compilación.

Unidades: segundos

Estadísticas de CloudWatch válidas: Media (recomendado), Máximo, Mínimo

Duración

Mide la duración de todas las compilaciones con el paso del tiempo.

Unidades: segundos

Estadísticas de CloudWatch válidas: Media (recomendado), Máximo, Mínimo

FailedBuilds

Mide el número de compilaciones con errores debido al error del cliente o a un tiempo de espera agotado.

Unidades: recuento

Estadísticas de CloudWatch válidas: Suma

FinalizingDuration

Mide la duración de la fase FINALIZING de la compilación.

Unidades: segundos

Estadísticas de CloudWatch válidas: Media (recomendado), Máximo, Mínimo

InstallDuration

Mide la duración de la fase INSTALL de la compilación.

Unidades: segundos

Estadísticas de CloudWatch válidas: Media (recomendado), Máximo, Mínimo

PostBuildDuration

Mide la duración de la fase POST_BUILD de la compilación.

Unidades: segundos

Estadísticas de CloudWatch válidas: Media (recomendado), Máximo, Mínimo

PreBuildDuration

Mide la duración de la fase PRE_BUILD de la compilación.

Unidades: segundos

Estadísticas de CloudWatch válidas: Media (recomendado), Máximo, Mínimo

ProvisioningDuration

Mide la duración de la fase PROVISIONING de la compilación.

Unidades: segundos

Estadísticas de CloudWatch válidas: Media (recomendado), Máximo, Mínimo

QueuedDuration

Mide la duración de la fase QUEUED de la compilación.

Unidades: segundos

Estadísticas de CloudWatch válidas: Media (recomendado), Máximo, Mínimo

SubmittedDuration

Mide la duración de la fase SUBMITTED de la compilación.

Unidades: segundos

Estadísticas de CloudWatch válidas: Media (recomendado), Máximo, Mínimo

SucceededBuilds

Mide el número de compilaciones correctas.

Unidades: recuento

Estadísticas de CloudWatch válidas: Suma

UploadArtifactsDuration

Mide la duración de la fase UPLOAD_ARTIFACTS de la compilación.

Unidades: segundos

Estadísticas de CloudWatch válidas: Media (recomendado), Máximo, Mínimo

Métricas de CloudWatch de la utilización de recursos de CodeBuild

Note

Las métricas de utilización de recursos de CodeBuild solo están disponibles en las regiones siguientes:

- Región Asia-Pacífico (Tokio)
- Región Asia-Pacífico (Seúl)
- Región Asia-Pacífico (Mumbai)
- Asia Pacific (Singapore) Region
- Asia Pacific (Sydney) Region
- Canada (Central) Region
- Europe (Frankfurt) Region
- Europe (Ireland) Region

- Europe (London) Region
- Región de Europa (París)
- South America (São Paulo) Region
- US East (N. Virginia) Region
- US East (Ohio) Region
- Región del oeste de EE. UU (N. California)
- Región del oeste de EE. UU (Oregon)

Es posible realizar un seguimiento de las métricas de utilización de recursos siguientes.

CPUUtilized

El número de unidades de CPU de procesamiento asignadas utilizadas por el contenedor de compilación.

Unidades: unidades de CPU

Estadísticas de CloudWatch válidas: Media (recomendado), Máximo, Mínimo

CPUUtilizedPercent

Porcentaje procesamiento asignado utilizado por el contenedor de compilación.

Unidades: porcentaje

Estadísticas de CloudWatch válidas: Media (recomendado), Máximo, Mínimo

MemoryUtilized

Número de megabytes de memoria utilizados por el contenedor de compilación.

Unidades: megabytes

Estadísticas de CloudWatch válidas: Media (recomendado), Máximo, Mínimo

MemoryUtilizedPercent

Porcentaje memoria asignada utilizada por el contenedor de compilación.

Unidades: porcentaje

Estadísticas de CloudWatch válidas: Media (recomendado), Máximo, Mínimo

StorageReadBytes

Velocidad de lectura de almacenamiento utilizada por el contenedor de compilación.

Unidades: bytes/segundo

Estadísticas de CloudWatch válidas: Media (recomendado), Máximo, Mínimo

StorageWriteBytes

Velocidad de escritura de almacenamiento utilizada por el contenedor de compilación.

Unidades: bytes/segundo

Estadísticas de CloudWatch válidas: Media (recomendado), Máximo, Mínimo

Dimensiones de CloudWatch de CodeBuild

CodeBuild proporciona las dimensiones métricas de CloudWatch siguientes. Si no se especifica ninguna de ellas, las métricas son para la cuenta de AWS actual.

BuildId, BuildNumber, ProjectName

Se proporcionan métricas para el identificador de compilación, el número de compilación y el nombre del proyecto.

ProjectName

Las métricas se proporcionan para un nombre de proyecto.

Alarmas de CloudWatch de CodeBuild

Puede utilizar la consola de CloudWatch para crear alarmas en función de las métricas de CodeBuild, lo que le permitirá reaccionar si hay algún problema con las compilaciones. Las dos métricas que son más útiles con alarmas son:

- **FailedBuild.** Puede crear una alarma que se active cuando se detecte un determinado número de compilaciones con errores en un número de segundos predeterminado. En CloudWatch, especifique el número de segundos y de compilaciones con errores que activarán una alarma.

- **Duration.** Puede crear una alarma que se activa cuando una compilación tarda más de lo esperado. Especifique la cantidad de segundos que debe transcurrir después de que se inicie una compilación y antes de que se complete una compilación para activar una alarma.

Para obtener más información acerca de cómo crear alarmas para métricas de CodeBuild, consulte [Monitorización de compilaciones con alarmas de CloudWatch](#). Para obtener más información sobre las alarmas, consulte [Crear alarmas de Amazon CloudWatch](#) en la Guía del usuario de Amazon CloudWatch.

Monitorización de métricas de CodeBuild

AWS CodeBuild monitoriza funciones en su nombre e informa sobre las métricas a través de Amazon CloudWatch. Estas métricas incluyen el número de compilaciones totales, compilaciones con errores, compilaciones correctas y la duración de las compilaciones.

Puede utilizar la consola de CodeBuild o la consola de CloudWatch para monitorizar métricas de CodeBuild. Los procedimientos siguientes muestran cómo acceder a las métricas.

Temas

- [Acceder a las métricas de compilación \(consola de CodeBuild\)](#)
- [Acceder a las métricas de compilación \(consola de Amazon CloudWatch\)](#)

Acceder a las métricas de compilación (consola de CodeBuild)

Note

No es posible personalizar las métricas ni los gráficos utilizados para mostrarlas en la consola de CodeBuild. Si desea personalizar la visualización, utilice la consola de Amazon CloudWatch para ver las métricas de compilación.

Métricas de nivel de cuenta

Para acceder a las métricas de nivel de cuenta de AWS

1. Inicie sesión en la AWS Management Console y abra la consola de AWS CodeBuild en <https://console.aws.amazon.com/codesuite/codebuild/home>.

2. En el panel de navegación, seleccione Métricas de la cuenta.

Métricas de nivel de proyecto

Para acceder a las métricas de nivel de proyecto

1. Inicie sesión en la AWS Management Console y abra la consola de AWS CodeBuild en <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. En el panel de navegación, elija Proyectos de compilación.
3. En la lista de proyectos de compilación, en la columna Name (Nombre), elija el proyecto en el que desee ver las métricas.
4. Elija la pestaña Metrics (Métricas).

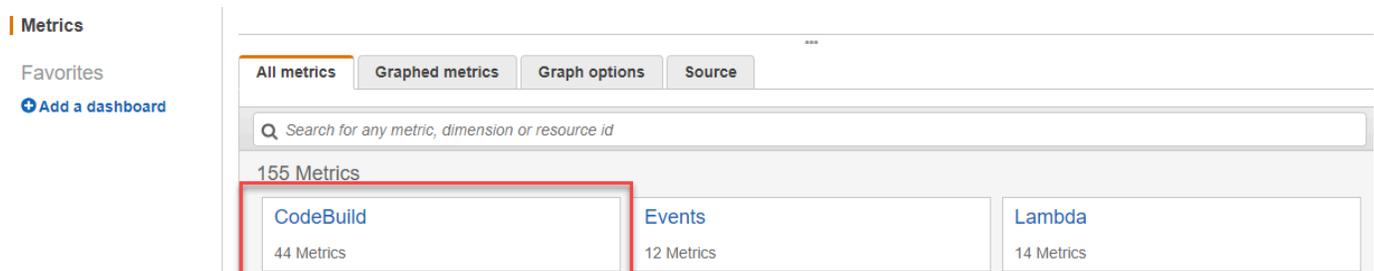
Acceder a las métricas de compilación (consola de Amazon CloudWatch)

Puede personalizar las métricas y los gráficos utilizados para mostrarlas en la consola de CloudWatch.

Métricas de nivel de cuenta

Para acceder a las métricas de nivel de cuenta

1. Inicie sesión en la AWS Management Console y abra la consola de CloudWatch en <https://console.aws.amazon.com/cloudwatch/>.
2. En el panel de navegación, seleccione Métricas.
3. En la pestaña Todas las métricas, seleccione CodeBuild.

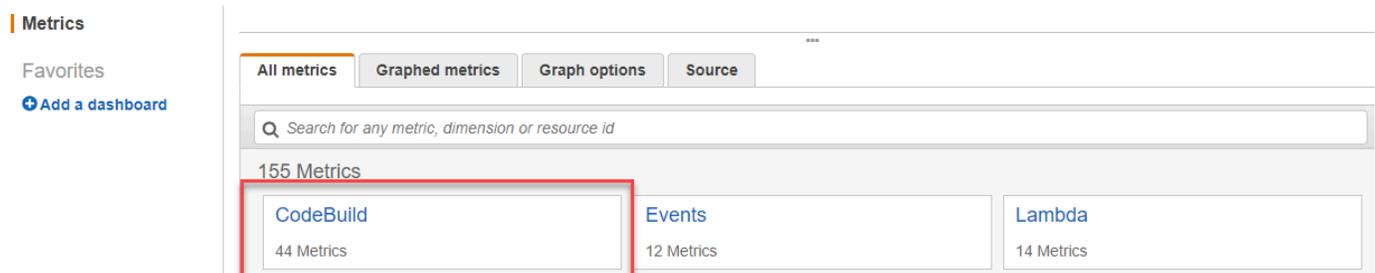


4. Elija Account Metrics (Métricas de cuenta).
5. Elija uno o varios proyectos y métricas. Por cada proyecto, puede elegir las métricas SucceededBuilds, FailedBuilds, Builds y Duration. Todas las combinaciones de métricas y proyectos seleccionados se muestran en el gráfico de la página.

Métricas de nivel de proyecto

Para acceder a las métricas de nivel de proyecto

1. Inicie sesión en la AWS Management Console y abra la consola de CloudWatch en <https://console.aws.amazon.com/cloudwatch/>.
2. En el panel de navegación, seleccione Métricas.
3. En la pestaña Todas las métricas, seleccione CodeBuild.



4. Elija By Project (Por proyecto).
5. Elija uno o varios combinaciones de proyectos y métricas. Por cada proyecto, puede elegir las métricas SucceededBuilds, FailedBuilds, Builds y Duration. Todas las combinaciones de métricas y proyectos seleccionados se muestran en el gráfico de la página.
6. (Opcional) Puede personalizar las métricas y los gráficos. Por ejemplo, en la lista desplegable de la columna Estadística, puede elegir otra estadística para mostrar. O bien, en el menú desplegable de la columna Período, puede elegir un periodo de tiempo distinto para utilizarlo con el fin de monitorizar las métricas.

Para obtener más información, consulte [Representación gráfica de métricas](#) y [Visualización de las métricas disponibles](#) en la Guía del usuario de Amazon CloudWatch.

Monitorización de las métricas de utilización de recursos de CodeBuild

AWS CodeBuild monitoriza la utilización de recursos de compilación en su nombre e informa sobre las métricas mediante Amazon CloudWatch. Estas incluyen métricas como utilización de CPU, memoria y almacenamiento.

Note

Las métricas de utilización de recursos de CodeBuild solo se registran para las compilaciones que se ejecutan durante más de un minuto.

Es posible utilizar la consola de CodeBuild o la consola de CloudWatch para monitorizar las métricas de utilización de recursos de CodeBuild.

Note

Las métricas de utilización de recursos de CodeBuild solo están disponibles en las regiones siguientes:

- Región de Asia-Pacífico (Tokio)
- Región de Asia-Pacífico (Seúl)
- Región de Asia-Pacífico (Bombay)
- Región de Asia-Pacífico (Singapur)
- Región de Asia-Pacífico (Sídney)
- Región de Canadá (centro)
- Región de Europa (Fráncfort)
- Región de Europa (Irlanda)
- Región de Europa (Londres)
- Región de Europa (París)
- Región de América del Sur (São Paulo)
- Región del este de EE. UU. (Norte de Virginia)
- Región del este de EE. UU. (Ohio)
- Región del oeste de EE. UU. (Norte de California)
- Región del oeste de EE. UU. (Oregón)

Los procedimientos siguientes muestran cómo acceder a las métricas de utilización de recursos.

Temas

- [Acceder a las métricas de utilización de recursos \(consola de CodeBuild\)](#)
- [Acceso a las métricas de utilización de recursos \(consola de Amazon CloudWatch\)](#)

Acceder a las métricas de utilización de recursos (consola de CodeBuild)

Note

No es posible personalizar las métricas ni los gráficos utilizados para mostrarlas en la consola de CodeBuild. Si desea personalizar la visualización, utilice la consola de Amazon CloudWatch para ver las métricas de compilación.

Métricas de utilización de recursos en el nivel de proyecto

Para acceder a las métricas de utilización de recursos en el nivel de proyecto

1. Inicie sesión en la AWS Management Console y abra la consola de AWS CodeBuild en <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. En el panel de navegación, elija Proyectos de compilación.
3. En la lista de proyectos de compilación, en la columna Nombre, elija el proyecto del que desee ver las métricas.
4. Elija la pestaña Metrics (Métricas). Las métricas de utilización de los recursos se muestran en la sección Métricas de utilización de los recursos.
5. Para ver las métricas de utilización de los recursos en el nivel de proyecto en la consola de CloudWatch, seleccione Ver en CloudWatch en la sección Métricas de uso de recursos.

Métricas de utilización de recursos en el nivel de compilación

Para acceder a las métricas de utilización de recursos en el nivel de compilación

1. Inicie sesión en la AWS Management Console y abra la consola de AWS CodeBuild en <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. En el panel de navegación, elija Historial de compilaciones.
3. En la lista de compilaciones, en la columna Ejecución de la compilación, elija la compilación cuyas métricas desee ver.
4. Seleccione la pestaña Utilización de recursos.
5. Para ver las métricas de uso de recursos en el nivel de compilación en la consola de CloudWatch, seleccione Ver en CloudWatch en la sección Métricas de utilización de recursos.

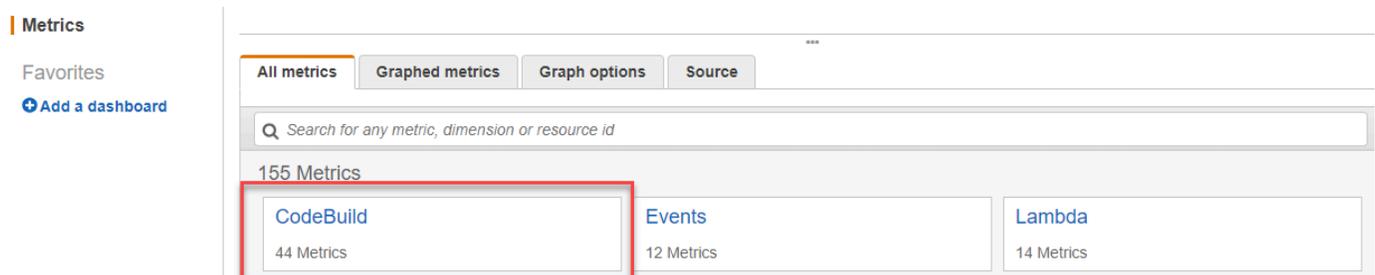
Acceso a las métricas de utilización de recursos (consola de Amazon CloudWatch)

La consola de Amazon CloudWatch se puede utilizar para acceder a las métricas de utilización de recursos de CodeBuild.

Métricas de utilización de recursos en el nivel de proyecto

Para acceder a las métricas de utilización de recursos en el nivel de proyecto

1. Inicie sesión en la AWS Management Console y abra la consola de CloudWatch en <https://console.aws.amazon.com/cloudwatch/>.
2. En el panel de navegación, seleccione Métricas.
3. En la pestaña Todas las métricas, seleccione CodeBuild.



4. Elija Por proyecto.
5. Elija una o más combinaciones de proyectos y métricas para añadirlas al gráfico. Todas las combinaciones de métricas y proyectos seleccionados se muestran en el gráfico de la página.
6. (Opcional) Es posible personalizar las métricas y los gráficos desde la pestaña Métricas diagramadas. Por ejemplo, en la lista desplegable de la columna Estadística, puede elegir otra estadística para mostrar. O bien, en el menú desplegable de la columna Período, puede elegir un periodo de tiempo distinto para utilizarlo con el fin de monitorizar las métricas.

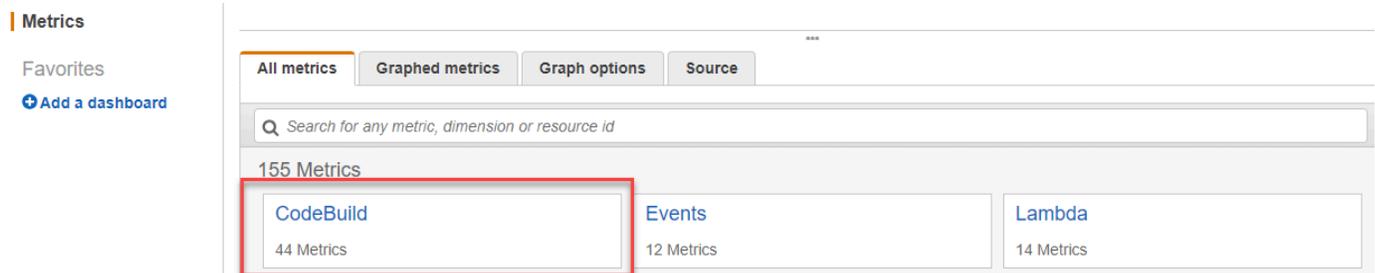
Para obtener más información, consulte [Representación gráfica de métricas](#) y [Visualización de las métricas disponibles](#) en la Guía del usuario de Amazon CloudWatch.

Métricas de utilización de recursos en el nivel de compilación

Para acceder a las métricas de utilización de recursos en el nivel de compilación

1. Inicie sesión en la AWS Management Console y abra la consola de CloudWatch en <https://console.aws.amazon.com/cloudwatch/>.
2. En el panel de navegación, seleccione Métricas.

3. En la pestaña Todas las métricas, seleccione CodeBuild.



4. Elija BuildID, BuildNumber, ProjectName.

5. Elija una o más combinaciones de compilaciones y métricas para añadirlas al gráfico. Todas las combinaciones de compilaciones y métricas seleccionadas se muestran en el gráfico de la página.

6. (Opcional) Es posible personalizar las métricas y los gráficos desde la pestaña Métricas diagramadas. Por ejemplo, en la lista desplegable de la columna Estadística, puede elegir otra estadística para mostrar. O bien, en el menú desplegable de la columna Período, puede elegir un periodo de tiempo distinto para utilizarlo con el fin de monitorizar las métricas.

Para obtener más información, consulte [Representación gráfica de métricas](#) y [Visualización de las métricas disponibles](#) en la Guía del usuario de Amazon CloudWatch.

Monitorización de compilaciones con alarmas de CloudWatch

Puede crear una alarma de CloudWatch para sus compilaciones. Una alarma vigila una única métrica durante el periodo de tiempo especificado y realiza una o varias acciones en función del valor de la métrica en relación con un determinado umbral durante una serie de periodos de tiempo. Con la funcionalidad de alarmas nativa de CloudWatch, es posible especificar cualquiera de las acciones admitidas por CloudWatch cuando se supera un umbral. Por ejemplo, es posible especificar que se envíe una notificación Amazon SNS cuando fallen más de tres compilaciones en su cuenta dentro de un plazo de quince minutos.

Para crear una alarma de CloudWatch para una métrica de CodeBuild

1. Inicie sesión en la AWS Management Console y abra la consola de CloudWatch en <https://console.aws.amazon.com/cloudwatch/>.
2. En el panel de navegación, elija Alarms.
3. Elija Create Alarm (Crear alarma).

4. En CloudWatch Metrics by Category (Métricas de CloudWatch por categoría), elija la categoría CodeBuild Metrics (Métricas CodeBuild). Si sabe que desea solo las métricas en el nivel de proyecto, elija By Project (Por proyecto). Si sabe que desea solo las métricas en el nivel de cuenta, elija Account Metrics (Métricas de cuenta).
5. En Create Alarm (Crear alarma) si aún no está seleccionada, elija Select Metric (Seleccionar métrica).
6. Seleccione una métrica para la que desee crear una alarma. Las opciones son By Project (Por proyecto) o Account Metrics (Métricas de cuenta).
7. Elija Next (Siguiente) o Define Alarm (Definir alarma) y, a continuación, cree la alarma. Para obtener más información, consulte [Creación de las alarmas de Amazon CloudWatch](#) en la Guía del usuario de Amazon CloudWatch. Para obtener más información sobre la configuración de las notificaciones de Amazon SNS cuando se desencadena una alarma, consulte [Configuración de notificaciones de Amazon SNS](#) en la Guía del desarrollador de Amazon SNS.
8. Elija Create Alarm (Crear alarma).

Seguridad en AWS CodeBuild

La seguridad en la nube AWS es la máxima prioridad. Como AWS cliente, usted se beneficia de una arquitectura de centro de datos y red diseñada para cumplir con los requisitos de las organizaciones más sensibles a la seguridad.

La seguridad y el cumplimiento son una responsabilidad compartida entre usted AWS y usted. Este modelo compartido puede ayudarlo a aliviar su carga operativa: AWS opera, administra y controla los componentes desde el sistema operativo anfitrión y la capa de virtualización hasta la seguridad física de las instalaciones de servicio. Usted asume la responsabilidad y la administración del sistema operativo invitado (incluidas las actualizaciones y los parches de seguridad) y de cualquier otro software de aplicaciones asociadas. También es responsable de la configuración del firewall del grupo de seguridad AWS proporcionado. Sus responsabilidades variarán en función de los servicios que utilice, la integración de estos con su entorno de TI y las leyes y regulaciones aplicables. Por lo tanto, debe prestar atención cuando valore los servicios que utiliza su organización. Para obtener más información, consulte el [Modelo de responsabilidad compartida](#).

Para obtener información sobre cómo proteger sus CodeBuild recursos, consulte los siguientes temas.

Temas

- [Protección de datos en AWS CodeBuild](#)
- [Administración de identidades y accesos en AWS CodeBuild](#)
- [Validación de conformidad para AWS CodeBuild](#)
- [Resiliencia en AWS CodeBuild](#)
- [Seguridad de la infraestructura en AWS CodeBuild](#)
- [Acceda a su proveedor de origen en CodeBuild](#)
- [Prevención de la sustitución confusa entre servicios](#)

Protección de datos en AWS CodeBuild

El modelo de [responsabilidad AWS compartida modelo](#) se aplica a la protección de datos en AWS CodeBuild. Como se describe en este modelo, AWS es responsable de proteger la infraestructura global que ejecuta todos los Nube de AWS. Usted es responsable de mantener el control sobre el contenido alojado en esta infraestructura. Usted también es responsable de las tareas de

administración y configuración de seguridad para los Servicios de AWS que utiliza. Para obtener más información sobre la privacidad de los datos, consulte las [Preguntas frecuentes sobre la privacidad de datos](#). Para obtener información sobre la protección de datos en Europa, consulte la publicación de blog sobre el [Modelo de responsabilidad compartida de AWS y GDPR](#) en el Blog de seguridad de AWS .

Con fines de protección de datos, le recomendamos que proteja Cuenta de AWS las credenciales y configure los usuarios individuales con AWS IAM Identity Center o AWS Identity and Access Management (IAM). De esta manera, solo se otorgan a cada usuario los permisos necesarios para cumplir sus obligaciones laborales. También recomendamos proteger sus datos de la siguiente manera:

- Utilice la autenticación multifactor (MFA) en cada cuenta.
- Utilice SSL/TLS para comunicarse con los recursos. AWS Se recomienda el uso de TLS 1.2 y recomendamos TLS 1.3.
- Configure la API y el registro de actividad de los usuarios con. AWS CloudTrail
- Utilice soluciones de AWS cifrado, junto con todos los controles de seguridad predeterminados 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 por FIPS 140-2 para acceder a AWS través de una interfaz de línea de comandos o una API, utilice un punto final 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 introducir nunca información confidencial o sensible, como, por ejemplo, direcciones de correo electrónico de clientes, en etiquetas o campos de formato libre, tales como el campo Nombre. Esto incluye cuando trabaja CodeBuild o Servicios de AWS utiliza la consola, la API o los SDK. AWS CLI AWS Cualquier dato que ingrese en etiquetas o campos de formato libre utilizados para nombres se puede emplear para los registros de facturación o diagnóstico. Si proporciona una URL a un servidor externo, recomendamos encarecidamente que no incluya información de credenciales en la URL a fin de validar la solicitud para ese servidor.

Para proteger la información confidencial, los CodeBuild registros ocultan lo siguiente:

- Cadenas especificadas mediante el almacén de parámetros en las variables de entorno CodeBuild del proyecto o en la sección `buildspecenv/parameter-store`. Para obtener más información,

consulte [Almacén de parámetros de Systems Manager](#) y [Tutorial de la consola del almacén de parámetros de Systems Manager](#) en la Guía del usuario de Amazon EC2 Systems Manager.

- Cadenas especificadas mediante variables de entorno CodeBuild del proyecto o AWS Secrets Manager en la sección `buildspec. env/secrets-manager`. Para obtener más información, consulte [Administración de claves](#).

Para obtener más información sobre la protección de datos, consulte la entrada de blog relativa al [modelo de responsabilidad compartida de AWS y GDPR](#) en el blog de seguridad de AWS .

Temas

- [Cifrado de datos](#)
- [Administración de claves](#)
- [Privacidad de tráfico](#)

Cifrado de datos

El cifrado es una parte importante de la seguridad. CodeBuild Algunos cifrados, por ejemplo, para el cifrado de datos en tránsito se proporcionan de forma predeterminada y no es necesario que haga nada. Otros, por ejemplo, para el cifrado de datos en reposo, se pueden configurar cuando cree su proyecto o compilación.

- Cifrado de los datos en reposo: los artefactos de compilación, como la memoria caché, los registros, los archivos de datos de informes de pruebas sin procesar exportados y los resultados de la compilación, se cifran de forma predeterminada mediante Claves administradas por AWS. Si no desea utilizar estas CMK, debe crear y configurar una clave de cliente personalizada administrada por el cliente. Para obtener más información, consulte [Creación de claves de KMS](#) y [Conceptos de AWS Key Management Service](#) en la Guía del usuario de AWS Key Management Service .
- Puede almacenar el identificador de la clave AWS KMS que se CodeBuild utiliza para cifrar el artefacto de salida de la compilación en la `CODEBUILD_KMS_KEY_ID` variable de entorno. Para obtener más información, consulte [Variables de entorno en los entornos de compilación](#).
- Puede especificar una clave administrada por el cliente cuando cree un proyecto de compilación. Para obtener más información, consulte [Set the Encryption Key Using the Console](#) y [Establecimiento de la clave de cifrado utilizando la CLI](#).

Los volúmenes de Amazon Elastic Block Store de su flota de construcción se cifran de forma predeterminada mediante Claves administradas por AWS.

- Cifrado de los datos en tránsito: todas las comunicaciones entre los clientes CodeBuild CodeBuild y sus dependencias posteriores están protegidas mediante conexiones TLS que se firman mediante el proceso de firma de la versión 4 de Signature. Todos los CodeBuild puntos finales utilizan certificados SHA-256 gestionados por. AWS Private Certificate Authority Para más información, consulte [Proceso de firma de Signature Version 4](#) y [¿Qué es PCA de ACM?](#).
- Cifrado de artefactos de compilación: la función de CodeBuild servicio asociada al proyecto de compilación requiere acceso a una clave de KMS para cifrar los artefactos de salida de la compilación. De forma predeterminada, CodeBuild usa una Clave administrada de AWS para Amazon S3 en su AWS cuenta. Si no desea utilizar esta Clave administrada de AWS, debe crear y configurar una clave administrada por el cliente. Para obtener más información, consulte [Creación de una clave administrada por el cliente](#) y [Creación de claves](#) en la Guía para desarrolladores de AWS KMS .

Administración de claves

Puede proteger su contenido del uso no autorizado mediante el cifrado. Guarde sus claves de cifrado y AWS Secrets Manager, a continuación, conceda permiso CodeBuild al rol de servicio asociado al proyecto de compilación para obtener las claves de cifrado de su cuenta de Secrets Manager.

Para obtener más información, consulte [Creación y configuración de una clave administrada por el cliente para CodeBuild](#), [Creación de un proyecto de compilación en AWS CodeBuild](#), [Ejecutar una compilación en AWS CodeBuild](#) y [Tutorial: Almacenamiento y recuperación de un secreto](#).

Utilice la variable de CODEBUILD_KMS_KEY_ID entorno en un comando de compilación para obtener el identificador AWS KMS clave. Para obtener más información, consulte [Variables de entorno en los entornos de compilación](#).

Puede utilizar Secrets Manager para proteger credenciales en un registro privado que almacena una imagen de Docker utilizada para su entorno en tiempo de ejecución. Para obtener más información, consulte [Registro privado con AWS Secrets Manager una muestra para CodeBuild](#).

Privacidad de tráfico

Puede mejorar la seguridad de sus compilaciones configurándolas CodeBuild para usar un punto final de VPC de interfaz. Para ello, no necesita una gateway de Internet, ni un dispositivo NAT, ni

una gateway privada virtual. Tampoco es obligatorio configurarlo PrivateLink, aunque se recomienda hacerlo. Para obtener más información, consulte [Uso de puntos de enlace de la VPC](#). Para obtener más información sobre los puntos PrivateLink de enlace de la VPC, consulte [Acceder a AWS PrivateLink](#) los servicios mediante PrivateLink

Administración de identidades y accesos en AWS CodeBuild

El acceso AWS CodeBuild requiere credenciales. Esas credenciales deben tener permisos para acceder a AWS los recursos, como almacenar y recuperar artefactos de compilación en buckets de S3 y ver Amazon CloudWatch Logs para compilaciones. En las siguientes secciones, se describe cómo puede utilizar [AWS Identity and Access Management](#)(IAM) y cómo puede ayudar CodeBuild a proteger el acceso a sus recursos:

Descripción general de la administración de los permisos de acceso a sus recursos AWS CodeBuild

Cada AWS recurso es propiedad de una AWS cuenta y los permisos para crear un recurso o acceder a él se rigen por las políticas de permisos. Un administrador de cuentas puede asociar políticas de permisos a identidades de IAM (es decir, usuarios, grupos y funciones).

Note

Un administrador de cuentas (o usuario administrador) es un usuario que tiene privilegios de administrador. Para obtener más información, consulte [Prácticas recomendadas de IAM](#) en la Guía del usuario de IAM.

Cuando concede permisos, decide quién obtiene los permisos, los recursos a los que se tiene acceso y las acciones que se pueden realizar en esos recursos.

Temas

- [AWS CodeBuild recursos y operaciones](#)
- [Titularidad de los recursos](#)
- [Administración del acceso a los recursos](#)
- [Especificar elementos de la política: acciones, efectos y entidades principales](#)

AWS CodeBuild recursos y operaciones

En AWS CodeBuild, el recurso principal es un proyecto de compilación. En una política, se usa un nombre de recurso de Amazon (ARN) para identificar el recurso al que se aplica la política. Las compilaciones son también recursos y tienen ARN asociados. Para obtener más información, consulte [Amazon Resource Names \(ARN\) y AWS Service Namespaces](#) en Referencia general de Amazon Web Services

Tipo de recurso	Formato de ARN
Proyecto de compilación	<code>arn:aws:codebuild: <i>region-ID</i> :<i>account-ID</i> :project/<i>project-name</i></code>
Compilación	<code>arn:aws:codebuild: <i>region-ID</i> :<i>account-ID</i> :build/<i>build-ID</i></code>
Grupo de informes	<code>arn:aws:codebuild: <i>region-ID</i> :<i>account-ID</i> :report-group/<i>report-group-name</i></code>
Informar	<code>arn:aws:codebuild: <i>region-ID</i> :<i>account-ID</i> :report/<i>report-ID</i></code>
CodeBuild Todos los recursos	<code>arn:aws:codebuild:*</code>
Todos CodeBuild los recursos que pertenecen a la cuenta especificada en la AWS región especificada	<code>arn:aws:codebuild: <i>region-ID</i> :<i>account-ID</i> :*</code>

Note

La mayoría de AWS los servicios utilizan dos puntos (:) o una barra diagonal (/) como el mismo carácter en los ARN. Sin embargo, CodeBuild utiliza una coincidencia exacta en las reglas y patrones de recursos. Asegúrese de utilizar los caracteres correctos cuando cree patrones de eventos para que coincidan con la sintaxis de ARN en el recurso.

Por ejemplo, puede indicar un proyecto de compilación específico (*myBuildProject*) en su declaración utilizando su ARN de la siguiente manera:

```
"Resource": "arn:aws:codebuild:us-east-2:123456789012:project/myBuildProject"
```

Para especificar todos los recursos, o si una acción de API no admite ARN, utilice el carácter comodín (*) en el elemento Resource de la siguiente manera:

```
"Resource": "*"
```

Algunas acciones de la CodeBuild API aceptan varios recursos (por ejemplo, `BatchGetProjects`). Para especificar varios recursos en una única instrucción, separe sus ARN con comas, tal y como se indica a continuación:

```
"Resource": [  
  "arn:aws:codebuild:us-east-2:123456789012:project/myBuildProject",  
  "arn:aws:codebuild:us-east-2:123456789012:project/myOtherBuildProject"  
]
```

CodeBuild proporciona un conjunto de operaciones para trabajar con los CodeBuild recursos. Para ver una lista, consulte [AWS CodeBuild referencia de permisos](#).

Titularidad de los recursos

La AWS cuenta es propietaria de los recursos que se crean en la cuenta, independientemente de quién los haya creado. En concreto, el propietario del recurso es la AWS cuenta de la [entidad principal](#) (es decir, la cuenta raíz, un usuario o un rol de IAM) que autentica la solicitud de creación de recursos. Los siguientes ejemplos ilustran cómo funciona:

- Si utilizas las credenciales de la cuenta raíz de tu AWS cuenta para crear una regla, tu AWS cuenta es la propietaria del CodeBuild recurso.
- Si creas un usuario en tu AWS cuenta y le concedes permisos para crear CodeBuild recursos, el usuario podrá crear CodeBuild recursos. Sin embargo, su AWS cuenta, a la que pertenece el usuario, es propietaria de los CodeBuild recursos.
- Si crea un rol de IAM en su AWS cuenta con permisos para crear CodeBuild recursos, cualquier persona que pueda asumir el rol podrá crear CodeBuild recursos. Tu AWS cuenta, a la que pertenece el rol, es propietaria de los CodeBuild recursos.

Administración del acceso a los recursos

Una política de permisos describe quién tiene acceso a qué recursos.

Note

En esta sección, se describe cómo se utiliza IAM en AWS CodeBuild. No se proporciona información detallada sobre el servicio de IAM. Para ver la documentación completa de IAM, consulte [¿Qué es IAM?](#) en la Guía del usuario de IAM. Para obtener más información acerca de la sintaxis y las descripciones de las políticas del IAM, consulte [Referencia de políticas de IAM de AWS](#) en la Guía del usuario de IAM.

Las políticas que se asocian a una identidad de IAM se denominan políticas basadas en identidades (o políticas de IAM). Las políticas asociadas a un recurso se denominan políticas basadas en recursos. CodeBuild admite políticas basadas en la identidad y políticas basadas en recursos para determinadas API de solo lectura con el fin de compartir recursos entre cuentas.

Acceso seguro a los buckets de S3

Le recomendamos encarecidamente que incluya los siguientes permisos en su función de IAM para comprobar que el bucket de S3 asociado a su CodeBuild proyecto es de su propiedad o de alguien de su confianza. Estos permisos no están incluidos en las políticas y funciones AWS administradas. Debe agregarlos usted.

- `s3:GetBucketAc1`
- `s3:GetBucketLocation`

Si cambia el propietario de un bucket de S3 que se utiliza en el proyecto, debe asegurarse de que usted sigue siendo el propietario del bucket y, si no es así, actualizar los permisos en el rol de rol de IAM. Para obtener más información, consulte [Agregar permisos de acceso de CodeBuild a un grupo o un usuario de IAM](#) y [Crear un rol de servicio de CodeBuild](#).

Especificar elementos de la política: acciones, efectos y entidades principales

Para cada AWS CodeBuild recurso, el servicio define un conjunto de operaciones de API. Para conceder permisos para estas operaciones de API, CodeBuild define un conjunto de acciones que puede especificar en una política. Algunas operaciones de API pueden requerir permisos para más

de una acción para poder realizar la operación de API. Para obtener más información, consulte [AWS CodeBuild recursos y operaciones](#) y [AWS CodeBuild referencia de permisos](#).

A continuación, se indican los elementos básicos de la política:

- **Recurso:** use un Nombre de recurso de Amazon (ARN) para identificar el recurso al que se aplica la política.
- **Acción:** use palabras de clave de acción para identificar las operaciones del recurso que desea permitir o denegar. Por ejemplo, el permiso `codebuild:CreateProject` concede al usuario permiso para realizar la operación `CreateProject`.
- **Efecto:** especifique el efecto (permitir o denegar) cuando el usuario solicite la acción específica. Si no concede acceso de forma explícita (permitir) a un recurso, el acceso se deniega implícitamente. También puede denegar explícitamente el acceso a un recurso. Puede hacer esto para asegurarse de que un usuario no pueda tener acceso a un recurso, aunque otra política le otorgue acceso.
- **Entidad principal:** en las políticas basadas en identidades (políticas de IAM), el usuario al que se asocia esta política es la entidad principal implícita. Para las políticas basadas en recursos, debe especificar el usuario, cuenta, servicio u otra entidad que desee que reciba permisos.

Para obtener más información sobre la sintaxis y descripciones de las políticas de IAM, consulte [Referencia de la política de IAM de AWS](#) en la Guía del usuario de IAM.

Para ver una tabla en la que se muestran todas las acciones de la CodeBuild API y los recursos a los que se aplican, consulta la [AWS CodeBuild referencia de permisos](#).

Uso de políticas basadas en la identidad para AWS CodeBuild

En este tema se ofrecen ejemplos de políticas basadas en identidad que muestran cómo un administrador de la cuenta puede asociar políticas de permisos a identidades de IAM (es decir, usuarios, grupos y roles) y, de ese modo, conceder permisos para realizar operaciones en recursos de AWS CodeBuild .

Important

Le recomendamos que revise primero los temas introductorios que explican los conceptos básicos y las opciones disponibles para administrar el acceso a sus CodeBuild recursos. Para obtener más información, consulte [Descripción general de la administración de los permisos de acceso a sus recursos AWS CodeBuild](#).

Temas

- [Permisos necesarios para usar la consola de AWS CodeBuild](#)
- [Permisos necesarios para conectarse AWS CodeBuild a Amazon Elastic Container Registry](#)
- [Permisos necesarios para que la AWS CodeBuild consola se conecte a los proveedores de origen](#)
- [AWS políticas gestionadas \(predefinidas\) para AWS CodeBuild](#)
- [CodeBuild políticas y notificaciones gestionadas](#)
- [CodeBuild actualizaciones de las políticas AWS gestionadas](#)
- [Ejemplos de políticas administradas por los clientes](#)

A continuación se muestra un ejemplo de una política de permisos que permite a un usuario obtener información acerca de los proyectos de compilación únicamente en la región us-east-2 de la cuenta 123456789012 para cualquier proyecto de compilación que empiece con el nombre my:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:BatchGetProjects",
      "Resource": "arn:aws:codebuild:us-east-2:123456789012:project/my*"
    }
  ]
}
```

Permisos necesarios para usar la consola de AWS CodeBuild

El usuario que usa la AWS CodeBuild consola debe tener un conjunto mínimo de permisos que le permita describir otros AWS recursos de la AWS cuenta. Debe tener permisos de los siguientes servicios:

- AWS CodeBuild
- Amazon CloudWatch
- CodeCommit (si está almacenando su código fuente en un AWS CodeCommit repositorio)
- Amazon Elastic Container Registry (Amazon ECR) (si utiliza un entorno de compilación que se basa en una imagen de Docker en un repositorio de Amazon ECR)

Note

El 26 de julio de 2022, se actualizó la política de IAM predeterminada. Para obtener más información, consulte [Permisos necesarios para conectarse AWS CodeBuild a Amazon Elastic Container Registry](#).

- Amazon Elastic Container Service (Amazon ECS) (si utiliza un entorno de compilación que se basa en una imagen de Docker en un repositorio de Amazon ECR)
- AWS Identity and Access Management (IAM)
- AWS Key Management Service (AWS KMS)
- Amazon Simple Storage Service (Amazon S3)

Si crea una política de IAM que sea más restrictiva que los permisos mínimos necesarios, la consola no funcionará según lo previsto.

Permisos necesarios para conectarse AWS CodeBuild a Amazon Elastic Container Registry

El 26 de julio de 2022, AWS CodeBuild actualizó su política de IAM predeterminada para el permiso ECR de Amazon. Se han eliminado los permisos siguientes de la política predeterminada:

```
"ecr:PutImage",  
"ecr:InitiateLayerUpload",  
"ecr:UploadLayerPart",  
"ecr:CompleteLayerUpload"
```

Para CodeBuild los proyectos que se crearon antes del 26 de julio de 2022, le recomendamos que actualice su política con la siguiente política de Amazon ECR:

```
"Action": [  
  "ecr:BatchCheckLayerAvailability",  
  "ecr:GetDownloadUrlForLayer",  
  "ecr:BatchGetImage"  
]
```

Para obtener más información sobre cómo actualizar sus políticas, consulte [Agregar permisos de acceso de CodeBuild a un grupo o un usuario de IAM](#).

Permisos necesarios para que la AWS CodeBuild consola se conecte a los proveedores de origen

La AWS CodeBuild consola utiliza las siguientes acciones de la API para conectarse a los proveedores de fuentes (por ejemplo, GitHub los repositorios).

- `codebuild:ListConnectedOAuthAccounts`
- `codebuild:ListRepositories`
- `codebuild:PersistOAuthToken`
- `codebuild:ImportSourceCredentials`

Puedes asociar proveedores de código fuente (como GitHub repositorios) a tus proyectos de compilación mediante la AWS CodeBuild consola. Para ello, primero debes añadir las acciones de API anteriores a las políticas de acceso de IAM asociadas al usuario que utilizas para acceder a la AWS CodeBuild consola.

Las acciones API `ListConnectedOAuthAccounts`, `ListRepositories` y `PersistOAuthToken` no están diseñadas para que puedan invocarse desde el código. Por lo tanto, estas acciones de la API no se incluyen en los AWS SDK AWS CLI ni.

AWS políticas gestionadas (predefinidas) para AWS CodeBuild

AWS aborda muchos casos de uso comunes al proporcionar políticas de IAM independientes que son creadas y administradas por AWS. Estas políticas AWS gestionadas conceden los permisos necesarios para los casos de uso más habituales, de forma que no tengas que investigar qué permisos son necesarios. Las políticas gestionadas CodeBuild también proporcionan permisos para realizar operaciones en otros servicios, como IAM, Amazon EC2, AWS CodeCommit, Amazon ECR, Amazon SNS y CloudWatch Amazon Events, según lo exijan las responsabilidades de los usuarios a los que se ha concedido la política en cuestión. Por ejemplo, la `AWSCodeBuildAdminAccess` política es una política de usuario de nivel administrativo que permite a los usuarios con esta política crear y administrar reglas de CloudWatch eventos para compilaciones de proyectos y temas de Amazon SNS para notificaciones sobre eventos relacionados con proyectos (temas cuyos nombres llevan el prefijo `arn:aws:codebuild:`), así como administrar proyectos y grupos de informes en CodeBuild. Para más información, consulte [Políticas administradas de AWS](#) en la Guía del usuario de IAM.

Las siguientes políticas AWS administradas, que puede adjuntar a los usuarios de su cuenta, son específicas de AWS CodeBuild

AWSCodeBuildAdminAccess

Proporciona acceso completo e CodeBuild incluye permisos para administrar proyectos de CodeBuild compilación.

AWSCodeBuildDeveloperAccess

Proporciona acceso a la administración de proyectos de compilación, CodeBuild pero no la permite.

AWSCodeBuildReadOnlyAccess

Proporciona acceso de solo lectura a. CodeBuild

Para acceder a los artefactos de salida de compilación que CodeBuild crea, también debe adjuntar la política AWS administrada denominada. `AmazonS3ReadOnlyAccess`

Para crear y administrar las funciones de CodeBuild servicio, también debe adjuntar la política AWS administrada denominada `IAMFullAccess`.

También puedes crear tus propias políticas de IAM personalizadas para permitir permisos para CodeBuild acciones y recursos. Puede asociar estas políticas personalizadas a los usuarios o grupos de que requieran esos permisos.

Temas

- [AWSCodeBuildAdminAccess](#)
- [AWSCodeBuildDeveloperAccess](#)
- [AWSCodeBuildReadOnlyAccess](#)

AWSCodeBuildAdminAccess

La `AWSCodeBuildAdminAccess` política proporciona acceso completo a los proyectos de CodeBuild construcción CodeBuild, incluidos los permisos para administrarlos. Aplica esta política solo a los usuarios de nivel administrativo para darles el control total sobre los CodeBuild proyectos, los grupos de informes y los recursos relacionados de tu AWS cuenta, incluida la posibilidad de eliminar proyectos y grupos de informes.

La política `AWSCodeBuildAdminAccess` contiene la siguiente instrucción:

```
{
  "Version": "2012-10-17",
```

```
"Statement": [
  {
    "Sid": "AWSServicesAccess",
    "Action": [
      "codebuild:*",
      "codecommit:GetBranch",
      "codecommit:GetCommit",
      "codecommit:GetRepository",
      "codecommit:ListBranches",
      "codecommit:ListRepositories",
      "cloudwatch:GetMetricStatistics",
      "ec2:DescribeVpcs",
      "ec2:DescribeSecurityGroups",
      "ec2:DescribeSubnets",
      "ecr:DescribeRepositories",
      "ecr:ListImages",
      "elasticfilesystem:DescribeFileSystems",
      "events>DeleteRule",
      "events:DescribeRule",
      "events:DisableRule",
      "events:EnableRule",
      "events:ListTargetsByRule",
      "events:ListRuleNamesByTarget",
      "events:PutRule",
      "events:PutTargets",
      "events:RemoveTargets",
      "logs:GetLogEvents",
      "s3:GetBucketLocation",
      "s3:ListAllMyBuckets"
    ],
    "Effect": "Allow",
    "Resource": "*"
  },
  {
    "Sid": "CWLDeleteLogGroupAccess",
    "Action": [
      "logs>DeleteLogGroup"
    ],
    "Effect": "Allow",
    "Resource": "arn:aws:logs:*:*:log-group:/aws/codebuild/*:log-stream:*"
  },
  {
    "Sid": "SSMParameterWriteAccess",
    "Effect": "Allow",
```

```

    "Action": [
      "ssm:PutParameter"
    ],
    "Resource": "arn:aws:ssm:*:*:parameter/CodeBuild/*"
  },
  {
    "Sid": "SSMStartSessionAccess",
    "Effect": "Allow",
    "Action": [
      "ssm:StartSession"
    ],
    "Resource": "arn:aws:ecs:*:*:task/*/*"
  },
  {
    "Sid": "CodeStarConnectionsReadWriteAccess",
    "Effect": "Allow",
    "Action": [
      "codestar-connections:CreateConnection",
      "codestar-connections>DeleteConnection",
      "codestar-connections:UpdateConnectionInstallation",
      "codestar-connections:TagResource",
      "codestar-connections:UntagResource",
      "codestar-connections:ListConnections",
      "codestar-connections:ListInstallationTargets",
      "codestar-connections:ListTagsForResource",
      "codestar-connections:GetConnection",
      "codestar-connections:GetIndividualAccessToken",
      "codestar-connections:GetInstallationUrl",
      "codestar-connections:PassConnection",
      "codestar-connections:StartOAuthHandshake",
      "codestar-connections:UseConnection"
    ],
    "Resource": [
      "arn:aws:codestar-connections:*:*:connection/*",
      "arn:aws:codeconnections:*:*:connection/*"
    ]
  },
  {
    "Sid": "CodeStarNotificationsReadWriteAccess",
    "Effect": "Allow",
    "Action": [
      "codestar-notifications:CreateNotificationRule",
      "codestar-notifications:DescribeNotificationRule",
      "codestar-notifications:UpdateNotificationRule",

```

```

    "codestar-notifications:DeleteNotificationRule",
    "codestar-notifications:Subscribe",
    "codestar-notifications:Unsubscribe"
  ],
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "codestar-notifications:NotificationsForResource": "arn:aws:codebuild:*"
    }
  }
},
{
  "Sid": "CodeStarNotificationsListAccess",
  "Effect": "Allow",
  "Action": [
    "codestar-notifications:ListNotificationRules",
    "codestar-notifications:ListEventTypes",
    "codestar-notifications:ListTargets",
    "codestar-notifications:ListTagsForResource"
  ],
  "Resource": "*"
},
{
  "Sid": "CodeStarNotificationsSNSTopicCreateAccess",
  "Effect": "Allow",
  "Action": [
    "sns:CreateTopic",
    "sns:SetTopicAttributes"
  ],
  "Resource": "arn:aws:sns:*:*:codestar-notifications*"
},
{
  "Sid": "SNSTopicListAccess",
  "Effect": "Allow",
  "Action": [
    "sns:ListTopics",
    "sns:GetTopicAttributes"
  ],
  "Resource": "*"
},
{
  "Sid": "CodeStarNotificationsChatbotAccess",
  "Effect": "Allow",
  "Action": [

```

```
        "chatbot:DescribeSlackChannelConfigurations",
        "chatbot:ListMicrosoftTeamsChannelConfigurations"
    ],
    "Resource": "*"
}
]
```

AWSCodeBuildDeveloperAccess

La `AWSCodeBuildDeveloperAccess` política permite el acceso a todas las funciones y a los recursos relacionados con los CodeBuild grupos de proyectos e informes. Esta política no permite a los usuarios eliminar CodeBuild proyectos o grupos de informes ni recursos relacionados en otros AWS servicios, como CloudWatch los eventos. Le recomendamos que aplique esta política a la mayoría de los usuarios.

La política `AWSCodeBuildDeveloperAccess` contiene la siguiente instrucción:

```
{
  "Statement": [
    {
      "Sid": "AWSServicesAccess",
      "Action": [
        "codebuild:StartBuild",
        "codebuild:StopBuild",
        "codebuild:StartBuildBatch",
        "codebuild:StopBuildBatch",
        "codebuild:RetryBuild",
        "codebuild:RetryBuildBatch",
        "codebuild:BatchGet*",
        "codebuild:GetResourcePolicy",
        "codebuild:DescribeTestCases",
        "codebuild:DescribeCodeCoverages",
        "codebuild:List*",
        "codecommit:GetBranch",
        "codecommit:GetCommit",
        "codecommit:GetRepository",
        "codecommit:ListBranches",
        "cloudwatch:GetMetricStatistics",
        "events:DescribeRule",
        "events:ListTargetsByRule",
        "events:ListRuleNamesByTarget",
        "logs:GetLogEvents",
```

```
    "s3:GetBucketLocation",
    "s3:ListAllMyBuckets"
  ],
  "Effect": "Allow",
  "Resource": "*"
},
{
  "Sid": "SSMParameterWriteAccess",
  "Effect": "Allow",
  "Action": [
    "ssm:PutParameter"
  ],
  "Resource": "arn:aws:ssm:*:*:parameter/CodeBuild/*"
},
{
  "Sid": "SSMStartSessionAccess",
  "Effect": "Allow",
  "Action": [
    "ssm:StartSession"
  ],
  "Resource": "arn:aws:ecs:*:*:task/*/*"
},
{
  "Sid": "CodeStarConnectionsUserAccess",
  "Effect": "Allow",
  "Action": [
    "codestar-connections:ListConnections",
    "codestar-connections:GetConnection"
  ],
  "Resource": [
    "arn:aws:codestar-connections:*:*:connection/*",
    "arn:aws:codeconnections:*:*:connection/*"
  ]
},
{
  "Sid": "CodeStarNotificationsReadWriteAccess",
  "Effect": "Allow",
  "Action": [
    "codestar-notifications:CreateNotificationRule",
    "codestar-notifications:DescribeNotificationRule",
    "codestar-notifications:UpdateNotificationRule",
    "codestar-notifications:Subscribe",
    "codestar-notifications:Unsubscribe"
  ]
},
```

```
"Resource": "*",
"Condition": {
  "StringLike": {
    "codestar-notifications:NotificationsForResource": "arn:aws:codebuild:*"
  }
},
{
  "Sid": "CodeStarNotificationsListAccess",
  "Effect": "Allow",
  "Action": [
    "codestar-notifications:ListNotificationRules",
    "codestar-notifications:ListEventTypes",
    "codestar-notifications:ListTargets",
    "codestar-notifications:ListTagsForResource"
  ],
  "Resource": "*"
},
{
  "Sid": "SNSTopicListAccess",
  "Effect": "Allow",
  "Action": [
    "sns:ListTopics",
    "sns:GetTopicAttributes"
  ],
  "Resource": "*"
},
{
  "Sid": "CodeStarNotificationsChatbotAccess",
  "Effect": "Allow",
  "Action": [
    "chatbot:DescribeSlackChannelConfigurations",
    "chatbot:ListMicrosoftTeamsChannelConfigurations"
  ],
  "Resource": "*"
}
],
"Version": "2012-10-17"
}
```

AWSCodeBuildReadOnlyAccess

La `AWSCodeBuildReadOnlyAccess` política otorga acceso de solo lectura a otros AWS servicios CodeBuild y a los recursos relacionados en ellos. Aplique esta política a los usuarios que pueden consultar y ejecutar compilaciones, consultar proyectos y consultar grupos de informes, pero que no pueden realizar ningún cambio en ellos.

La política `AWSCodeBuildReadOnlyAccess` contiene la siguiente instrucción:

```
{
  "Statement": [
    {
      "Sid": "AWSServicesAccess",
      "Action": [
        "codebuild:BatchGet*",
        "codebuild:GetResourcePolicy",
        "codebuild:List*",
        "codebuild:DescribeTestCases",
        "codebuild:DescribeCodeCoverages",
        "codecommit:GetBranch",
        "codecommit:GetCommit",
        "codecommit:GetRepository",
        "cloudwatch:GetMetricStatistics",
        "events:DescribeRule",
        "events:ListTargetsByRule",
        "events:ListRuleNamesByTarget",
        "logs:GetLogEvents"
      ],
      "Effect": "Allow",
      "Resource": "*"
    },
    {
      "Sid": "CodeStarConnectionsUserAccess",
      "Effect": "Allow",
      "Action": [
        "codestar-connections:ListConnections",
        "codestar-connections:GetConnection"
      ],
      "Resource": [
        "arn:aws:codestar-connections:*:*:connection/*",
        "arn:aws:codeconnections:*:*:connection/*"
      ]
    }
  ],
}
```

```

{
  "Sid": "CodeStarNotificationsPowerUserAccess",
  "Effect": "Allow",
  "Action": [
    "codestar-notifications:DescribeNotificationRule"
  ],
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "codestar-notifications:NotificationsForResource": "arn:aws:codebuild:*"
    }
  }
},
{
  "Sid": "CodeStarNotificationsListAccess",
  "Effect": "Allow",
  "Action": [
    "codestar-notifications:ListNotificationRules",
    "codestar-notifications:ListEventTypes",
    "codestar-notifications:ListTargets"
  ],
  "Resource": "*"
}
],
"Version": "2012-10-17"
}

```

CodeBuild políticas y notificaciones gestionadas

CodeBuild admite las notificaciones, que pueden notificar a los usuarios los cambios importantes en los proyectos de construcción. Las políticas gestionadas CodeBuild incluyen declaraciones de políticas para la funcionalidad de notificación. Para obtener más información, consulte [¿Qué son las notificaciones?](#)

Permisos relacionados con las notificaciones en políticas administradas de acceso total

La política administrada `AWSCodeBuildFullAccess` incluye las siguientes instrucciones para permitir el acceso completo a las notificaciones. Los usuarios con esta política administrada aplicada también pueden crear y administrar temas de Amazon SNS para notificaciones, suscribirse y cancelar la suscripción de los usuarios a los temas, mostrar los temas que se pueden elegir como destinos para las reglas de notificación y mostrar los clientes de AWS Chatbot configurados para Slack.

```

{
  "Sid": "CodeStarNotificationsReadWriteAccess",
  "Effect": "Allow",
  "Action": [
    "codestar-notifications:CreateNotificationRule",
    "codestar-notifications:DescribeNotificationRule",
    "codestar-notifications:UpdateNotificationRule",
    "codestar-notifications>DeleteNotificationRule",
    "codestar-notifications:Subscribe",
    "codestar-notifications:Unsubscribe"
  ],
  "Resource": "*",
  "Condition": {
    "StringLike": {"codestar-notifications:NotificationsForResource" :
"arn:aws:codebuild:*"}
  }
},
{
  "Sid": "CodeStarNotificationsListAccess",
  "Effect": "Allow",
  "Action": [
    "codestar-notifications:ListNotificationRules",
    "codestar-notifications:ListTargets",
    "codestar-notifications:ListTagsForResource",
    "codestar-notifications:ListEventTypes"
  ],
  "Resource": "*"
},
{
  "Sid": "CodeStarNotificationsSNSTopicCreateAccess",
  "Effect": "Allow",
  "Action": [
    "sns:CreateTopic",
    "sns:SetTopicAttributes"
  ],
  "Resource": "arn:aws:sns:*:*:codestar-notifications*"
},
{
  "Sid": "SNSTopicListAccess",
  "Effect": "Allow",
  "Action": [
    "sns:ListTopics"
  ],

```

```

    "Resource": "*"
  },
  {
    "Sid": "CodeStarNotificationsChatbotAccess",
    "Effect": "Allow",
    "Action": [
      "chatbot:DescribeSlackChannelConfigurations",
      "chatbot:ListMicrosoftTeamsChannelConfigurations"
    ],
    "Resource": "*"
  }
}

```

Permisos relacionados con las notificaciones en políticas administradas de solo lectura

La política administrada `AWSCodeBuildReadOnlyAccess` incluye las siguientes instrucciones para permitir el acceso de solo lectura a las notificaciones. Los usuarios con esta política administrada aplicada pueden ver notificaciones de recursos, pero no pueden crearlas, administrarlas ni suscribirse a ellas.

```

{
  "Sid": "CodeStarNotificationsPowerUserAccess",
  "Effect": "Allow",
  "Action": [
    "codestar-notifications:DescribeNotificationRule"
  ],
  "Resource": "*",
  "Condition": {
    "StringLike": {"codestar-notifications:NotificationsForResource" :
"arn:aws:codebuild:*"}
  }
},
{
  "Sid": "CodeStarNotificationsListAccess",
  "Effect": "Allow",
  "Action": [
    "codestar-notifications:ListNotificationRules",
    "codestar-notifications:ListEventTypes",
    "codestar-notifications:ListTargets"
  ],
  "Resource": "*"
}

```

Permisos relacionados con las notificaciones en otras políticas administradas

La política `AWSCodeBuildDeveloperAccess` administrada incluye las siguientes instrucciones para que los usuarios puedan crear notificaciones, editarlas y suscribirse a ellas. Los usuarios no pueden eliminar reglas de notificación ni administrar etiquetas para recursos.

```
{
  "Sid": "CodeStarNotificationsReadWriteAccess",
  "Effect": "Allow",
  "Action": [
    "codestar-notifications:CreateNotificationRule",
    "codestar-notifications:DescribeNotificationRule",
    "codestar-notifications:UpdateNotificationRule",
    "codestar-notifications:Subscribe",
    "codestar-notifications:Unsubscribe"
  ],
  "Resource": "*",
  "Condition": {
    "StringLike": {"codestar-notifications:NotificationsForResource" :
"arn:aws:codebuild*"}
  }
},
{
  "Sid": "CodeStarNotificationsListAccess",
  "Effect": "Allow",
  "Action": [
    "codestar-notifications:ListNotificationRules",
    "codestar-notifications:ListTargets",
    "codestar-notifications:ListTagsForResource",
    "codestar-notifications:ListEventTypes"
  ],
  "Resource": "*"
},
{
  "Sid": "SNSTopicListAccess",
  "Effect": "Allow",
  "Action": [
    "sns:ListTopics"
  ],
  "Resource": "*"
},
{
  "Sid": "CodeStarNotificationsChatbotAccess",
```

```

    "Effect": "Allow",
    "Action": [
        "chatbot:DescribeSlackChannelConfigurations",
        "chatbot:ListMicrosoftTeamsChannelConfigurations"
    ],
    "Resource": "*"
}

```

Para obtener más información acerca de las notificaciones de IAM, consulte [Identity and Access Management para notificaciones de AWS CodeStar](#).

CodeBuild actualizaciones de las políticas AWS gestionadas

Consulte los detalles sobre las actualizaciones de las políticas AWS administradas CodeBuild desde que este servicio comenzó a rastrear estos cambios. Para obtener alertas automáticas sobre cambios en esta página, suscríbese a la fuente RSS en [AWS CodeBuild Historial de documentos de la Guía del usuario](#).

Cambio	Descripción	Fecha
<p><code>AWSCodeBuildAdminAccess</code>, <code>AWSCodeBuildDeveloperAccess</code>, y <code>AWSCodeBuildReadOnlyAccess</code> — Actualización de las políticas existentes</p>	<p>CodeBuild agregó un recurso a estas políticas para respaldar el cambio de AWS CodeConnections marca.</p> <p>El <code>AWSCodeBuildAdminAccess</code>, <code>AWSCodeBuildDeveloperAccess</code>, y <code>AWSCodeBuildReadOnlyAccess</code> las políticas se han modificado para añadir un recurso, <code>arn:aws:codeconnections:*:*:connection/*</code>.</p>	18 de abril de 2024
<p><code>AWSCodeBuildAdminAccess</code> y <code>AWSCodeBuildDeveloperAccess</code></p>	<p>CodeBuild agregó un permiso a estas políticas para admitir</p>	16 de mayo de 2023

Cambio	Descripción	Fecha
: actualización de políticas existentes	<p>el uso de un tipo de notificación adicional AWS Chatbot.</p> <p>Las políticas de <code>AWSCodeBuildAdminAccess</code> y <code>AWSCodeBuildDeveloperAccess</code> se han modificado para añadir un permiso, <code>chatbot:ListMicrosoftTeamsChannelConfigurations</code>.</p>	
CodeBuild comenzó a rastrear los cambios	CodeBuild comenzó a realizar un seguimiento de los cambios de sus políticas AWS gestionadas.	16 de mayo de 2021

Ejemplos de políticas administradas por los clientes

En esta sección, encontrará ejemplos de políticas de usuario que conceden permisos para acciones de AWS CodeBuild. Estas políticas funcionan cuando utilizas la CodeBuild API, los AWS SDK o. AWS CLI Cuando se utiliza la consola, se deben conceder permisos adicionales específicos de la consola. Para obtener más información, consulte [Permisos necesarios para usar la consola de AWS CodeBuild](#).

Puede utilizar los siguientes ejemplos de políticas de IAM para limitar el CodeBuild acceso de sus usuarios y funciones.

Temas

- [Permitir que un usuario obtenga información sobre los proyectos de compilación](#)
- [Permitir que un usuario obtenga información sobre los grupos de informes](#)
- [Permitir que un usuario obtenga información sobre los informes](#)
- [Permitir que un usuario pueda crear proyectos de compilación](#)
- [Permitir que un usuario pueda crear un grupo de informes](#)

- [Permitir que un usuario pueda eliminar un grupo de informes](#)
- [Permitir que un usuario pueda eliminar un informe](#)
- [Permitir que un usuario pueda eliminar proyectos de compilación](#)
- [Permitir que un usuario pueda obtener una lista con los nombres de los proyectos de compilación](#)
- [Permitir que un usuario pueda cambiar la información de los proyectos de compilación](#)
- [Permitir que un usuario pueda cambiar un grupo de informes](#)
- [Permitir que un usuario obtenga información sobre las compilaciones](#)
- [Permitir un usuario pueda obtener una lista con los identificadores de compilación de un proyecto de compilación](#)
- [Permitir que un usuario pueda obtener una lista de identificadores de compilación](#)
- [Permitir que a un usuario pueda obtener una lista de grupos de informes](#)
- [Permitir que un usuario pueda obtener una lista de informes](#)
- [Permitir que un usuario pueda obtener una lista con los informes de un grupo de informes](#)
- [Permitir que un usuario pueda obtener una lista con los casos de prueba de un informe](#)
- [Permitir que un usuario pueda comenzar a ejecutar compilaciones](#)
- [Permitir que un usuario pueda intentar detener las compilaciones](#)
- [Permitir que un usuario pueda intentar eliminar compilaciones](#)
- [Permite a un usuario obtener información sobre las imágenes de Docker administradas por CodeBuild](#)
- [Permitir el CodeBuild acceso a AWS los servicios necesarios para crear una interfaz de red de VPC](#)
- [Utilice una declaración de denegación para evitar la desconexión AWS CodeBuild de los proveedores de origen](#)

Permitir que un usuario obtenga información sobre los proyectos de compilación

A continuación se muestra una instrucción de política de ejemplo que permite a un usuario obtener información acerca de los proyectos de compilación en la región us-east-2 de la cuenta 123456789012 para cualquier proyecto de compilación que empiece con el nombre my:

```
{
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Action": "codebuild:BatchGetProjects",
    "Resource": "arn:aws:codebuild:us-east-2:123456789012:project/my*"
  }
]
```

Permitir que un usuario obtenga información sobre los grupos de informes

El ejemplo siguiente de instrucción de política permite a un usuario obtener información sobre grupos de informes en la región `us-east-2` para la cuenta `123456789012`:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:BatchGetReportGroups",
      "Resource": "arn:aws:codebuild:us-east-2:123456789012:report-group/*"
    }
  ]
}
```

Permitir que un usuario obtenga información sobre los informes

El ejemplo siguiente de instrucción de política permite a un usuario obtener información sobre informes en la región `us-east-2` para la cuenta `123456789012`:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:BatchGetReports",
      "Resource": "arn:aws:codebuild:us-east-2:123456789012:report-group/*"
    }
  ]
}
```

Permitir que un usuario pueda crear proyectos de compilación

El siguiente ejemplo de declaración de política permite a un usuario crear proyectos de construcción con cualquier nombre, pero solo en la us-east-2 región como cuenta 123456789012 y únicamente utilizando la función de CodeBuild servicio especificada:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:CreateProject",
      "Resource": "arn:aws:codebuild:us-east-2:123456789012:project/*"
    },
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::123456789012:role/CodeBuildServiceRole"
    }
  ]
}
```

El siguiente ejemplo de declaración de política permite a un usuario crear proyectos de construcción con cualquier nombre, pero solo en la us-east-2 región como cuenta 123456789012 y únicamente con el rol de CodeBuild servicio especificado. También establece que el usuario solo puede usar el rol de servicio especificado con AWS CodeBuild ningún otro AWS servicio.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:CreateProject",
      "Resource": "arn:aws:codebuild:us-east-2:123456789012:project/*"
    },
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::123456789012:role/CodeBuildServiceRole",
      "Condition": {
        "StringEquals": {"iam:PassedToService": "codebuild.amazonaws.com"}
      }
    }
  ]
}
```

```
    }  
  ]  
}}
```

Permitir que un usuario pueda crear un grupo de informes

El ejemplo siguiente de la instrucción de una política permite a un usuario crear un grupo de informes en la región `us-east-2` para la cuenta `123456789012`:

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": "codebuild:CreateReportGroup",  
      "Resource": "arn:aws:codebuild:us-east-2:123456789012:report-group/*"  
    }  
  ]  
}
```

Permitir que un usuario pueda eliminar un grupo de informes

El ejemplo siguiente de instrucción de política permite a un usuario eliminar un grupo de informes en la región `us-east-2` para la cuenta `123456789012`:

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": "codebuild>DeleteReportGroup",  
      "Resource": "arn:aws:codebuild:us-east-2:123456789012:report-group/*"  
    }  
  ]  
}
```

Permitir que un usuario pueda eliminar un informe

El ejemplo siguiente de instrucción de política permite a un usuario eliminar un informe en la región `us-east-2` para la cuenta `123456789012`:

```
{
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Action": "codebuild:DeleteReport",
    "Resource": "arn:aws:codebuild:us-east-2:123456789012:report-group/*"
  }
]
```

Permitir que un usuario pueda eliminar proyectos de compilación

A continuación se muestra una instrucción de política de ejemplo que permite a un usuario eliminar proyectos de compilación en la región us-east-2 de la cuenta 123456789012 para cualquier proyecto de compilación que empiece con el nombre my:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:DeleteProject",
      "Resource": "arn:aws:codebuild:us-east-2:123456789012:project/my*"
    }
  ]
}
```

Permitir que un usuario pueda obtener una lista con los nombres de los proyectos de compilación

La siguiente instrucción de política de ejemplo permite a un usuario obtener una lista de nombres de proyectos de compilación de la misma cuenta:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:ListProjects",
      "Resource": "*"
    }
  ]
}
```

Permitir que un usuario pueda cambiar la información de los proyectos de compilación

La siguiente instrucción de política de ejemplo permite a un usuario cambiar la información de los proyectos de compilación con cualquier nombre, pero solo en la región us-east-2 de la cuenta 123456789012 y solo mediante el rol de servicio de AWS CodeBuild especificado:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:UpdateProject",
      "Resource": "arn:aws:codebuild:us-east-2:123456789012:project/*"
    },
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::123456789012:role/CodeBuildServiceRole"
    }
  ]
}
```

Permitir que un usuario pueda cambiar un grupo de informes

El ejemplo siguiente de instrucción de política permite a un usuario cambiar un grupo de informes en la región us-east-2 para la cuenta 123456789012:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:UpdateReportGroup",
      "Resource": "arn:aws:codebuild:us-east-2:123456789012:report-group/*"
    }
  ]
}
```

Permitir que un usuario obtenga información sobre las compilaciones

La siguiente instrucción de política de ejemplo permite a un usuario obtener información acerca de las compilaciones en la región `us-east-2` de la cuenta `123456789012` para los proyectos de compilación denominados `my-build-project` y `my-other-build-project`:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:BatchGetBuilds",
      "Resource": [
        "arn:aws:codebuild:us-east-2:123456789012:project/my-build-project",
        "arn:aws:codebuild:us-east-2:123456789012:project/my-other-build-project"
      ]
    }
  ]
}
```

Permitir un usuario pueda obtener una lista con los identificadores de compilación de un proyecto de compilación

La siguiente instrucción de política de ejemplo permite a un usuario obtener una lista de identificadores de compilación en la región `us-east-2` de la cuenta `123456789012` para los proyectos de compilación denominados `my-build-project` y `my-other-build-project`:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:ListBuildsForProject",
      "Resource": [
        "arn:aws:codebuild:us-east-2:123456789012:project/my-build-project",
        "arn:aws:codebuild:us-east-2:123456789012:project/my-other-build-project"
      ]
    }
  ]
}
```

Permitir que un usuario pueda obtener una lista de identificadores de compilación

La siguiente instrucción de política de ejemplo permite a un usuario obtener una lista de todos los identificadores de compilación de la misma cuenta:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:ListBuilds",
      "Resource": "*"
    }
  ]
}
```

Permitir que a un usuario pueda obtener una lista de grupos de informes

El ejemplo siguiente de instrucción de política permite a un usuario obtener una lista de los grupos de informes en la región us-east-2 para la cuenta 123456789012:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:ListReportGroups",
      "Resource": "*"
    }
  ]
}
```

Permitir que un usuario pueda obtener una lista de informes

El ejemplo siguiente de instrucción de política permite a un usuario obtener una lista de los informes en la región us-east-2 para la cuenta 123456789012:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```
    "Effect": "Allow",
    "Action": "codebuild:ListReports",
    "Resource": "*"
  }
]
```

Permitir que un usuario pueda obtener una lista con los informes de un grupo de informes

El ejemplo siguiente de instrucción de política permite a un usuario obtener una lista de informes de un grupos de informes en la región us-east-2 para la cuenta 123456789012:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:ListReportsForReportGroup",
      "Resource": "arn:aws:codebuild:us-east-2:123456789012:report-group/*"
    }
  ]
}
```

Permitir que un usuario pueda obtener una lista con los casos de prueba de un informe

La siguiente instrucción de política de ejemplo permite a un usuario obtener una lista de casos de prueba para un informe en la región us-east-2 para la cuenta 123456789012:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:DescribeTestCases",
      "Resource": "arn:aws:codebuild:us-east-2:123456789012:report-group/*"
    }
  ]
}
```

Permitir que un usuario pueda comenzar a ejecutar compilaciones

La siguiente instrucción de política de ejemplo permite a un usuario ejecutar compilaciones en la región us-east-2 de la cuenta 123456789012 para proyectos de compilación que empiezan con el nombre my:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:StartBuild",
      "Resource": "arn:aws:codebuild:us-east-2:123456789012:project/my*"
    }
  ]
}
```

Permitir que un usuario pueda intentar detener las compilaciones

A continuación se muestra una instrucción de política de ejemplo que permite a un usuario intentar detener la ejecución de compilaciones únicamente en la región us-east-2 de la cuenta 123456789012 para cualquier proyecto de compilación que empiece con el nombre my:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:StopBuild",
      "Resource": "arn:aws:codebuild:us-east-2:123456789012:project/my*"
    }
  ]
}
```

Permitir que un usuario pueda intentar eliminar compilaciones

La siguiente instrucción de la política de ejemplo permite que un usuario pueda intentar eliminar compilaciones únicamente en la región us-east-2 de la cuenta 123456789012 para cualquier proyecto de compilación que empiece con el nombre my:

```
{
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Action": "codebuild:BatchDeleteBuilds",
    "Resource": "arn:aws:codebuild:us-east-2:123456789012:project/my*"
  }
]
```

Permite a un usuario obtener información sobre las imágenes de Docker administradas por CodeBuild

El siguiente ejemplo de declaración de política permite al usuario obtener información sobre todas las imágenes de Docker que administra: CodeBuild

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:ListCuratedEnvironmentImages",
      "Resource": "*"
    }
  ]
}
```

Permitir el CodeBuild acceso a AWS los servicios necesarios para crear una interfaz de red de VPC

El siguiente ejemplo de declaración de política concede AWS CodeBuild permiso para crear una interfaz de red en una VPC con dos subredes:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateNetworkInterface",
        "ec2:DescribeDhcpOptions",
        "ec2:DescribeNetworkInterfaces",
        "ec2>DeleteNetworkInterface",

```

```

    "ec2:DescribeSubnets",
    "ec2:DescribeSecurityGroups",
    "ec2:DescribeVpcs"
  ],
  "Resource": "*"
},
{
  "Effect": "Allow",
  "Action": [
    "ec2:CreateNetworkInterfacePermission"
  ],
  "Resource": "arn:aws:ec2:region:account-id:network-interface/*",
  "Condition": {
    "StringEquals": {
      "ec2:AuthorizedService": "codebuild.amazonaws.com"
    },
    "ArnEquals": {
      "ec2:Subnet": [
        "arn:aws:ec2:region:account-id:subnet/subnet-id-1",
        "arn:aws:ec2:region:account-id:subnet/subnet-id-2"
      ]
    }
  }
}
]
}

```

Utilice una declaración de denegación para evitar la desconexión AWS CodeBuild de los proveedores de origen

La siguiente instrucción de política de ejemplo utiliza una instrucción de denegación para evitar que AWS CodeBuild se desconecte de los proveedores de código fuente. Utiliza `codebuild:DeleteAuthToken`, que es lo contrario de `codebuild:PersistAuthToken` y `codebuild:ImportSourceCredentials`, para establecer conexión con los proveedores de código fuente. Para obtener más información, consulte [Permisos necesarios para que la AWS CodeBuild consola se conecte a los proveedores de origen](#).

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",

```

```
    "Action": "codebuild:DeleteOAuthToken",
    "Resource": "*"
  }
]
```

AWS CodeBuild referencia de permisos

Puedes usar claves AWS de condición generales en tus AWS CodeBuild políticas para expresar las condiciones. Para ver una lista, consulte [Claves disponibles](#) en la Guía de usuario de IAM.

Las acciones se especifican en el campo `Action` de la política. Para especificar una acción, use el prefijo `codebuild:` seguido del nombre de operación de API (por ejemplo, `codebuild:CreateProject` y `codebuild:StartBuild`). Para especificar varias acciones en una única instrucción, sepárelas con comas (por ejemplo, `"Action": ["codebuild:CreateProject", "codebuild:StartBuild"]`).

Uso de caracteres comodín

Debe especificar un ARN, con o sin un carácter comodín (*), como el valor del recurso en el campo `Resource` de la política. Puede utilizar un carácter comodín para especificar varias acciones o recursos. Por ejemplo, `codebuild:*` especifica todas CodeBuild las acciones y `codebuild:Batch*` especifica todas CodeBuild las acciones que comienzan por la palabra `Batch`. El siguiente ejemplo otorga acceso a todos los proyectos de compilación con nombres que comienzan por `my`:

```
arn:aws:codebuild:us-east-2:123456789012:project/my*
```

CodeBuild Operaciones de la API y permisos necesarios para las acciones

BatchDeleteBuilds

Acción: `codebuild:BatchDeleteBuilds`

Necesario para eliminar compilaciones.

Recurso: `arn:aws:codebuild:region-ID:account-ID:project/project-name`

BatchGetBuilds

Acción: `codebuild:BatchGetBuilds`

Necesario para obtener información acerca de las compilaciones.

Recurso: `arn:aws:codebuild:region-ID:account-ID:project/project-name`

BatchGetProjects

Acción: `codebuild:BatchGetProjects`

Necesario para obtener información acerca de los proyectos de compilación.

Recurso: `arn:aws:codebuild:region-ID:account-ID:project/project-name`

BatchGetReportGroups

Acción: `codebuild:BatchGetReportGroups`

Necesario para obtener información sobre grupos de informes.

Recurso: `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

BatchGetReports

Acción: `codebuild:BatchGetReports`

Necesario para obtener información sobre informes.

Recurso: `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

BatchPutTestCases¹

Acción: `codebuild:BatchPutTestCases`

Necesario para crear o actualizar un informe de prueba.

Recurso: `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

CreateProject

Acciones: `codebuild:CreateProject`, `iam:PassRole`

Necesario para crear proyectos de compilación.

Recursos:

- `arn:aws:codebuild:region-ID:account-ID:project/project-name`

- `arn:aws:iam::account-ID:role/role-name`

CreateReport¹

Acción: `codebuild:CreateReport`

Necesario para crear un informe de prueba.

Recurso: `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

CreateReportGroup

Acción: `codebuild:CreateReportGroup`

Necesario para crear un grupo de informes.

Recurso: `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

CreateWebhook

Acción: `codebuild:CreateWebhook`

Se necesita para crear un webhook.

Recurso: `arn:aws:codebuild:region-ID:account-ID:project/project-name`

DeleteProject

Acción: `codebuild>DeleteProject`

Necesario para eliminar un CodeBuild proyecto.

Recurso: `arn:aws:codebuild:region-ID:account-ID:project/project-name`

DeleteReport

Acción: `codebuild>DeleteReport`

Necesario para eliminar un informe.

Recurso: `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

DeleteReportGroup

Acción: `codebuild>DeleteReportGroup`

Necesario para eliminar un grupo de informes.

Recurso: `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

DeleteSourceCredentials

Acción: `codebuild>DeleteSourceCredentials`

Necesario para eliminar un conjunto de `SourceCredentialsInfo` objetos que contienen información sobre las credenciales de un GitHub repositorio de GitHub Enterprise Server o Bitbucket.

Recurso: *

DeleteWebhook

Acción: `codebuild>DeleteWebhook`

Se necesita para crear un webhook.

Recurso: `arn:aws:codebuild:region-ID:account-ID:project/project-name`

DescribeTestCases

Acción: `codebuild:DescribeTestCases`

Necesario para devolver una lista paginada de casos de prueba.

Recurso: `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

ImportSourceCredentials

Acción: `codebuild:ImportSourceCredentials`

Necesario para importar un conjunto de `SourceCredentialsInfo` objetos que contienen información sobre las credenciales de un GitHub repositorio de GitHub Enterprise Server o Bitbucket.

Recurso: *

InvalidateProjectCache

Acción: `codebuild:InvalidateProjectCache`

Necesario para restablecer la caché de un proyecto.

Recurso: `arn:aws:codebuild:region-ID:account-ID:project/project-name`

ListBuildBatches

Acción: `codebuild:ListBuildBatches`

Necesario para obtener una lista de identificadores de lotes para compilación.

Recurso: *

ListBuildBatchesForProject

Acción: `codebuild:ListBuildBatchesForProject`

Necesario para obtener una lista de identificadores de lotes para compilación de un proyecto específico.

Recurso: `arn:aws:codebuild:region-ID:account-ID:project/project-name`

ListBuilds

Acción: `codebuild:ListBuilds`

Necesario para obtener una lista de identificadores de compilación.

Recurso: *

ListBuildsForProject

Acción: `codebuild:ListBuildsForProject`

Necesario para obtener una lista de identificadores de compilaciones de un proyecto de compilación.

Recurso: `arn:aws:codebuild:region-ID:account-ID:project/project-name`

ListCuratedEnvironmentImages

Acción: `codebuild:ListCuratedEnvironmentImages`

Necesario para obtener información sobre todas las imágenes de Docker administradas por AWS CodeBuild.

Recurso: * (es necesario, pero no hace referencia a un recurso de AWS atribuible)

ListProjects

Acción: `codebuild:ListProjects`

Necesario para obtener una lista de nombres de proyectos de compilación.

Recurso: *

ListReportGroups

Acción: `codebuild:ListReportGroups`

Necesario para obtener una lista de grupos de informes.

Recurso: *

ListReports

Acción: `codebuild:ListReports`

Necesario para obtener una lista de informes.

Recurso: *

ListReportsForReportGroup

Acción: `codebuild:ListReportsForReportGroup`

Necesario para obtener una lista de informes de un grupo de informes.

Recurso: `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

RetryBuild

Acción: `codebuild:RetryBuild`

Required to retry builds.

Recurso: `arn:aws:codebuild:region-ID:account-ID:project/project-name`

StartBuild

Acción: `codebuild:StartBuild`

Necesario para empezar a ejecutar compilaciones.

Recurso: `arn:aws:codebuild:region-ID:account-ID:project/project-name`

StopBuild

Acción: `codebuild:StopBuild`

Necesario para intentar detener la ejecución de compilaciones.

Recurso: `arn:aws:codebuild:region-ID:account-ID:project/project-name`

UpdateProject

Acciones: `codebuild:UpdateProject`, `iam:PassRole`

Necesario para cambiar la información sobre las compilaciones.

Recursos:

- `arn:aws:codebuild:region-ID:account-ID:project/project-name`
- `arn:aws:iam::account-ID:role/role-name`

UpdateProjectVisibility

Acciones: `codebuild:UpdateProjectVisibility`, `iam:PassRole`

Necesario para cambiar la visibilidad pública de las compilaciones de un proyecto.

Recursos:

- `arn:aws:codebuild:region-ID:account-ID:project/project-name`
- `arn:aws:iam::account-ID:role/role-name`

UpdateReport¹

Acción: `codebuild:UpdateReport`

Necesario para crear o actualizar un informe de prueba.

Recurso: `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

UpdateReportGroup

Acción: `codebuild:UpdateReportGroup`

Necesario para actualizar un grupo de informes.

Recurso: `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

UpdateWebhook

Acción: `codebuild:UpdateWebhook`

Se necesita para actualizar un webhook.

Recurso: `arn:aws:codebuild:region-ID:account-ID:project/project-name`

¹ Se utiliza solo para permisos. No hay API para esta acción.

Uso de etiquetas para controlar el acceso a los recursos de AWS CodeBuild

Las condiciones de las declaraciones de política de IAM forman parte de la sintaxis que se puede utilizar para especificar los permisos para las acciones CodeBuild basadas en proyectos. Puede crear una política que permita o deniegue acciones sobre los proyectos en función de las etiquetas asociadas a dichos proyectos y aplicar después esas políticas a los grupos de IAM que configure para administrar usuarios. Para obtener información sobre cómo aplicar etiquetas a un proyecto mediante la consola o consulte AWS CLI. [Creación de un proyecto de compilación en AWS CodeBuild](#) Para obtener información sobre cómo aplicar etiquetas mediante el CodeBuild SDK, consulte [CreateProject](#) la referencia sobre las [etiquetas](#) en la CodeBuild API. Para obtener información sobre el uso de etiquetas para controlar el acceso a AWS los recursos, consulte [Control del acceso a AWS los recursos mediante etiquetas de recursos](#) en la Guía del usuario de IAM.

Example Ejemplo 1: Limite las acciones CodeBuild del proyecto en función de las etiquetas de los recursos

En el ejemplo siguiente, se deniegan todas las acciones BatchGetProjects en los proyectos etiquetados con la clave Environment y el valor de clave Production. El administrador de un usuario debe asociar esta política de IAM, junto con la política de usuario administrada, a los usuarios de que no estén autorizados. La clave de condición `aws:ResourceTag` se utiliza para controlar el acceso a los recursos en función de sus etiquetas.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "codebuild:BatchGetProjects"
      ],
      "Resource": "*",
      "Condition": {
        "ForAnyValue:StringEquals": {
```

```

        "aws:ResourceTag/Environment": "Production"
    }
}
]
}

```

Example Ejemplo 2: Limitar las acciones CodeBuild del proyecto en función de las etiquetas de solicitud

La siguiente política deniega a los usuarios el permiso para la acción `CreateProject` si la solicitud contiene una etiqueta con la clave `Environment` y el valor de clave `Production`. Además, la política impide que estos usuarios no autorizados puedan modificar los proyectos utilizando la clave de condición `aws:TagKeys` y no permite `UpdateProject` si la solicitud contiene una etiqueta con la clave `Environment`. Un administrador debe asociar esta política de IAM, junto con la política de usuario administrada, a los usuarios que no estén autorizados para realizar estas acciones. (La clave de condición `aws:RequestTag` se utiliza para controlar qué etiquetas se pueden pasar en una solicitud de IAM).

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "codebuild:CreateProject"
      ],
      "Resource": "*",
      "Condition": {
        "ForAnyValue:StringEquals": {
          "aws:RequestTag/Environment": "Production"
        }
      }
    },
    {
      "Effect": "Deny",
      "Action": [
        "codebuild:UpdateProject"
      ],
      "Resource": "*",
      "Condition": {
        "ForAnyValue:StringEquals": {

```

```

        "aws:TagKeys": ["Environment"]
    }
}
]
}

```

Example Ejemplo 3: denegar o permitir acciones en grupos de informes en función de etiquetas de recursos

Puede crear una política que permita o deniegue acciones en CodeBuild los recursos (proyectos y grupos de informes) en función de las AWS etiquetas asociadas a esos recursos y, a continuación, aplicar esas políticas a los grupos de IAM que configure para administrar los usuarios. Por ejemplo, puede crear una política que deniegue todas CodeBuild las acciones en cualquier grupo de informes con la clave de AWS etiqueta Status y el valor clave de ySecret, a continuación, aplicar esa política al grupo de IAM que creó para desarrolladores generales (*desarrolladores*). Tendrá que asegurarse de que los desarrolladores que trabajan en esos grupos de informes etiquetados no sean miembros de ese grupo de *Desarrolladores* general, pero pertenezcan en su lugar a un grupo de IAM distinto que no tiene la política restrictiva aplicada (SecretDevelopers).

El siguiente ejemplo deniega todas CodeBuild las acciones en los grupos de informes etiquetados con la clave Status y el valor clave deSecret:

```

{
  "Version": "2012-10-17",
  "Statement" : [
    {
      "Effect" : "Deny",
      "Action" : [
        "codebuild:BatchGetReportGroups",
        "codebuild:CreateReportGroup",
        "codebuild>DeleteReportGroup",
        "codebuild:ListReportGroups",
        "codebuild:ListReportsForReportGroup",
        "codebuild:UpdateReportGroup"
      ]
      "Resource" : "*",
      "Condition" : {
        "StringEquals" : "aws:ResourceTag/Status": "Secret"
      }
    }
  ]
}

```

```
]
}
```

Example Ejemplo 4: Limitar CodeBuild las acciones AWSCodeBuildDeveloperAccess a las etiquetas de recursos

Puede crear políticas que permitan CodeBuild realizar acciones en todos los grupos de informes y proyectos que no estén etiquetados con etiquetas específicas. Por ejemplo, la siguiente política permite el equivalente de permisos [AWSCodeBuildDeveloperAccess](#) para todos los grupos de informes y proyectos excepto los etiquetados con las etiquetas especificadas:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "codebuild:StartBuild",
        "codebuild:StopBuild",
        "codebuild:BatchGet*",
        "codebuild:GetResourcePolicy",
        "codebuild:DescribeTestCases",
        "codebuild:List*",
        "codecommit:GetBranch",
        "codecommit:GetCommit",
        "codecommit:GetRepository",
        "codecommit:ListBranches",
        "cloudwatch:GetMetricStatistics",
        "events:DescribeRule",
        "events:ListTargetsByRule",
        "events:ListRuleNamesByTarget",
        "logs:GetLogEvents",
        "s3:GetBucketLocation",
        "s3:ListAllMyBuckets"
      ],
      "Resource": "*",
      "Condition": {
        "StringNotEquals": {
          "aws:ResourceTag/Status": "Secret",
          "aws:ResourceTag/Team": "Saanvi"
        }
      }
    }
  ]
}
```

```
]
}
```

Visualización de recursos en la consola

La AWS CodeBuild consola requiere el `ListRepositories` permiso para mostrar una lista de los repositorios de tu AWS cuenta en la AWS región en la que has iniciado sesión. La consola también incluye una función `Go to resource` (Ir al recurso) para realizar una búsqueda rápida de recursos sin distinción entre mayúsculas y minúsculas. Esta búsqueda se realiza en tu AWS cuenta de la AWS región en la que has iniciado sesión. Los siguientes recursos se muestran en los siguientes servicios:

- AWS CodeBuild: proyectos de compilación
- AWS CodeCommit: repositorios
- AWS CodeDeploy: aplicaciones
- AWS CodePipeline: canalizaciones

Para realizar esta búsqueda en los recursos de todos los servicios, debe contar con los siguientes permisos:

- CodeBuild: `ListProjects`
- CodeCommit: `ListRepositories`
- CodeDeploy: `ListApplications`
- CodePipeline: `ListPipelines`

Los resultados de los recursos de un servicio no se devuelven si no tiene permisos para ese servicio. Aunque tenga permisos para ver los recursos, algunos recursos no se devolverán si hay un permiso `Deny` explícito para ver esos recursos.

Validación de conformidad para AWS CodeBuild

Los auditores externos evalúan la seguridad y el cumplimiento AWS CodeBuild como parte de varios programas de AWS cumplimiento. Estos incluyen SOC, PCI, FedRAMP, HIPAA y otros.

Para obtener una lista de AWS los servicios incluidos en el ámbito de los programas de cumplimiento específicos, consulte [AWS los servicios incluidos en el ámbito de aplicación por programa de cumplimiento](#). Para obtener información general, consulte [Programas de conformidad de AWS](#).

Puede descargar informes de auditoría de terceros utilizando AWS Artifact. Para obtener más información, consulta [Descarga de informes en AWS Artifact](#).

Su responsabilidad de cumplimiento al CodeBuild utilizarlos viene determinada por la confidencialidad de sus datos, los objetivos de cumplimiento de su empresa y las leyes y reglamentos aplicables. Si su uso de CodeBuild está sujeto al cumplimiento de normas como HIPAA, PCI o FedRAMP, proporciona recursos que le ayudarán a: AWS

- Guías de [inicio rápido sobre seguridad y cumplimiento: estas guías](#) de implementación analizan las consideraciones arquitectónicas y proporcionan los pasos necesarios para implementar entornos básicos centrados en la seguridad y el cumplimiento. AWS
- Documento técnico sobre [cómo diseñar una arquitectura para la seguridad y el cumplimiento de la HIPAA: este documento técnico describe cómo las](#) empresas pueden utilizar para crear aplicaciones que cumplan con la HIPAA. AWS
- [AWS Recursos de cumplimiento](#): esta colección de libros de trabajo y guías puede aplicarse a su sector y ubicación.
- [AWS Config](#)— Este AWS servicio evalúa en qué medida las configuraciones de sus recursos cumplen con las prácticas internas, las directrices del sector y las normativas.
- [AWS Security Hub](#)— Supervise su uso en lo que AWS CodeBuild respecta a las mejores prácticas de seguridad mediante el uso [AWS Security Hub](#)de. Security Hub utiliza controles de seguridad para evaluar las configuraciones de los recursos y los estándares de seguridad para ayudarle a cumplir varios marcos de conformidad. Para obtener más información sobre el uso de Security Hub para evaluar CodeBuild los recursos, consulte [AWS CodeBuild los controles](#) en la Guía del AWS Security Hub usuario.

Resiliencia en AWS CodeBuild

La infraestructura AWS global se basa en AWS regiones y zonas de disponibilidad. AWS Las regiones proporcionan varias zonas de disponibilidad aisladas y separadas físicamente, que están conectadas mediante redes de baja latencia, alto rendimiento y alta redundancia. Con las zonas de disponibilidad, puede diseñar y utilizar aplicaciones y bases de datos que realizan una conmutación por error automática entre zonas de disponibilidad sin interrupciones. Las zonas de disponibilidad tienen una mayor disponibilidad, tolerancia a errores y escalabilidad que las infraestructuras tradicionales de centros de datos únicos o múltiples.

[Para obtener más información sobre AWS las regiones y las zonas de disponibilidad, consulte la infraestructura global.AWS](#)

Seguridad de la infraestructura en AWS CodeBuild

Como servicio gestionado, AWS CodeBuild está protegido por la seguridad de la red AWS global. Para obtener información sobre los servicios AWS de seguridad y cómo se AWS protege la infraestructura, consulte [Seguridad AWS en la nube](#). Para diseñar su AWS entorno utilizando las mejores prácticas de seguridad de la infraestructura, consulte [Protección de infraestructuras en un marco](#) de buena AWS arquitectura basado en el pilar de la seguridad.

Utiliza las llamadas a la API AWS publicadas para acceder a CodeBuild través de la red. Los clientes deben admitir lo siguiente:

- Seguridad de la capa de transporte (TLS). Exigimos TLS 1.2 y recomendamos TLS 1.3.
- Conjuntos de cifrado con confidencialidad directa total (PFS) como DHE (Ephemeral Diffie-Hellman) o ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). La mayoría de los sistemas modernos como Java 7 y posteriores son compatibles con estos modos.

Además, las solicitudes deben estar firmadas mediante un ID de clave de acceso y una clave de acceso secreta que esté asociada a una entidad de seguridad de IAM principal. También puede utilizar [AWS Security Token Service](#) (AWS STS) para generar credenciales de seguridad temporales para firmar solicitudes.

Acceda a su proveedor de origen en CodeBuild

En el GitHub caso de GitHub Enterprise Server, utilizas un token de acceso personal o una aplicación OAuth para acceder al proveedor de origen. En el caso de Bitbucket, utilizas un token de acceso, una contraseña de aplicación o una aplicación OAuth para acceder al proveedor de origen.

Note

GitLab y a los proveedores de GitLab fuentes autogestionados no se accede directamente, CodeBuild sino a través de ellos. AWS CodeConnections

Temas

- [GitHub y un token de acceso a GitHub Enterprise Server](#)
- [GitHub Aplicación OAuth](#)

- [Contraseña o token de acceso de la aplicación Bitbucket](#)
- [Aplicación OAuth de Bitbucket](#)

GitHub y un token de acceso a GitHub Enterprise Server

Requisitos previos de los tokens de acceso

Antes de empezar, debe añadir los ámbitos de permiso adecuados a su token de GitHub acceso.

Pues GitHub, su token de acceso personal debe tener los siguientes ámbitos.

- `repo`: Concede a control total de repositorios privados.
- `repo:status`: otorga acceso de lectura y escritura a los estados de confirmación de los repositorios públicos y privados.
- `admin:repo_hook`: Concede control total de enlaces de repositorio. Este ámbito no es necesario si su token tiene el ámbito `repo`.

Para obtener más información, consulta Cómo [entender los ámbitos de las aplicaciones de OAuth](#) en el sitio web. GitHub

Si utilizas tokens de acceso personal específicos, según el caso de uso, es posible que tu token de acceso personal necesite los siguientes permisos:

- `Contenido: de solo lectura`: otorga acceso a repositorios privados. Este permiso es obligatorio si utilizas repositorios privados como fuente.
- `Estados de confirmación: lectura y escritura`: otorga permiso para crear estados de confirmación. Este permiso es obligatorio si tu proyecto tiene un webhook configurado o si tienes habilitada la función de estado de creación de informes.
- `Webhooks: lectura y escritura`: otorga permiso para gestionar los webhooks. Este permiso es obligatorio si tu proyecto tiene un webhook configurado.
- `Solicitudes de extracción: de solo lectura`: otorga permiso para acceder a las solicitudes de extracción. Este permiso es obligatorio si tu webhook tiene un `FILE_PATH` filtro para los eventos de solicitudes de atracción.
- `Administración: lectura y escritura`: este permiso es necesario si utilizas la función GitHub Actions Runner autohospedada con. CodeBuild Para obtener más información, consulta [Crear un token](#)

[de registro para un repositorio](#) y [Configura ejecutores de GitHub Actions autohospedados en AWS CodeBuild](#).

Note

Si quieres acceder a los repositorios de la organización, asegúrate de especificar la organización como propietaria del recurso del token de acceso.

Para obtener más información, consulta el artículo sobre [los permisos necesarios para los tokens de acceso personal detallados](#) en el sitio web. GitHub

Conéctese GitHub con un token de acceso (consola)

Para usar la consola para conectar tu proyecto GitHub mediante un token de acceso, haz lo siguiente al crear un proyecto. Para obtener más información, consulte [Creación de un proyecto de compilación \(consola\)](#).

1. En Proveedor de código fuente, selecciona GitHub.
2. En Repositorio, selecciona Conectar con un token de acceso GitHub personal.
3. En el token de acceso GitHub personal, introduce tu token de acceso GitHub personal.
4. Seleccione Save token (Guardar token).

Conéctese GitHub con un token de acceso (CLI)

Siga estos pasos para usar el y AWS CLI conectar su proyecto GitHub mediante un token de acceso. Para obtener información sobre el uso del AWS CLI with AWS CodeBuild, consulta la [Referencia de la línea de comandos](#).

1. Ejecute el comando import-source-credentials:

```
aws codebuild import-source-credentials --generate-cli-skeleton
```

En el resultado se muestran datos con formato JSON. Copie los datos en un archivo (por ejemplo, *import-source-credentials.json*) en una ubicación del equipo o instancia local en la que AWS CLI esté instalado. Modifique los datos copiados como se indica a continuación y guarde los resultados.

```
{
  "serverType": "server-type",
  "authType": "auth-type",
  "shouldOverwrite": "should-overwrite",
  "token": "token",
  "username": "username"
}
```

Sustituya lo siguiente:

- *tipo-servidor*: valor obligatorio. El proveedor de código fuente utilizado para esta credencial. Los valores válidos son GITHUB o GITHUB_ENTERPRISE.
 - *tipo-autenticación*: valor obligatorio. El tipo de autenticación que se utiliza para conectarse a un repositorio de GitHub Enterprise Server GitHub o a uno de ellos. Los valores válidos incluyen PERSONAL_ACCESS_TOKEN y BASIC_AUTH. No puede usar la CodeBuild API para crear una conexión OAUTH. En su lugar, debes usar la CodeBuild consola.
 - *should-overwrite*: valor opcional. Establézcalo en `false` para impedir que se sobrescriban las credenciales de origen del repositorio. Establézcalo en `true` para sobrescribir las credenciales de origen del repositorio. El valor predeterminado es `true`.
 - *token*: valor obligatorio. En el GitHub caso de GitHub Enterprise Server, este es el token de acceso personal.
 - *nombre de usuario*: valor opcional. Este parámetro se omite para GitHub los proveedores fuente de GitHub Enterprise Server.
2. Para conectar su cuenta con un token de acceso, cambie al directorio que contiene el archivo `import-source-credentials.json` que guardó en el paso 1 y vuelva a ejecutar el comando `import-source-credentials`.

```
aws codebuild import-source-credentials --cli-input-json file://import-source-credentials.json
```

Los datos con formato JSON aparecen en el resultado con un nombre de recurso de Amazon (ARN).

```
{
  "arn": "arn:aws:codebuild:region:account-id:token/server-type"
}
```

Note

Si ejecuta el comando `import-source-credentials` con el mismo tipo de servidor y tipo de autorización una segunda vez, se actualiza el token de acceso almacenado.

Una vez que tu cuenta esté conectada con un token de acceso, podrás usarlo `create-project` para crear tu CodeBuild proyecto. Para obtener más información, consulte [Crear un proyecto de compilación \(AWS CLI\)](#).

3. Para ver los tokens de acceso conectados, ejecute el comando `list-source-credentials`.

```
aws codebuild list-source-credentials
```

Un objeto `sourceCredentialsInfos` con formato JSON aparece en el resultado:

```
{
  "sourceCredentialsInfos": [
    {
      "authType": "auth-type",
      "serverType": "server-type",
      "arn": "arn"
    }
  ]
}
```

El `sourceCredentialsObject` contiene una lista de información de credenciales de código fuente conectado:

- El `authType` es el tipo de autenticación que utilizan las credenciales. Este valor puede ser `OAUTH`, `BASIC_AUTH` o `PERSONAL_ACCESS_TOKEN`.
 - El `serverType` es el tipo de proveedor de código fuente. Este valor puede ser `GITHUB`, `GITHUB_ENTERPRISE` o `BITBUCKET`.
 - El `arn` es el ARN del token.
4. Para desconectarse de un proveedor de código fuente y eliminar sus tokens de acceso, ejecute el comando `delete-source-credentials` con su ARN.

```
aws codebuild delete-source-credentials --arn arn-of-your-credentials
```

Los datos con formato JSON se devuelven con un ARN de las credenciales eliminadas.

```
{  
  "arn": "arn:aws:codebuild:region:account-id:token/server-type"  
}
```

GitHub Aplicación OAuth

Conectarse GitHub mediante OAuth (consola)

Para usar la consola para conectar tu proyecto a GitHub una aplicación OAuth, haz lo siguiente al crear un proyecto. Para obtener más información, consulte [Creación de un proyecto de compilación \(consola\)](#).

1. En Proveedor de fuentes, selecciona. GitHub
2. En Repository, selecciona Connect using OAuth.
3. Selecciona Conectar a GitHub, iniciar sesión y autorizar tu cuenta.
4. Selecciona Confirmar para conectarte CodeBuild a tu GitHub cuenta.
5. En el GitHub repositorio, introduce el enlace de tu GitHub repositorio.

Para revisar sus aplicaciones OAuth autorizadas, vaya a [Aplicaciones](#) en y compruebe que aparezca en GitHub la lista una aplicación cuyo nombre sea AWS CodeBuild (*region*) propiedad de [aws-codesuite](#).

Contraseña o token de acceso de la aplicación Bitbucket

Requisitos previos

Antes de empezar, debes añadir los ámbitos de permiso adecuados a la contraseña o al token de acceso de tu aplicación de Bitbucket.

En el caso de Bitbucket, la contraseña o el token de acceso de tu aplicación deben tener los siguientes ámbitos.

- `repository:read`: Concede acceso de lectura a todos los repositorios a los que tiene acceso el usuario de autorización.
- `pullrequest:read`: Concede acceso de lectura a las solicitudes de extracción. Si tu proyecto tiene un webhook de Bitbucket, la contraseña o el token de acceso de tu aplicación deben tener este alcance.
- `webhook`: Concede acceso a webhooks. Si tu proyecto tiene una operación de webhook, la contraseña o el token de acceso de tu aplicación deben tener este alcance.

Para obtener más información, consulte [Scopes for Bitbucket Cloud REST API](#) y [OAuth on Bitbucket Cloud](#) en el sitio web de Bitbucket.

Conectarse a Bitbucket con una contraseña de aplicación (consola)

Para utilizar la consola para conectar el proyecto a Bitbucket con una contraseña de aplicación, haga lo siguiente a la hora de crear un proyecto. Para obtener más información, consulte [Creación de un proyecto de compilación \(consola\)](#).

1. En Source provider (Proveedor de código fuente), elija Bitbucket.

Note

CodeBuild no es compatible con Bitbucket Server.

2. En Repository (Repositorio), elija Connect with a Bitbucket app password (Conectar con una contraseña de aplicación de Bitbucket).
3. En Bitbucket username (Nombre de usuario de Bitbucket), escriba su nombre de usuario de Bitbucket.
4. En Bitbucket app password (Contraseña de aplicación de Bitbucket), escriba su contraseña de aplicación de Bitbucket.
5. Seleccione Save Bitbucket credentials (Guardar credenciales de Bitbucket).

Connect Bitbucket con un token de acceso (consola)

Para usar la consola para conectar tu proyecto a Bitbucket mediante un token de acceso, haz lo siguiente al crear un proyecto. Para obtener más información, consulte [Creación de un proyecto de compilación \(consola\)](#).

1. En Source provider (Proveedor de código fuente), elija Bitbucket.

 Note

CodeBuild no es compatible con Bitbucket Server.

2. En Repositorio, selecciona Conectar con un token de acceso de Bitbucket.
3. En el token de acceso de Bitbucket, introduce tu token de acceso de Bitbucket.
4. Seleccione Save token (Guardar token).

Connect Bitbucket con una contraseña de aplicación o un token de acceso (CLI)

Sigue estos pasos para usar el y conectar tu proyecto AWS CLI a Bitbucket mediante una contraseña de aplicación o un token de acceso. Para obtener información sobre cómo usar el AWS CLI with AWS CodeBuild, consulta la [Referencia de la línea de comandos](#).

1. Ejecute el comando import-source-credentials:

```
aws codebuild import-source-credentials --generate-cli-skeleton
```

En el resultado se muestran datos con formato JSON. Copie los datos en un archivo (por ejemplo, *import-source-credentials.json*) en una ubicación del equipo o instancia local en la que AWS CLI esté instalado. Modifique los datos copiados como se indica a continuación y guarde los resultados.

```
{
  "serverType": "BITBUCKET",
  "authType": "auth-type",
  "shouldOverwrite": "should-overwrite",
  "token": "token",
  "username": "username"
}
```

Sustituya lo siguiente:

- *tipo-autenticación*: valor obligatorio. El tipo de autenticación que se utiliza para conectarse a un repositorio de Bitbucket. Los valores válidos incluyen

PERSONAL_ACCESS_TOKEN y BASIC_AUTH. No puedes usar la CodeBuild API para crear una conexión OAuth. Debe usar la consola de CodeBuild en su lugar.

- *should-override*: valor opcional. Establézcalo en `false` para impedir que se sobrescriban las credenciales de origen del repositorio. Establézcalo en `true` para sobrescribir las credenciales de origen del repositorio. El valor predeterminado es `true`.
 - *token*: valor obligatorio. En el caso de Bitbucket, se trata del token de acceso o de la contraseña de la aplicación.
 - *nombre de usuario*: valor opcional. El nombre de usuario de Bitbucket cuando el `authType` es BASIC_AUTH. Este parámetro se omite para otros tipos de proveedores de código fuente o conexiones.
2. Para conectar tu cuenta con una contraseña de aplicación o un token de acceso, cambia al directorio que contiene el `import-source-credentials.json` archivo que guardaste en el paso 1 y vuelve a ejecutar el `import-source-credentials` comando.

```
aws codebuild import-source-credentials --cli-input-json file://import-source-credentials.json
```

Los datos con formato JSON aparecen en el resultado con un nombre de recurso de Amazon (ARN).

```
{
  "arn": "arn:aws:codebuild:region:account-id:token/server-type"
}
```

Note

Si ejecuta el comando `import-source-credentials` con el mismo tipo de servidor y tipo de autorización una segunda vez, se actualiza el token de acceso almacenado.

Una vez que tu cuenta esté conectada con una contraseña de aplicación, podrás `create-project` utilizarla para crear tu CodeBuild proyecto. Para obtener más información, consulte [Crear un proyecto de compilación \(AWS CLI\)](#).

3. Para ver las contraseñas de las aplicaciones conectadas o los tokens de acceso, ejecuta el `list-source-credentials` comando.

```
aws codebuild list-source-credentials
```

Un objeto `sourceCredentialsInfos` con formato JSON aparece en el resultado:

```
{
  "sourceCredentialsInfos": [
    {
      "authType": "auth-type",
      "serverType": "BITBUCKET",
      "arn": "arn"
    }
  ]
}
```

El `sourceCredentialsObject` contiene una lista de información de credenciales de código fuente conectado:

- El `authType` es el tipo de autenticación que utilizan las credenciales. Este valor puede ser OAUTH, BASIC_AUTH o PERSONAL_ACCESS_TOKEN.
 - El `arn` es el ARN del token.
4. Para desconectarse de un proveedor de origen y eliminar la contraseña de la aplicación o los tokens de acceso, ejecute el `delete-source-credentials` comando con su ARN.

```
aws codebuild delete-source-credentials --arn arn-of-your-credentials
```

Los datos con formato JSON se devuelven con un ARN de las credenciales eliminadas.

```
{
  "arn": "arn:aws:codebuild:region:account-id:token/server-type"
}
```

Aplicación OAuth de Bitbucket

Connect Bitbucket mediante OAuth (consola)

Para usar la consola para conectar tu proyecto a Bitbucket mediante una aplicación OAuth, haz lo siguiente al crear un proyecto. Para obtener más información, consulte [Creación de un proyecto de compilación \(consola\)](#).

1. En Source provider (Proveedor de código fuente), elija Bitbucket.
2. En Repository, selecciona Connect using OAuth.
3. Elige Connect to Bitbucket, inicia sesión y autoriza tu cuenta.
4. Selecciona Confirmar para conectarte CodeBuild a tu cuenta de Bitbucket.
5. En el repositorio de Bitbucket, introduce el enlace de tu repositorio de Bitbucket.

Para revisar tus aplicaciones OAuth autorizadas, dirígete a [Autorizaciones de aplicaciones en Bitbucket y comprueba que aparezca una aplicación](#) con su nombre. AWS CodeBuild (*region*)

Prevención de la sustitución confusa entre servicios

El problema de la sustitución confusa es un problema de seguridad en el que una entidad que no tiene permiso para realizar una acción puede obligar a una entidad con más privilegios a realizar la acción. En AWS, la suplantación de identidad entre servicios puede provocar el confuso problema de un diputado. La suplantación entre servicios puede producirse cuando un servicio (el servicio que lleva a cabo las llamadas) llama a otro servicio (el servicio al que se llama). El servicio que lleva a cabo las llamadas se puede manipular para utilizar sus permisos a fin de actuar en función de los recursos de otro cliente de una manera en la que no debe tener permiso para acceder. Para evitarlo, AWS proporciona herramientas que lo ayudan a proteger sus datos para todos los servicios con entidades principales de servicio a las que se les ha dado acceso a los recursos de su cuenta.

Se recomienda utilizar las claves de contexto de condición [aws:SourceAccount](#) global [aws:SourceArn](#) las claves de contexto en las políticas de recursos para limitar los permisos que se AWS CodeBuild otorgan a otro servicio al recurso. Utilice `aws:SourceArn` si desea que solo se asocie un recurso al acceso entre servicios. Utilice `aws:SourceAccount` si quiere permitir que cualquier recurso de esa cuenta se asocie al uso entre servicios.

La forma más eficaz de protegerse contra el problema de la sustitución confusa es utilizar la clave de contexto de condición global de `aws:SourceArn` con el ARN completo del recurso. Si no conoce

el ARN completo del recurso o si está especificando varios recursos, utilice la clave de condición de contexto global `aws:SourceArn` con caracteres comodines (*) para las partes desconocidas del ARN. Por ejemplo, `arn:aws:codebuild:*:123456789012:*`.

Si el valor de `aws:SourceArn` no contiene el ID de cuenta, como un ARN de bucket de Amazon S3, debe utilizar ambas claves de contexto de condición global para limitar los permisos.

El valor de `aws:SourceArn` debe ser el ARN CodeBuild del proyecto.

El siguiente ejemplo muestra cómo puede utilizar las claves de contexto de condición `aws:SourceAccount` global `aws:SourceArn` y las claves contextuales CodeBuild para evitar el confuso problema de los diputados.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "codebuild.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceArn": "arn:aws:codebuild:region-ID:account-ID:project/project-name"
        }
      }
    }
  ]
}
```

Temas avanzados

Esta sección contiene varios temas avanzados que son útiles para los usuarios de AWS CodeBuild con más experiencia.

Temas

- [Configuración avanzada](#)
- [Referencia de la línea de comandos de AWS CodeBuild](#)
- [Referencia de los SDK y las herramientas de AWS para AWS CodeBuild](#)
- [Especificación del punto de conexión de AWS CodeBuild](#)
- [Usar AWS CodePipeline con AWS CodeBuild para probar el código y ejecutar compilaciones](#)
- [Uso de AWS CodeBuild con Jenkins](#)
- [Uso de AWS CodeBuild con Codecov](#)
- [Utilizar AWS CodeBuild con aplicaciones sin servidor](#)

Configuración avanzada

Si sigue los pasos de [Primeros pasos con la consola](#) para acceder a AWS CodeBuild por primera vez, lo más probable es que no necesite la información de este tema. Sin embargo, si sigue usando CodeBuild, es posible que quiera realizar algunas acciones, como conceder a los grupos y usuarios de IAM de su organización acceso a CodeBuild, modificar los roles de servicio existentes en IAM o AWS KMS keys para acceder a CodeBuild, o configurar AWS CLI en las estaciones de trabajo de su organización para acceder a CodeBuild. En este tema se describe cómo realizar los pasos de configuración relacionados.

En él se presupone que ya tiene una cuenta de AWS. No obstante, si todavía no tiene una, vaya a <http://aws.amazon.com>, elija Sign In to the Console y siga las instrucciones en línea.

Temas

- [Agregar permisos de acceso de CodeBuild a un grupo o un usuario de IAM](#)
- [Crear un rol de servicio de CodeBuild](#)
- [Creación y configuración de una clave administrada por el cliente para CodeBuild](#)
- [Instalación y configuración de AWS CLI](#)

Agregar permisos de acceso de CodeBuild a un grupo o un usuario de IAM

Para acceder a AWS CodeBuild con un grupo o un usuario de IAM, debe agregar permisos de acceso. En esta sección, se describe cómo hacer esto con la consola de IAM o la AWS CLI.

Si va a obtener acceso a CodeBuild con una cuenta de usuario raíz de AWS (no se recomienda) o un usuario administrador de la cuenta de AWS, no es necesario que siga estas instrucciones.

Para obtener información sobre las cuentas de usuario raíz y los usuarios administradores de AWS, consulte [El usuario raíz de Cuenta de AWS](#) y [Creación del primer usuario raíz y grupo de Cuenta de AWS](#) en la Guía del usuario.

Para añadir permisos de acceso de CodeBuild a un grupo o a un usuario de IAM (consola)

1. Abra la consola de IAM en <https://console.aws.amazon.com/iam/>.

Debe haber iniciado sesión en la AWS Management Console con alguna de las entidades siguientes:

- Su cuenta raíz de AWS. No se recomienda. Para obtener más información, consulte [El usuario raíz de Cuenta de AWS](#) en la Guía del usuario.
- Un usuario administrador en su cuenta de AWS. Para obtener más información, consulte [Creación del primer usuario raíz y grupo de Cuenta de AWS](#) en la Guía del usuario.
- Un usuario en su cuenta de AWS con permiso para realizar el conjunto mínimo de acciones siguiente:

```
iam:AttachGroupPolicy
iam:AttachUserPolicy
iam:CreatePolicy
iam>ListAttachedGroupPolicies
iam>ListAttachedUserPolicies
iam>ListGroups
iam>ListPolicies
iam>ListUsers
```

Para obtener más información, consulte [Información general sobre las políticas de IAM](#) en la Guía del usuario.

2. En el panel de navegación, seleccione Políticas (Políticas).

3. Para añadir un conjunto personalizado de permisos de acceso de AWS CodeBuild a un grupo o a un usuario de IAM, vaya al paso 4 de este procedimiento.

Para añadir un conjunto predeterminado de permisos de acceso de CodeBuild a un grupo o a un usuario de IAM, seleccione Tipo de política, Administrada por AWS y, a continuación, haga lo siguiente:

- Para agregar permisos de acceso completo a CodeBuild, seleccione el cuadro `AWSCodeBuildAdminAccess`, elija Acciones de la política y, a continuación, elija Asociar. Active la casilla situada junto al grupo o el usuario de IAM de destino y, a continuación, seleccione Asociar política. Repita esta operación para las políticas denominadas `AmazonS3ReadOnlyAccess` e `IAMFullAccess`.
- Si desea agregar permisos de acceso a CodeBuild para todo excepto para la administración del proyecto de compilación, seleccione la casilla `AWSCodeBuildDeveloperAccess`, seleccione Acciones de la política y, a continuación, Asociar. Active la casilla situada junto al grupo o el usuario de IAM de destino y, a continuación, seleccione Asociar política. Repita esta operación para la política `AmazonS3ReadOnlyAccess`.
- Para añadir permisos de solo lectura a CodeBuild, active las casillas `AWSCodeBuildReadOnlyAccess`. Active la casilla situada junto al grupo o el usuario de IAM de destino y, a continuación, seleccione Asociar política. Repita esta operación para la política `AmazonS3ReadOnlyAccess`.

Ahora ha añadido un conjunto predeterminado de permisos de acceso de CodeBuild a un grupo o a un usuario de IAM. Omite el resto de los pasos de este procedimiento.

4. Elija Create Policy (Crear política).
5. En la página Create Policy, junto a Create Your Own Policy, elija Select.
6. En la página Review Policy (Revisar política), en Policy Name (Nombre de política), escriba un nombre para la política (por ejemplo, **CodeBuildAccessPolicy**). Si elige otro nombre, no olvide utilizarlo durante todo este procedimiento.
7. En Policy Document (Documento de la política), escriba lo siguiente y elija Create Policy (Crear política).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```
"Sid": "CodeBuildAccessPolicy",
"Effect": "Allow",
"Action": [
  "codebuild:*"
],
"Resource": "*"
},
{
  "Sid": "CodeBuildRolePolicy",
  "Effect": "Allow",
  "Action": [
    "iam:PassRole"
  ],
  "Resource": "arn:aws:iam::account-ID:role/role-name"
},
{
  "Sid": "CloudWatchLogsAccessPolicy",
  "Effect": "Allow",
  "Action": [
    "logs:FilterLogEvents",
    "logs:GetLogEvents"
  ],
  "Resource": "*"
},
{
  "Sid": "S3AccessPolicy",
  "Effect": "Allow",
  "Action": [
    "s3:CreateBucket",
    "s3:GetObject",
    "s3:List*",
    "s3:PutObject"
  ],
  "Resource": "*"
},
{
  "Sid": "S3BucketIdentity",
  "Effect": "Allow",
  "Action": [
    "s3:GetBucketAcl",
    "s3:GetBucketLocation"
  ],
  "Resource": "*"
}
```

```
]
}
```

Note

Esta política permite el acceso a todas las acciones de CodeBuild y a un número potencialmente grande de recursos de AWS. Para restringir los permisos a acciones de CodeBuild específicas, cambie el valor de `codebuild:*` en la instrucción de política de CodeBuild. Para obtener más información, consulte [Administración de identidades y accesos](#). Para restringir el acceso a recursos de AWS específicos, cambie el valor del objeto `Resource`. Para obtener más información, consulte [Administración de identidades y accesos](#).

La declaración `CodeBuildRolePolicy` es necesaria para permitir la creación o modificación de un proyecto de compilación.

8. En el panel de navegación, elija Groups o Users.
9. En la lista de grupos o usuarios, elija el nombre del grupo o el usuario de IAM al que desea añadir permisos de acceso de CodeBuild.
10. Si se trata de un grupo, en la página de configuración del grupo, en la pestaña Permissions (Permisos), expanda Managed Policies (Políticas administradas) y elija Attach Policy (Asociar política).

Para un usuario, en la página de configuración del usuario, en la pestaña Permissions, seleccione Add permissions.

11. En un grupo, en la página Asociar política, seleccione CodeBuildAccessPolicy y haga clic en Asociar política.

Para un usuario, en la página Añadir permisos, elija Asociar políticas existentes directamente. Seleccione AccessCodeBuildAccessPolicy, Siguiente: revisión y Añadir permisos.

Para añadir permisos de acceso de CodeBuild a un grupo o a un usuario de IAM (AWS CLI)

1. Asegúrese de que ha configurado la AWS CLI con la clave de acceso de AWS y la clave de acceso secreta de AWS correspondientes a alguna de las entidades de IAM, tal y como se ha descrito en el procedimiento anterior. Para obtener más información, consulte [Configuración inicial de la AWS Command Line Interface](#) en la Guía del usuario de AWS Command Line Interface.

2. Para agregar un conjunto personalizado de permisos de acceso de AWS CodeBuild a un grupo o un usuario de IAM, vaya al paso 3 de este procedimiento.

Para añadir un conjunto predeterminado de permisos de acceso de CodeBuild a un grupo o usuario de IAM, haga lo siguiente:

Ejecute uno de los comandos siguientes, en función de si desea añadir permisos a un usuario o un grupo de IAM:

```
aws iam attach-group-policy --group-name group-name --policy-arn policy-arn

aws iam attach-user-policy --user-name user-name --policy-arn policy-arn
```

Debe ejecutar el comando tres veces, reemplazando *group-name* o *user-name* por el nombre del grupo o de usuario de IAM, y reemplazando *policy-arn* una vez por cada uno de los siguientes nombres de recursos de Amazon (ARN) de políticas:

- Para añadir permisos de acceso completos a CodeBuild, use los ARN de política siguientes:
 - `arn:aws:iam::aws:policy/AWSCodeBuildAdminAccess`
 - `arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess`
 - `arn:aws:iam::aws:policy/IAMFullAccess`
- Para añadir permisos de acceso a CodeBuild para todo excepto para administrar proyectos de compilación, use los ARN de política siguientes:
 - `arn:aws:iam::aws:policy/AWSCodeBuildDeveloperAccess`
 - `arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess`
- Para añadir permisos de acceso de solo lectura a CodeBuild, use los ARN de política siguientes:
 - `arn:aws:iam::aws:policy/AWSCodeBuildReadOnlyAccess`
 - `arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess`

Ahora ha añadido un conjunto predeterminado de permisos de acceso de CodeBuild a un grupo o a un usuario de IAM. Omite el resto de los pasos de este procedimiento.

3. En un directorio vacío de la estación de trabajo o la instancia local en la que esté instalada la AWS CLI, cree un archivo denominado `put-group-policy.json` o `put-user-`

`policy.json`. Si elige otro nombre de archivo, no olvide utilizarlo durante todo este procedimiento.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CodeBuildAccessPolicy",
      "Effect": "Allow",
      "Action": [
        "codebuild:*"
      ],
      "Resource": "*"
    },
    {
      "Sid": "CodeBuildRolePolicy",
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": "arn:aws:iam::account-ID:role/role-name"
    },
    {
      "Sid": "CloudWatchLogsAccessPolicy",
      "Effect": "Allow",
      "Action": [
        "logs:FilterLogEvents",
        "logs:GetLogEvents"
      ],
      "Resource": "*"
    },
    {
      "Sid": "S3AccessPolicy",
      "Effect": "Allow",
      "Action": [
        "s3:CreateBucket",
        "s3:GetObject",
        "s3:List*",
        "s3:PutObject"
      ],
      "Resource": "*"
    },
    {

```

```

    "Sid": "S3BucketIdentity",
    "Effect": "Allow",
    "Action": [
        "s3:GetBucketAcl",
        "s3:GetBucketLocation"
    ],
    "Resource": "*"
  }
]
}

```

Note

Esta política permite el acceso a todas las acciones de CodeBuild y a un número potencialmente grande de recursos de AWS. Para restringir los permisos a acciones de CodeBuild específicas, cambie el valor de `codebuild:*` en la instrucción de política de CodeBuild. Para obtener más información, consulte [Administración de identidades y accesos](#). Para restringir el acceso a recursos de AWS específicos, cambie el valor del objeto `Resource` relacionado. Para obtener más información, consulte [Administración de identidades y accesos](#) o la documentación de seguridad del servicio de AWS específico.

La declaración `CodeBuildRolePolicy` es necesaria para permitir la creación o modificación de un proyecto de compilación.

4. Cambie al directorio donde ha guardado el archivo y, a continuación, ejecute uno de los siguientes comandos. Puede utilizar diferentes valores para `CodeBuildGroupAccessPolicy` y `CodeBuildUserAccessPolicy`. Si emplea valores diferentes, asegúrese de usarlos aquí.

Para un grupo de IAM:

```
aws iam put-group-policy --group-name group-name --policy-name
CodeBuildGroupAccessPolicy --policy-document file://put-group-policy.json
```

Para un usuario de :

```
aws iam put-user-policy --user-name user-name --policy-name
CodeBuildUserAccessPolicy --policy-document file://put-user-policy.json
```

En los comandos anteriores, sustituya *group-name* o *user-name* por el nombre de grupo o de usuario de IAM de destino.

Crear un rol de servicio de CodeBuild

Necesita un rol de servicio de AWS CodeBuild para que CodeBuild pueda interactuar en su nombre con los servicios de AWS dependientes. Puede crear un rol de servicio de CodeBuild utilizando las consolas de CodeBuild o AWS CodePipeline. Para obtener información, consulte:

- [Creación de un proyecto de compilación \(consola\)](#)
- [Crear una canalización que use CodeBuild \(consola de CodePipeline\)](#)
- [Agregar una acción de compilación de CodeBuild a una canalización \(consola de CodePipeline\)](#)
- [Cambiar la configuración de un proyecto de compilación \(consola\)](#)

Si no tiene intención de utilizar estas consolas, en esta sección se describe cómo crear un rol de servicio de CodeBuild con la consola de IAM o la AWS CLI.

Important

CodeBuild utiliza el rol de servicio para todas las operaciones que se realizan en su nombre. Si el rol incluye permisos que el usuario no debería tener, puede escalar involuntariamente los permisos de un usuario. Asegúrese de que el rol concede [privilegios mínimos](#). El rol de servicio que se describe en esta página contiene una política que concede los permisos mínimos necesarios para utilizar CodeBuild. Es posible que tenga que añadir permisos adicionales en función del caso de uso.

Para crear un rol de servicio de CodeBuild (consola)

1. Abra la consola de IAM en <https://console.aws.amazon.com/iam/>.

Debe haber iniciado sesión en la consola con alguna de las entidades siguientes:

- Su cuenta raíz de AWS. No se recomienda. Para obtener más información, consulte [El usuario raíz de Cuenta de AWS](#) en la Guía del usuario.

- Un usuario administrador en su cuenta de AWS. Para obtener más información, consulte [Creación del primer usuario raíz y grupo de Cuenta de AWS](#) en la Guía del usuario.
- Un usuario en su cuenta de AWS con permiso para realizar el conjunto mínimo de acciones siguiente:

```
iam:AddRoleToInstanceProfile
iam:AttachRolePolicy
iam:CreateInstanceProfile
iam:CreatePolicy
iam:CreateRole
iam:GetRole
iam>ListAttachedRolePolicies
iam>ListPolicies
iam>ListRoles
iam:PassRole
iam:PutRolePolicy
iam:UpdateAssumeRolePolicy
```

Para obtener más información, consulte [Información general sobre las políticas de IAM](#) en la Guía del usuario.

2. En el panel de navegación, seleccione Políticas (Políticas).
3. Elija Create Policy (Crear política).
4. En la página Create Policy, elija JSON.
5. En la política JSON, escriba lo siguiente y elija Review Policy (Consultar política):

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CloudWatchLogsPolicy",
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "*"
    },
    {
      "Sid": "CodeCommitPolicy",
```

```
"Effect": "Allow",
"Action": [
  "codecommit:GitPull"
],
"Resource": "*"
},
{
  "Sid": "S3GetObjectPolicy",
  "Effect": "Allow",
  "Action": [
    "s3:GetObject",
    "s3:GetObjectVersion"
  ],
  "Resource": "*"
},
{
  "Sid": "S3PutObjectPolicy",
  "Effect": "Allow",
  "Action": [
    "s3:PutObject"
  ],
  "Resource": "*"
},
{
  "Sid": "ECRPullPolicy",
  "Effect": "Allow",
  "Action": [
    "ecr:BatchCheckLayerAvailability",
    "ecr:GetDownloadUrlForLayer",
    "ecr:BatchGetImage"
  ],
  "Resource": "*"
},
{
  "Sid": "ECRAuthPolicy",
  "Effect": "Allow",
  "Action": [
    "ecr:GetAuthorizationToken"
  ],
  "Resource": "*"
},
{
  "Sid": "S3BucketIdentity",
  "Effect": "Allow",
```

```
"Action": [  
  "s3:GetBucketAcl",  
  "s3:GetBucketLocation"  
],  
"Resource": "*" ]  
}
```

Note

Esta política contiene instrucciones que permiten tener acceso a número potencialmente grande de recursos de AWS. Para restringir el acceso de AWS CodeBuild a recursos de AWS específicos, cambie el valor de la matriz Resource. Para obtener más información, consulte la documentación de seguridad del servicio de AWS.

6. En la página Review Policy (Consultar política), en Policy Name (Nombre de la política), escriba un nombre para la política (por ejemplo, **CodeBuildServiceRolePolicy**) y elija Create policy (Crear política).

Note

Si elige otro nombre, no olvide utilizarlo durante todo este procedimiento.

7. Seleccione Roles en el panel de navegación.
8. Elija Create role (Crear rol).
9. En la página Crear rol, con la opción Servicio AWS seleccionada, elija CodeBuild y después elija Siguiente: Permisos.
10. En la página Asociar políticas de permisos, seleccione SerCodeBuildServiceRolePolicy y Siguiente: Revisión.
11. En la página Create role and review (Crear función y revisar), en Role name (Nombre del rol), escriba un nombre para el rol (por ejemplo, **CodeBuildServiceRole**) y elija Create role (Crear rol).

Crear un rol de servicio de CodeBuild (AWS CLI)

1. Asegúrese de que ha configurado la AWS CLI con la clave de acceso de AWS y la clave de acceso secreta de AWS correspondientes a alguna de las entidades de IAM, tal y como se ha descrito en el procedimiento anterior. Para obtener más información, consulte [Configuración inicial de la AWS Command Line Interface](#) en la Guía del usuario de AWS Command Line Interface.
2. En un directorio vacío de la estación de trabajo o la instancia local donde se ha instalado la AWS CLI, cree dos archivos llamados `create-role.json` y `put-role-policy.json`. Si elige nombres de archivo diferentes, no olvide utilizarlos durante todo este procedimiento.

`create-role.json`:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "codebuild.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Note

Le recomendamos que utilice las claves de condición `aws:SourceAccount` y `aws:SourceArn` para protegerse contra el [problema del suplente confuso](#). Por ejemplo, podría editar la política de confianza anterior con los bloques de condición siguientes. `aws:SourceAccount` es el propietario del proyecto de CodeBuild y `aws:SourceArn` es el ARN del proyecto de CodeBuild.

Si desea restringir su rol de servicio a una cuenta de AWS, `create-role.json` podría tener un aspecto similar al siguiente:

```
{
```

```

"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Principal": {
      "Service": "codebuild.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "StringEquals": {
        "aws:SourceAccount": [
          "account-ID"
        ]
      }
    }
  }
]
}

```

Si desea restringir su rol de servicio a un proyecto específico de CodeBuild, `create-role.json` podría tener un aspecto similar al siguiente:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "codebuild.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceArn": "arn:aws:codebuild:region-ID:account-ID:project/project-name"
        }
      }
    }
  ]
}

```

Note

Si no conoce o no ha elegido un nombre para su proyecto de CodeBuild y desea una restricción de la política de confianza para un patrón de ARN en particular, puede reemplazar esa parte del ARN por un comodín (*). Después de crear el proyecto, puede actualizar la política de confianza.

put-role-policy.json:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CloudWatchLogsPolicy",
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "*"
    },
    {
      "Sid": "CodeCommitPolicy",
      "Effect": "Allow",
      "Action": [
        "codecommit:GitPull"
      ],
      "Resource": "*"
    },
    {
      "Sid": "S3GetObjectPolicy",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion"
      ],
      "Resource": "*"
    }
  ]
}
```

```
    "Sid": "S3PutObjectPolicy",
    "Effect": "Allow",
    "Action": [
      "s3:PutObject"
    ],
    "Resource": "*"
  },
  {
    "Sid": "S3BucketIdentity",
    "Effect": "Allow",
    "Action": [
      "s3:GetBucketAcl",
      "s3:GetBucketLocation"
    ],
    "Resource": "*"
  }
]
```

Note

Esta política contiene instrucciones que permiten tener acceso a número potencialmente grande de recursos de AWS. Para restringir el acceso de AWS CodeBuild a recursos de AWS específicos, cambie el valor de la matriz `Resource`. Para obtener más información, consulte la documentación de seguridad del servicio de AWS.

3. Cambie al directorio donde guardó los archivos anteriores y, a continuación, ejecute los dos comandos siguientes, uno cada vez, en este orden. Puede utilizar valores diferentes para `CodeBuildServiceRole` y `CodeBuildServiceRolePolicy`, pero no olvide emplearlos aquí.

```
aws iam create-role --role-name CodeBuildServiceRole --assume-role-policy-document
file://create-role.json
```

```
aws iam put-role-policy --role-name CodeBuildServiceRole --policy-name
CodeBuildServiceRolePolicy --policy-document file://put-role-policy.json
```

Creación y configuración de una clave administrada por el cliente para CodeBuild

Para que AWS CodeBuild cifre los artefactos de salida de la compilación, necesita acceso a una clave KMS. De forma predeterminada, CodeBuild utiliza Clave administrada de AWS para Amazon S3 en su cuenta de AWS.

Si no desea utilizar la Clave administrada de AWS, debe crear y configurar una clave administrada por el cliente por su cuenta. En esta sección se describe cómo hacer esto con la consola de IAM.

Para obtener más información acerca de las claves administradas por el cliente, consulte [Conceptos de AWS Key Management Service](#) y [Creación de claves](#) en la Guía para desarrolladores de AWS KMS.

Si desea configurar una clave administrada por el cliente para utilizarla en CodeBuildServiceRolePolicy, siga las instrucciones que se indican en la sección "Cómo modificar una política de claves" de [Modificación de una política de claves en la Guía para desarrolladores de AWS KMS](#). A continuación, añada las siguientes instrucciones (entre **### BEGIN ADDING STATEMENTS HERE ###** y **### END ADDING STATEMENTS HERE ###**) a la política de la clave. Los puntos suspensivos (. . .) se usan por motivos de brevedad y para ayudarle a encontrar el lugar donde debe añadir las instrucciones. No elimine ninguna instrucción ni incluya estos puntos suspensivos en la política de la clave.

```
{
  "Version": "2012-10-17",
  "Id": "...",
  "Statement": [
    ### BEGIN ADDING STATEMENTS HERE ###
    {
      "Sid": "Allow access through Amazon S3 for all principals in the account that are
authorized to use Amazon S3",
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
      ]
    }
  ]
}
```

```
    ],
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "kms:ViaService": "s3.region-ID.amazonaws.com",
        "kms:CallerAccount": "account-ID"
      }
    }
  },
  {
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::account-ID:role/CodeBuild-service-role"
    },
    "Action": [
      "kms:Encrypt",
      "kms:Decrypt",
      "kms:ReEncrypt*",
      "kms:GenerateDataKey*",
      "kms:DescribeKey"
    ],
    "Resource": "*"
  },
  ### END ADDING STATEMENTS HERE ###
  {
    "Sid": "Enable IAM User Permissions",
    ...
  },
  {
    "Sid": "Allow access for Key Administrators",
    ...
  },
  {
    "Sid": "Allow use of the key",
    ...
  },
  {
    "Sid": "Allow attachment of persistent resources",
    ...
  }
]
}
```

- *region-ID* representa el ID de la región de AWS donde se encuentran los buckets de Amazon S3 asociados a CodeBuild (por ejemplo, us-east-1).
- *account-ID* representa el ID de la cuenta de AWS que posee la clave administrada por el cliente.
- *CodeBuild-service-role* representa el nombre del rol de servicio de CodeBuild que se ha creado o identificado anteriormente en este tema.

Note

Para crear o configurar una clave administrada por el cliente a través de la consola de IAM, primero debe iniciar sesión en AWS Management Console utilizando alguna de las opciones siguientes:

- Su cuenta raíz de AWS. No se recomienda. Para obtener más información, consulte [El usuario raíz de la cuenta](#) en la Guía del usuario.
- Un usuario administrador en su cuenta de AWS. Para obtener más información, consulte [Creación del primer usuario raíz y grupo de Cuenta de AWS](#) en la Guía del usuario.
- Un usuario de su cuenta de AWS con permiso para crear o modificar la clave administrada por el cliente. Para obtener más información, consulte [Permisos necesarios para usar la consola de AWS KMS](#) en la Guía para desarrolladores de AWS KMS.

Instalación y configuración de AWS CLI

Para acceder a AWS CodeBuild, puede utilizar la AWS CLI con (o en lugar de) la consola de CodeBuild, la consola de CodePipeline o los SDK de AWS. Para instalar y configurar la AWS CLI, consulte [Instalación de AWS Command Line Interface](#) en la Guía del usuario de AWS Command Line Interface.

1. Ejecute el comando siguiente para confirmar si la instalación de AWS CLI admite CodeBuild:

```
aws codebuild list-builds
```

Si el comando se ejecuta correctamente, aparecerá información similar a la siguiente en el resultado:

```
{  
  "ids": []
```

```
}
```

Los corchetes vacíos indican que aún no ha ejecutado ninguna compilación.

2. Si se produce un error, debe desinstalar la versión actual de la AWS CLI y, a continuación, instalar la versión más reciente. Para obtener más información, consulte el tema acerca de [cómo desinstalar la AWS CLI](#) y [cómo instalar la AWS Command Line Interface](#) en la Guía del usuario de AWS Command Line Interface.

Referencia de la línea de comandos de AWS CodeBuild

La AWS CLI dispone de comandos para automatizar AWS CodeBuild. Utilice la información de este tema como complemento de la [Guía del usuario de AWS Command Line Interface](#) y la [Referencia de AWS CLI para AWS CodeBuild](#).

¿No es lo que está buscando? Si desea utilizar los SDK de AWS para llamar a CodeBuild, consulte [Referencia de los SDK y las herramientas de AWS](#).

Para utilizar la información de este tema, debe haber instalado la AWS CLI y haberla configurado para utilizarla con CodeBuild, tal y como se describe en [Instalación y configuración de AWS CLI](#).

Si desea utilizar la AWS CLI para especificar el punto de conexión de CodeBuild, consulte [Especificación del punto de conexión de AWS CodeBuild \(AWS CLI\)](#).

Ejecute este comando para obtener una lista de comandos de CodeBuild.

```
aws codebuild help
```

Ejecute este comando para obtener información acerca de un comando de CodeBuild, donde *command-name* es el nombre del comando.

```
aws codebuild command-name help
```

Los comandos de CodeBuild incluyen:

- `batch-delete-builds`: elimina una o varias compilaciones de CodeBuild. Para obtener más información, consulte [Eliminación de compilaciones \(AWS CLI\)](#).
- `batch-get-builds`: obtiene información sobre varias compilaciones de CodeBuild. Para obtener más información, consulte [Ver detalles de las compilaciones \(AWS CLI\)](#).

- `batch-get-projects`: obtiene información acerca de uno o varios proyectos de compilación especificados. Para obtener más información, consulte [Ver los detalles de un proyecto de compilación \(AWS CLI\)](#).
- `create-project`: crea un proyecto de compilación. Para obtener más información, consulte [Crear un proyecto de compilación \(AWS CLI\)](#).
- `delete-project`: elimina un proyecto de compilación. Para obtener más información, consulte [Eliminar un proyecto de compilación \(AWS CLI\)](#).
- `list-builds`: muestra los nombres de recursos de Amazon (ARN) de las compilaciones de CodeBuild. Para obtener más información, consulte [Ver una lista de identificadores de compilación \(AWS CLI\)](#).
- `list-builds-for-project`: obtiene una lista de los identificadores de compilación asociados con un proyecto de compilación especificado. Para obtener más información, consulte [Ver una lista de identificadores de compilación de un proyecto de compilación \(AWS CLI\)](#).
- `list-curated-environment-images`: obtiene una lista con las imágenes de Docker administradas por CodeBuild que puede utilizar en sus compilaciones. Para obtener más información, consulte [Imágenes de Docker proporcionadas por CodeBuild](#).
- `list-projects`: obtiene una lista de nombres de proyecto de compilación. Para obtener más información, consulte [Ver una lista de nombres de proyectos de compilación \(AWS CLI\)](#).
- `start-build`: empieza a ejecutar una compilación. Para obtener más información, consulte [Ejecutar una compilación \(AWS CLI\)](#).
- `stop-build`: intenta detener la ejecución de la compilación especificada. Para obtener más información, consulte [Detener una compilación \(AWS CLI\)](#).
- `update-project`: cambia la información sobre el proyecto de compilación especificado. Para obtener más información, consulte [Cambiar la configuración de un proyecto de compilación \(AWS CLI\)](#).

Referencia de los SDK y las herramientas de AWS para AWS CodeBuild

Si desea utilizar uno de los SDK o las herramientas de AWS para automatizar AWS CodeBuild, consulte los siguientes recursos.

Si desea usar la AWS CLI para ejecutar CodeBuild, consulte [Referencia de la línea de comandos](#).

SDK y herramientas de AWS compatibles con AWS CodeBuild

Los siguientes SDK y herramientas de AWS son compatibles con CodeBuild:

- [SDK de AWS para C++](#). Para obtener más información, consulte la sección del espacio de nombres [Aws::CodeBuild](#) en la Referencia del SDK de AWS para la API de C++.
- [SDK de AWS para Go](#). Para obtener más información, consulte la sección [codebuild](#) de la Referencia del SDK de AWS para la API de Go.
- [SDK de AWS para Java](#). Para obtener más información, consulte las secciones `com.amazonaws.services.codebuild` y `com.amazonaws.services.codebuild.model` de la [Referencia del SDK de AWS para la API de Java](#).
- [SDK de AWS para JavaScript en el navegador](#) y [SDK de AWS para JavaScript en Node.js](#). Para obtener más información, consulte la sección [Class: AWS.CodeBuild](#) de la Referencia del SDK de AWS para la API de JavaScript.
- [SDK de AWS para .NET](#). Para obtener más información, consulte las secciones del espacio de nombres [Amazon.CodeBuild](#) y [Amazon.CodeBuild.Model](#) de la Referencia del SDK de AWS para la API de .NET.
- [SDK de AWS para PHP](#). Para obtener más información, consulte la sección [Namespace Aws \CodeBuild](#) de la Referencia del SDK de AWS para la API de PHP.
- [SDK de AWS para Python \(Boto3\)](#). Para obtener más información, consulte la sección [CodeBuild](#) de la Documentación de Boto 3.
- [SDK de AWS para Ruby](#). Para obtener más información, consulte la sección [Module: Aws::CodeBuild](#) de la Referencia del SDK de AWS para la API de Ruby.
- [Herramientas de AWS para PowerShell](#). Para obtener más información, consulte la sección [AWS CodeBuild](#) de la Referencia de herramientas de AWS para los cmdlet de PowerShell.

Especificación del punto de conexión de AWS CodeBuild

Puede utilizar la AWS Command Line Interface (AWS CLI) o uno de los SDK de AWS para especificar el punto de conexión que se utiliza en AWS CodeBuild. Hay un punto de conexión para cada región donde CodeBuild está disponible. Además de un punto de conexión regional, hay cuatro regiones que también tienen un punto de conexión que cumple el Estándar federal de procesamiento de información (FIPS). Para obtener más información sobre los puntos de enlace de FIPS, consulte [Información general sobre FIPS 140-2](#).

La especificación de un punto de conexión es opcional. Si no se indica explícitamente qué punto de conexión debe utilizar CodeBuild, el servicio usará el punto de conexión asociado a la región que utiliza la cuenta de AWS. CodeBuild no utiliza nunca de forma predeterminada un punto de conexión FIPS. Si desea utilizar un punto de conexión FIPS, debe asociarlo a CodeBuild mediante uno de los métodos siguientes.

Note

Puede utilizar un alias o un nombre de región para especificar un punto de conexión que usa un SDK de AWS. Si utiliza la AWS CLI, debe usar el nombre completo del punto de conexión.

Para obtener información sobre los puntos de conexión que se pueden utilizar con CodeBuild, consulte [Regiones y puntos de conexión de CodeBuild](#).

Temas

- [Especificación del punto de conexión de AWS CodeBuild \(AWS CLI\)](#)
- [Especificación del punto de conexión de AWS CodeBuild \(SDK de AWS\)](#)

Especificación del punto de conexión de AWS CodeBuild (AWS CLI)

Puede usar la AWS CLI para especificar el punto de conexión a través del que se obtiene acceso a AWS CodeBuild utilizando el argumento `--endpoint-url` en cualquiera de los comandos de CodeBuild. Por ejemplo, ejecute este comando para obtener una lista de los nombres de proyectos de compilación que utilizan el punto de conexión que cumple el Estándar federal de procesamiento de información (FIPS) en la región Este de EE. UU. (Norte de Virginia):

```
aws codebuild list-projects --endpoint-url https://codebuild-fips.us-east-1.amazonaws.com
```

Incluya el código `https://` al principio del punto de conexión.

El argumento `--endpoint-url` de la AWS CLI está disponible para todos los servicios de AWS. Para obtener más información sobre este y otros argumentos de la AWS CLI, consulte la [Referencia de comandos de AWS CLI](#).

Especificación del punto de conexión de AWS CodeBuild (SDK de AWS)

Puede utilizar un SDK de AWS para especificar el punto de conexión a través del que se obtiene acceso a AWS CodeBuild. Aunque en este ejemplo se utiliza el [SDK de AWS para Java](#), puede especificar el punto de conexión con los demás SDK de AWS.

Utilice el método `withEndpointConfiguration` cuando cree el cliente `AWSCodeBuild`. Utilice el siguiente formato:

```
AWSCodeBuild awsCodeBuild = AWSCodeBuildClientBuilder.standard().
    withEndpointConfiguration(new AwsClientBuilder.EndpointConfiguration("endpoint",
    "region")).
    withCredentials(new AWSStaticCredentialsProvider(sessionCredentials)).
    build();
```

Para obtener información sobre `AWSCodeBuildClientBuilder`, consulte [Clase AWSCodeBuildClientBuilder](#).

Las credenciales utilizadas en `withCredentials` deben ser del tipo `AWSCredentialsProvider`. Para obtener más información, consulte [Uso de credenciales de AWS](#).

No incluya el código `https://` al principio del punto de conexión.

Si desea especificar un punto de conexión que no sea de FIPS, puede utilizar la región en lugar del punto de conexión propiamente dicho. Por ejemplo, para especificar el punto de conexión de la región Este de EE. UU. (Norte de Virginia), puede utilizar `us-east-1` en lugar del nombre completo del punto de conexión, `codebuild.us-east-1.amazonaws.com`.

Si desea especificar un punto de conexión de FIPS, puede utilizar un alias para simplificar el código. Solo los puntos de enlace de FIPS disponen de un alias. El resto de puntos de enlace se deben especificar utilizando la región o el nombre completo.

En la tabla siguiente se muestra el alias para cada uno de los cuatro puntos de enlace de FIPS disponibles:

Nombre de la región	Región	Punto de conexión	Alias
US East (N. Virginia)	us-east-1	codebuild-fips.us-east-1.amazonaws.com	us-east-1-fips
US East (Ohio)	us-east-2	codebuild-fips.us-east-2.amazonaws.com	us-east-2-fips
EE. UU. Oeste (Norte de California)	us-west-1	codebuild-fips.us-west-1.amazonaws.com	us-west-1-fips
EE. UU. Oeste (Oregon)	us-west-2	codebuild-fips.us-west-2.amazonaws.com	us-west-2-fips

Para especificar el uso del punto de conexión FIPS de la región Oeste de EE. UU. (Oregón) utilizando un alias:

```
AWSCodeBuild awsCodeBuild = AWSCodeBuildClientBuilder.standard().
    withEndpointConfiguration(new AwsClientBuilder.EndpointConfiguration("us-west-2-
fips", "us-west-2")).
    withCredentials(new AWSStaticCredentialsProvider(sessionCredentials)).
    build();
```

Para especificar el uso del punto de conexión no FIPS de la región Este de EE. UU. (Norte de Virginia):

```
AWSCodeBuild awsCodeBuild = AWSCodeBuildClientBuilder.standard().
    withEndpointConfiguration(new AwsClientBuilder.EndpointConfiguration("us-east-1",
"us-east-1")).
    withCredentials(new AWSStaticCredentialsProvider(sessionCredentials)).
    build();
```

Para especificar el uso del punto de conexión no FIPS de la región Asia-Pacífico (Bombay):

```
AWSCodeBuild awsCodeBuild = AWSCodeBuildClientBuilder.standard().
    withEndpointConfiguration(new AwsClientBuilder.EndpointConfiguration("ap-south-1",
"ap-south-1")).
    withCredentials(new AWSStaticCredentialsProvider(sessionCredentials)).
    build();
```

Usar AWS CodePipeline con AWS CodeBuild para probar el código y ejecutar compilaciones

Puede automatizar el proceso de lanzamiento utilizando AWS CodePipeline para probar el código y ejecutar las compilaciones con AWS CodeBuild.

En la siguiente tabla se muestran las tareas y los métodos disponibles que se pueden realizar. El uso de los SDK de AWS para realizar estas tareas queda fuera del ámbito de este tema.

Tarea	Enfoques disponibles	Enfoques descritos en este tema
Cree una canalización de entrega continua (CD) CodePipeline que automatice las compilaciones con CodeBuild	<ul style="list-style-type: none"> CodePipeline consola AWS CLI SDK de AWS 	<ul style="list-style-type: none"> Usar la consola de CodePipeline Usar AWS CLI Puede adaptar la información de este tema para utilizar los SDK de AWS. Para obtener más información, consulte la documentación de la acción <code>create-pipeline</code> adecuada para su lenguaje de programación en la sección SDK de Herramientas de Amazon Web Services o consulte CreatePipeline en la Referencia de la API de AWS CodePipeline.
Añada la automatización, pruebe y	<ul style="list-style-type: none"> CodePipeline consola AWS CLI 	<ul style="list-style-type: none"> Utilice la CodePipeline consola para añadir la automatización de la compilación

Tarea	Enfoques disponibles	Enfoques descritos en este tema
desarrolle con CodeBuild ella a una canalización existente en CodePipeline	<ul style="list-style-type: none"> • SDK de AWS 	<ul style="list-style-type: none"> • Utilice la CodePipeline consola para añadir la automatización de pruebas • Para elloAWS CLI, puedes adaptar la información de este tema para crear una canalización que contenga una acción de CodeBuild construcción o una acción de prueba. Para obtener más información, consulte Editar una canalización (AWS CLI) y la referencia sobre la estructura de la CodePipeline canalización en la Guía del AWS CodePipeline usuario. • Puede adaptar la información de este tema para utilizar los SDK de AWS. Para obtener más información, consulte la documentación de la acción <code>update-pipeline</code> adecuada para su lenguaje de programación en la sección SDK de Herramientas de Amazon Web Services o consulte UpdatePipeline en la Referencia de la API de AWS CodePipeline.

Requisitos previos

1. Responda a las preguntas de [Planificación de una compilación](#).
2. Si utiliza un usuario para acceder CodePipeline en lugar de una cuenta AWS raíz o un usuario administrador, adjunte la política gestionada nombrada `AWSCodePipelineFullAccess` al usuario (o al grupo de IAM al que pertenece el usuario). No se recomienda usar una cuenta raíz de AWS. Esta política concede al usuario permiso para crear la canalización en CodePipeline. Para obtener más información, consulte [Asociar políticas administradas](#) en la Guía del usuario.

Note

La entidad de IAM que asocia la política al usuario (o al grupo de IAM al que pertenece el usuario) debe tener permiso en IAM para asociar políticas. Para obtener más información, consulte [Delegación de permisos para administrar usuarios, grupos y credenciales de IAM](#) en la Guía del usuario.

3. Cree un rol de CodePipeline servicio si aún no tiene uno disponible en su AWS cuenta. CodePipeline utiliza este rol de servicio para interactuar con otros AWS servicios AWS CodeBuild, incluso en su nombre. Por ejemplo, para usar la función de servicio AWS CLI para crear una función CodePipeline de servicio, ejecute el `create-role` comando de IAM:

Para Linux, macOS o Unix:

```
aws iam create-role --role-name AWS-CodePipeline-CodeBuild-Service-Role
--assume-role-policy-document '{"Version":"2012-10-17","Statement":
{"Effect":"Allow","Principal":
{"Service":"codepipeline.amazonaws.com"},"Action":"sts:AssumeRole"}}'
```

Para Windows:

```
aws iam create-role --role-name AWS-CodePipeline-CodeBuild-Service-Role --assume-
role-policy-document "{\"Version\":\"2012-10-17\",\"Statement\":{\"Effect\":
\"Allow\",\"Principal\":{\"Service\":\"codepipeline.amazonaws.com\"},\"Action\":
\"sts:AssumeRole\"}}"
```

Note

La entidad de IAM que crea esta función de CodePipeline servicio debe tener permiso en IAM para crear funciones de servicio.

4. Tras crear una función de CodePipeline servicio o identificar una existente, debe añadir la política de función de CodePipeline servicio predeterminada a la función de servicio, tal como se describe en la sección [Revise la política de función de CodePipeline servicio predeterminada](#) de la Guía del AWS CodePipeline usuario, si aún no forma parte de la política de la función.

Note

La entidad de IAM que añada esta política de funciones de CodePipeline servicio debe tener permiso en IAM para añadir políticas de funciones de servicio a las funciones de servicio.

5. Cree y cargue el código fuente en un tipo de repositorio compatible con CodeBuild y CodePipeline, por ejemplo CodeCommit, Amazon S3, Bitbucket o GitHub. El código fuente debe contener un archivo `buildspec`, pero puede declarar uno cuando defina un proyecto de

compilación más adelante en este tema. Para obtener más información, consulte [Referencia de la especificación de compilación](#).

⚠ Important

Si tiene previsto utilizar la canalización para implementar el código fuente compilado, el artefacto de salida de la compilación debe ser compatible con el sistema de implementación que utilice.

- Para elloAWS OpsWorks, consulte el [código fuente de la aplicación](#) y su [uso CodePipeline con AWS OpsWorks](#) en la Guía del AWS OpsWorks usuario.

Temas

- [Crear una canalización que use CodeBuild \(consola de CodePipeline\)](#)
- [Crear una canalización que use CodeBuild \(AWS CLI\)](#)
- [Agregar una acción de compilación de CodeBuild a una canalización \(consola de CodePipeline\)](#)
- [Agregar una acción de prueba de CodeBuild a una canalización \(consola de CodePipeline\)](#)

Crear una canalización que use CodeBuild (consola de CodePipeline)

Utilice el procedimiento siguiente para crear una canalización que utilice CodeBuild para compilar e implementar el código fuente.

Para crear una canalización que solo pruebe el código fuente:

- Utilice el siguiente procedimiento para crear la canalización y, a continuación, elimine las fases Build y Beta de la compilación. A continuación, use el procedimiento [Agregar una acción de prueba de CodeBuild a una canalización \(consola de CodePipeline\)](#) de este tema para añadir a la canalización una acción de prueba que use CodeBuild.
- Utilice uno de los otros procedimientos de este tema para crear la canalización y, a continuación, use el procedimiento [Agregar una acción de prueba de CodeBuild a una canalización \(consola de CodePipeline\)](#) de este tema para añadir a la canalización una acción de prueba que use CodeBuild.

Para utilizar el asistente para crear canalizaciones de CodePipeline para crear una canalización que use CodeBuild

1. Inicie sesión en AWS Management Console utilizando:

- Su cuenta raíz de AWS. No se recomienda. Para obtener más información, consulte [El usuario raíz de la cuenta](#) en la Guía del usuario.
- Un usuario administrador en su cuenta de AWS. Para obtener más información, consulte [Creación del primer usuario raíz y grupo de Cuenta de AWS](#) en la Guía del usuario.
- Un usuario de su cuenta de AWS con permiso para utilizar el conjunto mínimo de acciones siguiente:

```
codepipeline:*
iam:ListRoles
iam:PassRole
s3:CreateBucket
s3:GetBucketPolicy
s3:GetObject
s3:ListAllMyBuckets
s3:ListBucket
s3:PutBucketPolicy
codecommit:ListBranches
codecommit:ListRepositories
codedeploy:GetApplication
codedeploy:GetDeploymentGroup
codedeploy:ListApplications
codedeploy:ListDeploymentGroups
elasticbeanstalk:DescribeApplications
elasticbeanstalk:DescribeEnvironments
lambda:GetFunctionConfiguration
lambda:ListFunctions
opsworks:DescribeStacks
opsworks:DescribeApps
opsworks:DescribeLayers
```

2. Abra la consola de AWS CodePipeline en <https://console.aws.amazon.com/codesuite/codepipeline/home>.
3. En el selector de regiones de AWS, elija la región de AWS en la que quiere que se alojen los recursos de AWS del proyecto de compilación. Debe ser una región de AWS en la que se admita CodeBuild. Para obtener más información, consulte [AWS CodeBuild](#) en la Referencia general de Amazon Web Services.

4. Cree una canalización. Si aparece una página con información de CodePipeline, seleccione Crear canalización. Si aparece la página Pipelines (Canalizaciones), seleccione Create pipeline (Crear canalización).
5. En la página Step 1: Choose pipeline settings (Paso 1: Elegir configuración de canalización), en Pipeline name (Nombre de canalización), escriba un nombre para la canalización (por ejemplo, **CodeBuildDemoPipeline**). Si elige otro nombre, asegúrese de utilizarlo durante todo este procedimiento.
6. En Role name (Nombre de la función), realice una de las operaciones siguientes:

Seleccione New service role (Nueva función de servicio) y, en Role Name (Nombre de la función), especifique un nombre para la nueva función de servicio.

Elija Rol de servicio existente y elija después el rol de servicio de CodePipeline que haya creado o identificado como parte de los requisitos previos de este tema.

7. En Artifact store (Almacén de artefactos), haga lo siguiente:
 - Seleccione Ubicación predeterminada para utilizar el almacén de artefactos predeterminado (por ejemplo, el bucket de artefactos de S3 designado como predeterminado) con la canalización en la región de AWS que haya seleccionado para la canalización.
 - Elija Ubicación personalizada si ya dispone de un almacén de artefactos existente como, por ejemplo, un bucket de artefactos de S3, en la misma región de AWS que la canalización.

Note

Esto no es el bucket de código fuente del código fuente de la canalización. Este es el almacén de artefactos de la canalización. Se necesita un almacén de artefactos diferente, como un bucket de S3, para cada canalización y debe estar en la misma región de AWS que la canalización.

8. Elija Next (Siguiendo).
9. En la página Step 2: Add source stage (Paso 2: Agregar una fase de origen), para Source provider (Proveedor de código fuente), proceda del modo siguiente:
 - Si el código fuente está almacenado en un bucket de S3, elija Amazon S3. En Bucket, seleccione el bucket de S3 que contiene el código fuente. En S3 object key (Clave de objeto de S3), escriba el nombre del archivo que contiene el código fuente (por ejemplo, *file-name.zip*). Elija Next (Siguiendo).

- Si el código fuente está almacenado en un repositorio de AWS CodeCommit, seleccione CodeCommit. En Repository name, elija el nombre del repositorio que contiene el código fuente. En Branch name (Nombre de la ramificación), elija el nombre de la ramificación que contiene la versión del código fuente que desea compilar. Elija Next (Siguiente).
- Si el código fuente está almacenado en un repositorio GitHub, elija GitHub. Elija Connect to GitHub y siga las instrucciones para autenticarse con GitHub. En Repository, elija el nombre del repositorio que contiene el código fuente. En Branch (Ramificación), elija el nombre de la ramificación que contiene la versión del código fuente que desea compilar.

Elija Next (Siguiente).

10. En la página Paso 3: Agregar una fase de compilación, en Proveedor de compilación, elija CodeBuild.
11. Si ya tiene un proyecto de compilación que desee utilizar, en Nombre del proyecto, seleccione el nombre del proyecto de compilación y vaya directamente al paso siguiente de este procedimiento.

Si necesita crear un nuevo proyecto de compilación de CodeBuild, siga las instrucciones que se indican en [Creación de un proyecto de compilación \(consola\)](#) y vuelva a este procedimiento.

Si elige un proyecto de compilación existente, debe haber definido ya la configuración de artefactos de salida de la compilación (aunque CodePipeline la anule). Para obtener más información, consulte [Cambiar la configuración de un proyecto de compilación \(consola\)](#).

Important

Si se habilitan webhooks para un proyecto de CodeBuild y el proyecto se usa como un paso de compilación en CodePipeline, se crearán dos compilaciones idénticas para cada confirmación. Una compilación se activa a través de los webhooks y la otra a través de CodePipeline. Como se factura por compilación, se le cobrarán ambas compilaciones. Por lo tanto, si utiliza CodePipeline, le recomendamos que deshabilite los webhooks en CodeBuild. En la consola de AWS CodeBuild, desactive la casilla Webhook. Para obtener más información, consulte [Cambiar la configuración de un proyecto de compilación \(consola\)](#).

12. En la página Step 4: Add deploy stage (Paso 4: Agregar una fase de implementación), realice una de las siguientes operaciones:

- Si no desea implementar el artefacto de salida de la compilación, elija Skip (Omitir) y confirme esta opción cuando se lo soliciten.
- Si desea implementar el artefacto de salida de la compilación, en Deploy provider (Proveedor de implementación), elija un proveedor de implementación y especifique la configuración cuando se lo soliciten.

Elija Next (Siguiente).

13. En la página Review (Revisar), revise las opciones y seleccione Create pipeline (Crear canalización).
14. Una vez que la canalización se ejecute correctamente, podrá obtener el artefacto de salida de la compilación. Con la canalización mostrada en la consola de CodePipeline, en la acción Compilar, seleccione la ayuda contextual. Anote el valor de Output artifact (por ejemplo, MyAppBuild).

Note

También puede obtener el artefacto de salida de la compilación eligiendo el enlace Artefactos de compilación en la página de detalles de la compilación de la consola de CodeBuild. Para ir a esta página, omita el resto de los pasos de este procedimiento y consulte [Ver detalles de las compilaciones \(consola\)](#).

15. Abra la consola de Amazon S3 en <https://console.aws.amazon.com/s3>.
16. En la lista de buckets, abra el bucket utilizado por la canalización. El nombre del bucket debe seguir el formato `codepipeline-region-ID-random-number`. Puede utilizar la AWS CLI para ejecutar el comando `get-pipeline` de CodePipeline para obtener el nombre del bucket, donde *my-pipeline-name* es el nombre para mostrar de la canalización:

```
aws codepipeline get-pipeline --name my-pipeline-name
```

En el resultado, el objeto `pipeline` contiene un objeto `artifactStore`, que a su vez contiene un valor `location` con el nombre del bucket.

17. Abra la carpeta que tiene el mismo nombre que la canalización (en función de la longitud del nombre de la canalización, el nombre de la carpeta podría aparecer truncado) y después abra la carpeta que tiene el mismo nombre que el valor de Output artifact (Artefacto de salida) que anotó anteriormente.

18. Extraiga el contenido del archivo . Si hay varios archivos en esa carpeta, extraiga el contenido del archivo con la última marca temporal Last Modified. (Es posible que tenga que asignar al archivo la extensión .zip para que pueda trabajar con él en la utilidad ZIP del sistema). El artefacto de salida de la compilación está en el contenido extraído del archivo.
19. Si ha indicado a CodePipeline que implementara el artefacto de salida de la compilación, utilice las instrucciones del proveedor de implementación para obtener el artefacto de salida de la compilación en los destinos de la implementación.

Crear una canalización que use CodeBuild (AWS CLI)

Utilice el siguiente procedimiento para crear una canalización que utilice CodeBuild para compilar el código fuente.

Para utilizar la AWS CLI para crear una canalización que implemente el código fuente compilado que solo pruebe el código fuente, puede adaptar las instrucciones de [Editar una canalización \(AWS CLI\)](#) y [Referencia de estructura de la canalización](#) en la Guía del usuario de AWS CodePipeline.

1. Cree o identifique un proyecto de compilación en CodeBuild. Para obtener más información, consulte [Creación de un proyecto de compilación](#).

Important

El proyecto de compilación debe definir la configuración del artefacto de salida de la compilación (aunque CodePipeline la invalide). Para obtener más información, consulte la descripción de artifacts en [Crear un proyecto de compilación \(AWS CLI\)](#).

2. Asegúrese de que ha configurado la AWS CLI con la clave de acceso de AWS y la clave de acceso secreta de AWS correspondientes a una de las entidades de IAM descritas en este tema. Para obtener más información, consulte [Configuración inicial de la AWS Command Line Interface](#) en la Guía del usuario de AWS Command Line Interface.
3. Crear un archivo con formato JSON que represente la estructura de la canalización. Asigne al archivo el nombre `create-pipeline.json` o uno similar. Por ejemplo, esta estructura con formato JSON crea una canalización con una acción de origen que hace referencia a un bucket de entrada de S3 y una acción de compilación que utiliza CodeBuild:

```
{  
  "pipeline": {
```

```
"roleArn": "arn:aws:iam::<account-id>:role/<AWS-CodePipeline-service-role-
name>",
"stages": [
  {
    "name": "Source",
    "actions": [
      {
        "inputArtifacts": [],
        "name": "Source",
        "actionTypeId": {
          "category": "Source",
          "owner": "AWS",
          "version": "1",
          "provider": "S3"
        },
        "outputArtifacts": [
          {
            "name": "MyApp"
          }
        ],
        "configuration": {
          "S3Bucket": "<bucket-name>",
          "S3ObjectKey": "<source-code-file-name.zip>"
        },
        "runOrder": 1
      }
    ]
  },
  {
    "name": "Build",
    "actions": [
      {
        "inputArtifacts": [
          {
            "name": "MyApp"
          }
        ],
        "name": "Build",
        "actionTypeId": {
          "category": "Build",
          "owner": "AWS",
          "version": "1",
          "provider": "CodeBuild"
        }
      }
    ]
  }
]
```

```
        "outputArtifacts": [
          {
            "name": "default"
          }
        ],
        "configuration": {
          "ProjectName": "<build-project-name>"
        },
        "runOrder": 1
      }
    ]
  },
  "artifactStore": {
    "type": "S3",
    "location": "<CodePipeline-internal-bucket-name>"
  },
  "name": "<my-pipeline-name>",
  "version": 1
}
```

En estos datos con formato JSON:

- El valor de `roleArn` debe coincidir con el ARN del rol de servicio de CodeBuild que ha creado o identificado como parte de los requisitos previos.
- Los valores de `S3Bucket` y `S3ObjectKey` en `configuration` presuponen que el código fuente está almacenado en un bucket de S3. Para obtener información sobre la configuración de otros tipos de repositorios de código fuente, consulte [Referencia de la estructura de la canalización de CodePipeline](#) en la Guía del usuario de AWS CodePipeline.
- El valor de `ProjectName` es el nombre del proyecto de compilación de CodeBuild que ha creado anteriormente en este procedimiento.
- El valor de `location` es el nombre del bucket de S3 que se utiliza en esta canalización. Para obtener más información, consulte este artículo acerca de [cómo crear una política para utilizar un bucket de S3 como almacén de artefactos de CodePipeline](#) en la Guía del usuario de AWS CodePipeline.
- El valor de `name` es el nombre de esta canalización. Todos los nombres de canalización deben ser únicos en su cuenta.

Aunque estos datos describen únicamente una acción de origen y una acción de compilación, puede agregar acciones para actividades relacionadas con las pruebas, la implementación del artefacto de salida de la compilación, las llamadas a funciones de AWS Lambda, etc. Para obtener más información, consulte [Referencia de la estructura de la canalización de AWS CodePipeline](#) en la Guía del usuario de AWS CodePipeline.

4. Cambie a la carpeta que contiene el archivo JSON y, a continuación, ejecute el comando [create-pipeline](#) de CodePipeline, especificando el nombre de archivo:

```
aws codepipeline create-pipeline --cli-input-json file://create-pipeline.json
```

Note

Debe crear la canalización en una región de AWS en la que se admita CodeBuild. Para obtener más información, consulte [AWS CodeBuild](#) en la Referencia general de Amazon Web Services.

Los datos con formato JSON aparecen en la salida y CodePipeline crea la canalización.

5. Para obtener información sobre el estado de la canalización, ejecute el comando [get-pipeline-state](#) de CodePipeline, especificando el nombre de la canalización:

```
aws codepipeline get-pipeline-state --name <my-pipeline-name>
```

En el resultado, busque información que confirme que la compilación se ha realizado correctamente. Los puntos suspensivos (. . .) se utilizan para mostrar los datos que se han omitido por motivos de brevedad.

```
{
  ...
  "stageStates": [
    ...
    {
      "actionStates": [
        {
          "actionName": "CodeBuild",
          "latestExecution": {
```

```
        "status": "SUCCEEDED",
        ...
    },
    ...
}
]
}
]
}
```

Si ejecuta este comando demasiado pronto, es posible que no vea ninguna información sobre la acción de compilación. Tal vez tenga que ejecutar este comando varias veces hasta que la canalización termine de ejecutar la acción de compilación.

- Una vez que la compilación se realice correctamente, siga estas instrucciones para obtener el artefacto de salida de la compilación. Abra la consola de Amazon S3 en <https://console.aws.amazon.com/s3>.

Note

También puede obtener el artefacto de salida de la compilación eligiendo el enlace Artefactos de compilación en la página de detalles de la compilación relacionada de la consola de CodeBuild. Para ir a esta página, omita el resto de los pasos de este procedimiento y consulte [Ver detalles de las compilaciones \(consola\)](#).

- En la lista de buckets, abra el bucket utilizado por la canalización. El nombre del bucket debe seguir el formato `codepipeline-<region-ID>-<random-number>`. Puede obtener el nombre del bucket del archivo `create-pipeline.json` o puede ejecutar el comando `get-pipeline` de CodePipeline para obtener el nombre del bucket.

```
aws codepipeline get-pipeline --name <pipeline-name>
```

En el resultado, el objeto `pipeline` contiene un objeto `artifactStore`, que a su vez contiene un valor `location` con el nombre del bucket.

- Abra la carpeta que coincide con el nombre de la canalización (por ejemplo, *<pipeline-name>*).
- En esa carpeta, abra la carpeta con el nombre `default`.

10. Extraiga el contenido del archivo . Si hay varios archivos en esa carpeta, extraiga el contenido del archivo con la última marca temporal Last Modified. (Es posible que tenga que asignar al archivo una extensión .zip para que pueda trabajar con él en la utilidad ZIP del sistema). El artefacto de salida de la compilación está en el contenido extraído del archivo.

Agregar una acción de compilación de CodeBuild a una canalización (consola de CodePipeline)

1. Inicie sesión en AWS Management Console utilizando:

- Su cuenta raíz de AWS. No se recomienda. Para obtener más información, consulte [El usuario raíz de la cuenta](#) en la Guía del usuario.
- Un usuario administrador en su cuenta de AWS. Para obtener más información, consulte [Creación del primer usuario raíz y grupo de Cuenta de AWS](#) en la Guía del usuario.
- Un usuario en su cuenta de AWS con permiso para realizar el conjunto mínimo de acciones siguiente:

```
codepipeline:*
iam:ListRoles
iam:PassRole
s3:CreateBucket
s3:GetBucketPolicy
s3:GetObject
s3:ListAllMyBuckets
s3:ListBucket
s3:PutBucketPolicy
codecommit:ListBranches
codecommit:ListRepositories
codedeploy:GetApplication
codedeploy:GetDeploymentGroup
codedeploy:ListApplications
codedeploy:ListDeploymentGroups
elasticbeanstalk:DescribeApplications
elasticbeanstalk:DescribeEnvironments
lambda:GetFunctionConfiguration
lambda:ListFunctions
opsworks:DescribeStacks
opsworks:DescribeApps
opsworks:DescribeLayers
```

- Abra la consola de CodePipeline en <https://console.aws.amazon.com/codesuite/codepipeline/home>.
- En el selector de regiones de AWS, elija la región de AWS en la que se encuentra la canalización. Debe ser una región en la que se admita CodeBuild. Para obtener más información, consulte [CodeBuild](#) en la Referencia general de Amazon Web Services.
- En la página Pipelines (Canalizaciones), seleccione el nombre de la canalización.
- En la página de detalles de la canalización, en la acción Source (Origen), elija la ayuda contextual. Anote el valor de Output artifact (Artefacto de salida) (por ejemplo, MyApp).

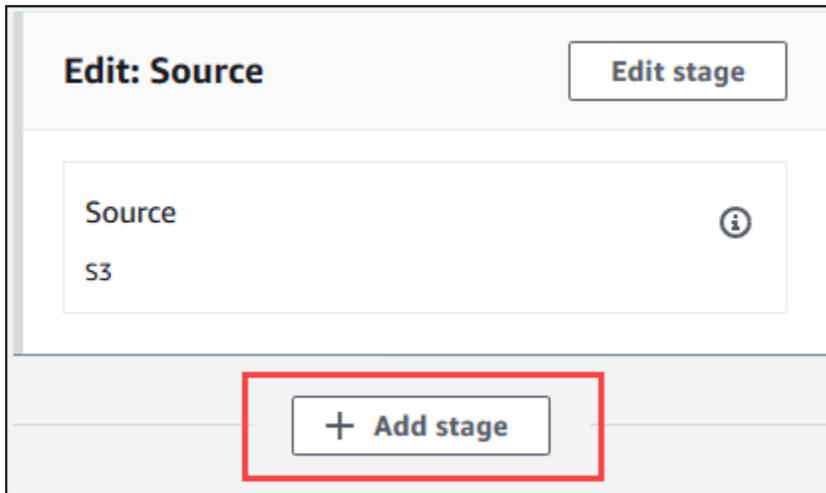
 Note

En este procedimiento se indica cómo añadir una acción de compilación en una fase de compilación entre las fases Source y Beta. Si desea añadir la acción de compilación en otro lugar, elija la ayuda contextual sobre la acción justo antes del lugar en el que desea añadir la acción de compilación y anote el valor de Output artifact (Artefacto de salida).

- Elija Edit (Editar).
- Entre las etapas Source (Origen) y Beta, seleccione Add stage (Añadir fase).

 Note

En este procedimiento, se explica cómo se agrega una etapa de compilación entre las etapas Source y Beta. Para añadir una acción de compilación a una fase existente, seleccione Edit stage (Editar fase) en la fase y vaya al paso 8 de este procedimiento. Para añadir la fase de compilación en otra parte, seleccione Add stage (Añadir fase) en el lugar que desee.



8. En Stage name (Nombre de fase), escriba el nombre de la fase de compilación (por ejemplo, **Build**). Si elige otro nombre, úselo en todo este procedimiento.
9. Dentro de la fase seleccionada, seleccione Add action (Añadir acción).

Note

En este procedimiento se muestra cómo agregar la acción de compilación dentro de una fase de compilación. Para añadir la acción de compilación en otra parte, seleccione Add action (Añadir acción) en el lugar que desee. Es posible que primero tenga que seleccionar Edit stage (Editar fase) en la fase existente en la que desea añadir la acción de compilación.

10. En Edit action (Editar acción), en Action name (Nombre de acción), escriba un nombre para la acción (por ejemplo, **CodeBuild**). Si elige otro nombre, úselo en todo este procedimiento.
11. Como Proveedor de acciones, elija CodeBuild.
12. Si ya tiene un proyecto de compilación que desee utilizar, en Nombre del proyecto, seleccione el nombre del proyecto de compilación y vaya directamente al paso siguiente de este procedimiento.

Si necesita crear un nuevo proyecto de compilación de CodeBuild, siga las instrucciones que se indican en [Creación de un proyecto de compilación \(consola\)](#) y vuelva a este procedimiento.

Si elige un proyecto de compilación existente, debe haber definido ya la configuración de artefactos de salida de la compilación (aunque CodePipeline la anule). Para obtener más

información, consulte la descripción de artefactos en [Creación de un proyecto de compilación \(consola\)](#) o [Cambiar la configuración de un proyecto de compilación \(consola\)](#).

 Important

Si se habilitan webhooks para un proyecto de CodeBuild y el proyecto se usa como un paso de compilación en CodePipeline, se crearán dos compilaciones idénticas para cada confirmación. Una compilación se activa a través de los webhooks y la otra a través de CodePipeline. Como se factura por compilación, se le cobrarán ambas compilaciones. Por lo tanto, si utiliza CodePipeline, le recomendamos que deshabilite los webhooks en CodeBuild. En la consola de CodeBuild, desactive la casilla Webhook. Para obtener más información, consultar [Cambiar la configuración de un proyecto de compilación \(consola\)](#)

13. En Input artifacts (Artefactos de entrada), elija el artefacto de salida que anotó anteriormente en este procedimiento.
14. En Output artifacts (Artefactos de salida), escriba un nombre para el artefacto de salida (por ejemplo, **MyAppBuild**).
15. Elija Añadir acción.
16. Elija Save (Guardar) y después otra vez Save (Guardar) para guardar los cambios realizados en la canalización.
17. Elija Release change.
18. Una vez que la canalización se ejecute correctamente, podrá obtener el artefacto de salida de la compilación. Con la canalización mostrada en la consola de CodePipeline, en la acción Compilar, seleccione la ayuda contextual. Anote el valor de Output artifact (por ejemplo, MyAppBuild).

 Note

También puede obtener el artefacto de salida de la compilación eligiendo el enlace Artefactos de compilación en la página de detalles de la compilación de la consola de CodeBuild. Para tener acceso a esta página, consulte [Ver detalles de las compilaciones \(consola\)](#) y vaya al paso 31 de este procedimiento.

19. Abra la consola de Amazon S3 en <https://console.aws.amazon.com/s3>.

20. En la lista de buckets, abra el bucket utilizado por la canalización. El nombre del bucket debe seguir el formato `codepipeline-region-ID-random-number`. Puede utilizar la AWS CLI para ejecutar el comando de CodeBuild `get-pipeline` para obtener el nombre del bucket:

```
aws codepipeline get-pipeline --name my-pipeline-name
```

En el resultado, el objeto `pipeline` contiene un objeto `artifactStore`, que a su vez contiene un valor `location` con el nombre del bucket.

21. Abra la carpeta que tiene el mismo nombre que la canalización (en función de la longitud del nombre de la canalización, el nombre de la carpeta podría aparecer truncado) y después abra la carpeta que tiene el mismo nombre que el valor de `Output artifact` (Artefacto de salida) que anotó anteriormente en este procedimiento.
22. Extraiga el contenido del archivo `.`. Si hay varios archivos en esa carpeta, extraiga el contenido del archivo con la última marca temporal `Last Modified`. (Es posible que tenga que asignar al archivo la extensión `.zip` para que pueda trabajar con él en la utilidad `ZIP` del sistema). El artefacto de salida de la compilación está en el contenido extraído del archivo.
23. Si ha indicado a CodePipeline que implementara el artefacto de salida de la compilación, utilice las instrucciones del proveedor de implementación para obtener el artefacto de salida de la compilación en los destinos de la implementación.

Agregar una acción de prueba de CodeBuild a una canalización (consola de CodePipeline)

1. Inicie sesión en AWS Management Console utilizando:
 - Su cuenta raíz de AWS. No se recomienda. Para obtener más información, consulte [El usuario raíz de la cuenta](#) en la Guía del usuario.
 - Un usuario administrador en su cuenta de AWS. Para obtener más información, consulte [Creación del primer usuario raíz y grupo de Cuenta de AWS](#) en la Guía del usuario.
 - Un usuario en su cuenta de AWS con permiso para realizar el conjunto mínimo de acciones siguiente:

```
codepipeline:*
iam:ListRoles
iam:PassRole
s3:CreateBucket
```

```
s3:GetBucketPolicy
s3:GetObject
s3:ListAllMyBuckets
s3:ListBucket
s3:PutBucketPolicy
codecommit:ListBranches
codecommit:ListRepositories
codedeploy:GetApplication
codedeploy:GetDeploymentGroup
codedeploy:ListApplications
codedeploy:ListDeploymentGroups
elasticbeanstalk:DescribeApplications
elasticbeanstalk:DescribeEnvironments
lambda:GetFunctionConfiguration
lambda:ListFunctions
opsworks:DescribeStacks
opsworks:DescribeApps
opsworks:DescribeLayers
```

2. Abra la consola de CodePipeline en <https://console.aws.amazon.com/codesuite/codepipeline/home>.
3. En el selector de regiones de AWS, elija la región de AWS en la que se encuentra la canalización. Debe ser una región de AWS en la que se admita CodeBuild. Para obtener más información, consulte [AWS CodeBuild](#) en la Referencia general de Amazon Web Services.
4. En la página Pipelines (Canalizaciones), seleccione el nombre de la canalización.
5. En la página de detalles de la canalización, en la acción Source (Origen), elija la ayuda contextual. Anote el valor de Output artifact (Artefacto de salida) (por ejemplo, MyApp).

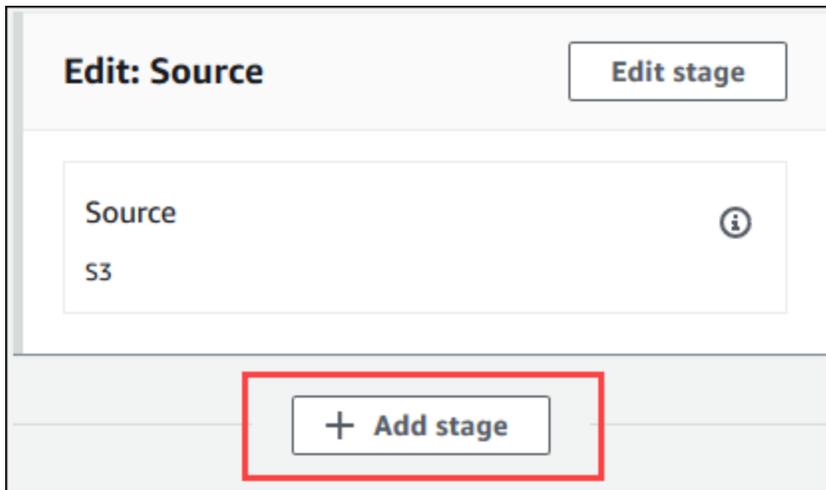
Note

En este procedimiento se indica cómo añadir una acción de prueba a una fase de prueba entre las fases Source y Beta. Si desea añadir la acción de prueba en otro lugar, coloque el puntero del ratón sobre la acción que aparece justo delante y anote el valor de Output artifact.

6. Elija Edit (Editar).
7. Inmediatamente después de la fase Source (Origen), seleccione Add stage (Añadir fase).

Note

En este procedimiento, se muestra también como se agrega a la canalización una etapa de prueba inmediatamente después de la fase Source. Para añadir una acción de prueba a una fase existente, seleccione Edit stage (Editar fase) en la fase y vaya al paso 8 de este procedimiento. Para añadir la fase de prueba en otra parte, seleccione Add stage (Añadir fase) en el lugar que desee.



8. En Stage name (Nombre de fase), escriba el nombre de la fase de prueba (por ejemplo, **Test**). Si elige otro nombre, úselo en todo este procedimiento.
9. En la fase seleccionada, elija Add action (Añadir acción).

Note

En este procedimiento se indica cómo añadir la acción de prueba en una fase de prueba. Para añadir la acción de prueba en otra parte, seleccione Add action (Añadir acción) en el lugar que desee. Es posible que primero tenga que seleccionar Edit (Editar) en la fase existente en la que desea añadir la acción de prueba.

10. En Edit action (Editar acción), en Action name (Nombre de acción), escriba un nombre para la acción (por ejemplo, **Test**). Si elige otro nombre, úselo en todo este procedimiento.
11. En Proveedor de acciones, en Prueba, elija CodeBuild.

12. Si ya tiene un proyecto de compilación que desee utilizar, en Nombre del proyecto, seleccione el nombre del proyecto de compilación y vaya directamente al paso siguiente de este procedimiento.

Si necesita crear un nuevo proyecto de compilación de CodeBuild, siga las instrucciones que se indican en [Creación de un proyecto de compilación \(consola\)](#) y vuelva a este procedimiento.

 Important

Si se habilitan webhooks para un proyecto de CodeBuild y el proyecto se usa como un paso de compilación en CodePipeline, se crearán dos compilaciones idénticas para cada confirmación. Una compilación se activa a través de los webhooks y la otra a través de CodePipeline. Como se factura por compilación, se le cobrarán ambas compilaciones. Por lo tanto, si utiliza CodePipeline, le recomendamos que deshabilite los webhooks en CodeBuild. En la consola de CodeBuild, desactive la casilla Webhook. Para obtener más información, consultar [Cambiar la configuración de un proyecto de compilación \(consola\)](#)

13. En Input artifacts (Artefactos de entrada), seleccione el valor de Output artifact (Artefacto de salida) que anotó anteriormente en este procedimiento.
14. (Opcional) Si desea que la acción de prueba produzca un artefacto de salida y configura la especificación de compilación en consecuencia, entonces en Output artifact (Artefacto de salida), escriba el valor que desea asignar al artefacto de salida.
15. Seleccione Save.
16. Elija Release change.
17. Una vez que la canalización se ejecute correctamente, obtendrá los resultados de prueba. En la fase Prueba de la canalización, elija el enlace CodeBuild para abrir la página del proyecto de compilación relacionada en la consola de CodeBuild.
18. En la página del proyecto de compilación, en el área Build history (Historial de compilaciones), elija el hipervínculo Build run (Ejecución de compilación) relacionado.
19. En la página de ejecución de la compilación, en Registros de compilación, elija el enlace Ver todo el registro para abrir el registro de compilación en la consola de Amazon CloudWatch.
20. Desplácese por el log de compilación para ver los resultados de la prueba.

Uso de AWS CodeBuild con Jenkins

Puede usar el complemento Jenkins de AWS CodeBuild para integrar CodeBuild con sus trabajos de compilación de Jenkins. En lugar de enviar los trabajos de compilación a los nodos de compilación de Jenkins, puede usar el complemento para enviar los trabajos de compilación a CodeBuild. De esta manera, ya no tendrá que aprovisionar, configurar ni administrar nodos de compilación de Jenkins.

Configuración de Jenkins

[Para obtener información sobre cómo configurar Jenkins con el complemento AWS CodeBuild y descargar el código fuente del complemento, consulte <https://github.com/aws-labs/aws-codebuild-jenkins-plugin>.](https://github.com/aws-labs/aws-codebuild-jenkins-plugin)

Instalación del complemento

Si ya ha configurado el servidor de Jenkins y solo desea instalar el complemento de AWS CodeBuild, en su instancia de Jenkins, en el administrador de complementos, busque **CodeBuild Plugin for Jenkins**.

Uso del complemento

Para utilizar AWS CodeBuild con orígenes que se encuentren fuera de una VPC

1. Cree un proyecto en la consola de CodeBuild. Para obtener más información, consulte [Creación de un proyecto de compilación \(consola\)](#).
 - Elija la región de AWS en la que desea ejecutar la compilación.
 - (Opcional) Establezca la configuración de Amazon VPC para permitir que el contenedor de compilaciones de CodeBuild obtenga acceso a los recursos de la VPC.
 - Anote el nombre de su proyecto. Lo necesitará en el paso 3.
 - (Opcional) Si su repositorio de fuentes no es compatible de forma nativa con CodeBuild, puede configurar Amazon S3 como el tipo de fuente de entrada del proyecto.
2. En la consola de IAM, cree el usuario para usarlo en el complemento Jenkins.
 - Cuando cree las credenciales del usuario, elija Programmatic Access.
 - Cree una política similar a la siguiente y, a continuación, asocie la política al usuario.

```
{
```

```

"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Resource": ["arn:aws:logs:{{region}}:{{awsAccountId}}:log-group:/aws/codebuild/{{projectName}}:*"],
    "Action": ["logs:GetLogEvents"]
  },
  {
    "Effect": "Allow",
    "Resource": ["arn:aws:s3:::{{inputBucket}}"],
    "Action": ["s3:GetBucketVersioning"]
  },
  {
    "Effect": "Allow",
    "Resource": ["arn:aws:s3:::{{inputBucket}}/{{inputObject}}"],
    "Action": ["s3:PutObject"]
  },
  {
    "Effect": "Allow",
    "Resource": ["arn:aws:s3:::{{outputBucket}}/*"],
    "Action": ["s3:GetObject"]
  },
  {
    "Effect": "Allow",
    "Resource": ["arn:aws:codebuild:{{region}}:{{awsAccountId}}:project/{{projectName}}"],
    "Action": ["codebuild:StartBuild",
      "codebuild:BatchGetBuilds",
      "codebuild:BatchGetProjects"]
  }
]
}

```

3. Cree un proyecto de estilo libre en Jenkins.

- En la página Configurar, elija Añadir paso de compilación y después Ejecutar compilación en CodeBuild.
- Configure el paso de compilación.
 - Proporcione valores para Region, Credentials y Project Name.
 - Elija Use Project source.
 - Guarde la configuración y ejecute una compilación desde Jenkins.

4. En Source Code Management, elija cómo desea recuperar el origen. Es posible que necesite instalar el complemento GitHub (o el complemento Jenkins del proveedor del repositorio de origen) en su servidor de Jenkins.
 - En la página Configure, elija Add build step y después Run build on AWS CodeBuild.
 - Configure el paso de compilación.
 - Proporcione valores para Region, Credentials y Project Name.
 - Elija Use Jenkins source.
 - Guarde la configuración y ejecute una compilación desde Jenkins.

Para utilizar el complemento de AWS CodeBuild con el complemento Pipeline de Jenkins

- En la página del proyecto de canalizaciones de Jenkins, utilice el generador de fragmentos para generar un script de canalización que añada CodeBuild como un paso de la canalización. Debe generar un script similar a este:

```
awsCodeBuild projectName: 'project', credentialsType: 'keys', region: 'us-west-2',
sourceControlType: 'jenkins'
```

Uso de AWS CodeBuild con Codecov

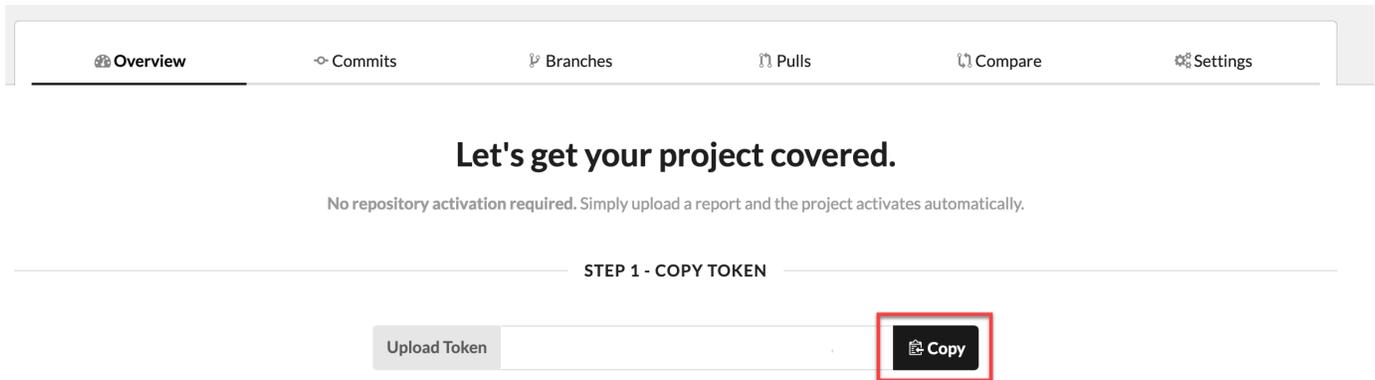
Codecov es una herramienta que mide la cobertura de las pruebas del código. Codecov identifica qué métodos e instrucciones del código no se han comprobado. Use los resultados para determinar dónde deben escribirse pruebas para mejorar la calidad del código. Codecov está disponible en tres de los repositorios de origen admitidos por CodeBuild: GitHub, GitHub Enterprise Server y Bitbucket. Si el proyecto de compilación usa GitHub Enterprise Server, debe utilizar Codecov Enterprise.

Cuando se ejecuta una compilación de un proyecto de CodeBuild integrado con Codecov, los informes de Codecov que analiza el código del repositorio se cargan en Codecov. Los registros de compilación incluyen un enlace a los informes. En este ejemplo, se muestra cómo se integra un proyecto de compilación de Python y Java con Codecov. Para ver una lista de los idiomas admitidos en Codecov, consulte [Codecov Supported Languages](#) en el sitio web de Codecov.

Integrar Codecov en un proyecto de compilación

Para integrar Codecov en un proyecto de compilación

1. Vaya a <https://codecov.io/signup> y regístrese en un repositorio de origen de GitHub o Bitbucket. Si usa GitHub Enterprise, consulte [Codecov Enterprise](#) en el sitio web de Codecov.
2. En Codecov, agregue el repositorio para el que desea cobertura.
3. Cuando se muestre la información del token, elija Copy (Copiar).



4. Agregue el token copiado como una variable de entorno llamada CODECOV_TOKEN al proyecto de compilación. Para obtener más información, consulte [Cambiar la configuración de un proyecto de compilación \(consola\)](#).
5. Cree un archivo de texto llamado `my_script.sh` en el repositorio. Copie lo siguiente en el archivo:

```
#!/bin/bash
bash <(curl -s https://codecov.io/bash) -t $CODECOV_TOKEN
```

6. Elija la pestaña Python o Java según corresponda en función de los usos del proyecto de compilación y siga estos pasos.

Java

1. Agregue el siguiente complemento de JacoCo a `pom.xml` en el repositorio.

```
<build>
  <plugins>
    <plugin>
      <groupId>org.jacoco</groupId>
      <artifactId>jacoco-maven-plugin</artifactId>
      <version>0.8.2</version>
```

```
<executions>
  <execution>
    <goals>
      <goal>prepare-agent</goal>
    </goals>
  </execution>
  <execution>
    <id>report</id>
    <phase>test</phase>
    <goals>
      <goal>report</goal>
    </goals>
  </execution>
</executions>
</plugin>
</plugins>
</build>
```

2. Especifique los siguientes comandos en el archivo buildspec. Para obtener más información, consulte [Sintaxis de buildspec](#).

```
build:
  - mvn test -f pom.xml -fn
postbuild:
  - echo 'Connect to CodeCov'
  - bash my_script.sh
```

Python

Especifique los siguientes comandos en el archivo buildspec. Para obtener más información, consulte [Sintaxis de buildspec](#).

```
build:
  - pip install coverage
  - coverage run -m unittest discover
postbuild:
  - echo 'Connect to CodeCov'
  - bash my_script.sh
```

7. Ejecute una compilación del proyecto. En los registros de compilación aparece un enlace a los informes de Codecov generados para el proyecto. Utilice este enlace para ver los informes de

Codecov. Para obtener más información, consulte [Ejecutar una compilación en AWS CodeBuild](#) y [Registrar llamadas a la API de AWS CodeBuild con AWS CloudTrail](#). La información de Codecov en los registros de compilación tiene el siguiente aspecto:

```
[Container] 2020/03/09 16:31:04 Running command bash my_script.sh
```

```

  ____  _
 / ____| | | |
| |  | | | | | | |
| |  | | | | | | |
| |  | | | | | | |
 \____|_|_|_|_|_|_|

```

```
Bash-20200303-bc4d7e6
```

```
·[0;90m==>·[0m AWS CodeBuild detected.
```

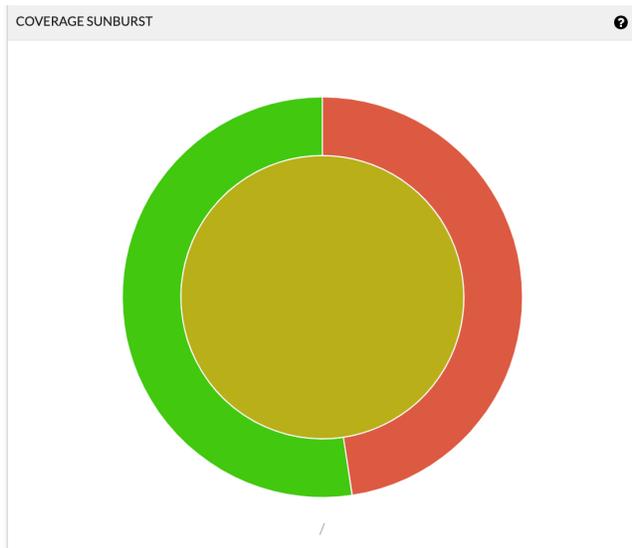
```
... The full list of Codecov log entries has been omitted for brevity ...
```

```
·
```

```
·[0;32m->·[0m View reports at ·[0;36mhttps://codecov.io/github/user/test\_py/commit/commit-id·[0m
```

```
[Container] 2020/03/09 16:31:07 Phase complete: POST_BUILD State: SUCCEEDED
```

Los informes tienen el siguiente aspecto:



Files	Files	Passed	Failed	Skipped	Coverage
<code>code.py</code>	10	7	0	3	70.00%
<code>tests.py</code>	11	11	0	0	100.00%
Project Totals (2 files)	21	18	0	3	85.71%

Utilizar AWS CodeBuild con aplicaciones sin servidor

AWS Serverless Application Model (AWS SAM) es un marco de trabajo de código abierto para crear aplicaciones sin servidor. Para obtener más información, consulte el repositorio del [modelo de aplicaciones sin servidor de AWS](#) en GitHub.

Es posible utilizar AWS CodeBuild para empaquetar e implementar aplicaciones sin servidor que sigan el estándar AWS SAM. En el paso de implementación, CodeBuild puede usar AWS CloudFormation. Para automatizar la compilación e implementación de aplicaciones sin servidor con CodeBuild y AWS CloudFormation, puede utilizar AWS CodePipeline.

Para obtener más información, consulte [Implementación de aplicaciones sin servidor](#) en la Guía para desarrolladores de AWS Serverless Application Model.

Recursos relacionados

- Para obtener información acerca de cómo empezar a trabajar con AWS CodeBuild, consulte [Empezar a trabajar con AWS CodeBuild utilizando la consola](#).

- Para obtener información acerca de cómo solucionar problemas en CodeBuild, consulte [Solución de problemas AWS CodeBuild](#).
- Para obtener información sobre las cuotas de CodeBuild, consulte [Cuotas para AWS CodeBuild](#).

Solución de problemas AWS CodeBuild

Utilice la información de este tema como ayuda para identificar, diagnosticar y resolver problemas. Para obtener información sobre cómo registrar y supervisar las CodeBuild compilaciones para solucionar problemas, consulte [Registro y monitorización](#).

Temas

- [Apache Maven crea artefactos de referencia en un repositorio incorrecto](#)
- [De forma predeterminada, los comandos de compilación se ejecutan como usuario raíz](#)
- [Las compilaciones pueden generar un error si los nombres de archivo contienen caracteres ajenos al inglés \(Estados Unidos\)](#)
- [Las compilaciones pueden fallar al obtener parámetros del almacén de parámetros de Amazon EC2](#)
- [No se puede obtener acceso al filtro de ramificaciones en la consola de CodeBuild](#)
- [No se puede ver si la compilación se ha realizado correctamente o no](#)
- [No se ha notificado el estado de la compilación al proveedor de fuentes](#)
- [No se puede encontrar ni seleccionar la imagen base de la plataforma Windows Server Core 2019](#)
- [Comandos anteriores de los archivos buildspec no reconocidos por comandos más recientes](#)
- [Error: "Access denied" \(Acceso denegado\) al intentar descargar la caché](#)
- [Aparece el error "BUILD_CONTAINER_UNABLE_TO_PULL_IMAGE" al usar una imagen de compilación personalizada](#)
- [Error: «Se encontró que el contenedor de compilación estaba inactivo antes de completar la compilación. El contenedor de compilación murió porque no tenía memoria suficiente o la imagen de Docker no es compatible. ErrorCode: 500»](#)
- [Error: "Cannot connect to the Docker daemon" \(No se puede conectar al demonio de Docker\) al ejecutar una compilación](#)
- [Error: "no CodeBuild está autorizado a ejecutar: sts:AssumeRole" al crear o actualizar un proyecto de compilación](#)
- [Error: «Error al llamar GetBucketAcl: el propietario del bucket ha cambiado o el rol de servicio ya no tiene permiso para llamar a s3:GetBucketAcl»](#)
- [Error: "Failed to upload artifacts: Invalid arn \(Error al cargar artefactos: arn no válido\)" al ejecutar una compilación](#)

- [Error: "Error de clon de Git: no es posible acceder a 'your-repository-URL': problema con el certificado de SSL: certificado autofirmado"](#)
- [Error: "The bucket you are attempting to access must be addressed using the specified endpoint \(El bucket al que intenta obtener acceso debe direccionarse utilizando el punto de conexión especificado\)" al ejecutar una compilación](#)
- [Error: "Esta imagen de compilación requiere seleccionar al menos una versión de tiempo de ejecución".](#)
- [Error: aparecer el error "QUEUED: INSUFFICIENT_SUBNET" cuando se produce un error en una compilación de una cola de compilaciones](#)
- [Error: «No se pudo descargar la caché RequestError: se produjo un error en el envío de la solicitud debido a: x509: no se pudieron cargar las raíces del sistema y no se proporcionaron las raíces»](#)
- [Error: «No se puede descargar el certificado de S3. AccessDenied»](#)
- [Error: "Unable to locate credentials" \(No se encuentran credenciales\)](#)
- [RequestError error de tiempo de espera cuando se ejecuta CodeBuild en un servidor proxy](#)
- [El shell de Bourne \(sh\) debe existir en las imágenes de compilación](#)
- [Advertencia: "Skipping install of runtimes. runtime version selection is not supported by this build image" \(Si se omite la instalación de los entornos de ejecución, no se podrá seleccionar la versión del entorno de ejecución en esta imagen de compilación\) al ejecutar una compilación](#)
- [Error: «No se puede verificar JobWorker la identidad» al abrir la CodeBuild consola](#)
- [No se ha podido iniciar la compilación](#)
- [Acceder a GitHub los metadatos en compilaciones almacenadas en caché local](#)
- [AccessDenied: El propietario del bucket del grupo de informes no coincide con el propietario del bucket de S3...](#)

Apache Maven crea artefactos de referencia en un repositorio incorrecto

Problema: [cuando usas Maven con un entorno AWS CodeBuild de compilación Java proporcionado, Maven extrae las dependencias de compilación y complementos del repositorio central seguro de Maven en <https://repo1.maven.org/maven2>](#). Esto es así aunque el archivo pom.xml del proyecto de compilación declare explícitamente otras ubicaciones para usar en su lugar.

Causa posible: los entornos CodeBuild de compilación de Java siempre incluyen un nombre de archivo `settings.xml` que está preinstalado en el directorio del entorno de compilación. `/root/.m2` Este archivo `settings.xml` contiene las siguientes declaraciones, que indican a Maven que obtenga siempre las dependencias de la compilación y los complementos del repositorio central seguro de Maven disponible en <https://repo1.maven.org/maven2>.

```
<settings>
  <activeProfiles>
    <activeProfile>securecentral</activeProfile>
  </activeProfiles>
  <profiles>
    <profile>
      <id>securecentral</id>
      <repositories>
        <repository>
          <id>central</id>
          <url>https://repo1.maven.org/maven2</url>
          <releases>
            <enabled>true</enabled>
          </releases>
        </repository>
      </repositories>
      <pluginRepositories>
        <pluginRepository>
          <id>central</id>
          <url>https://repo1.maven.org/maven2</url>
          <releases>
            <enabled>true</enabled>
          </releases>
        </pluginRepository>
      </pluginRepositories>
    </profile>
  </profiles>
</settings>
```

Solución recomendada: realice la siguiente operación:

1. Añada un archivo `settings.xml` al código fuente.
2. En este archivo `settings.xml`, use el formato de `settings.xml` anterior como guía para declarar los repositorios de los que desee que Maven obtenga las dependencias de la compilación y los complementos en su lugar.

3. En la `install` fase de tu proyecto de compilación, pide CodeBuild que copie el `settings.xml` archivo en el directorio del entorno de compilación. `/root/.m2` Por ejemplo, considere el siguiente fragmento de un archivo `buildspec.yml` que ilustra este comportamiento.

```
version 0.2

phases:
  install:
    commands:
      - cp ./settings.xml /root/.m2/settings.xml
```

De forma predeterminada, los comandos de compilación se ejecutan como usuario raíz

Problema: AWS CodeBuild ejecuta tus comandos de compilación como usuario root. Esto ocurre incluso si el Dockerfile de la imagen de compilación correspondiente establece la instrucción `USER` en otro usuario.

Causa: de forma predeterminada, CodeBuild ejecuta todos los comandos de compilación como usuario root.

Solución recomendada: ninguna.

Las compilaciones pueden generar un error si los nombres de archivo contienen caracteres ajenos al inglés (Estados Unidos)

Problema: cuando se ejecuta una compilación que usa archivos con nombres que contienen caracteres ajenos al inglés de Estados Unidos (por ejemplo, caracteres chinos), la compilación genera un error.

Causa posible: los entornos de compilación proporcionados por AWS CodeBuild tienen su configuración regional predeterminada establecida en. POSIX POSIXla configuración de localización es menos compatible con CodeBuild los nombres de archivo que no son de EE. UU. Son caracteres ingleses y pueden provocar errores en las compilaciones relacionadas.

Solución recomendada: añada los siguientes comandos a la sección `pre_build` del archivo `buildspec`. Estos comandos hacen que el entorno de compilación utilice UTF-8 en inglés

estadounidense como configuración de localización, lo que es más compatible con CodeBuild los nombres de archivo que no contienen caracteres estadounidenses. Caracteres ingleses.

Para los entornos de compilación basados en Ubuntu:

```
pre_build:
  commands:
    - export LC_ALL="en_US.UTF-8"
    - locale-gen en_US en_US.UTF-8
    - dpkg-reconfigure locales
```

Para los entornos de compilación basados en Amazon Linux:

```
pre_build:
  commands:
    - export LC_ALL="en_US.utf8"
```

Las compilaciones pueden fallar al obtener parámetros del almacén de parámetros de Amazon EC2

Problema: cuando una compilación intenta obtener el valor de uno o varios parámetros almacenados en el almacén de parámetros de Amazon EC2, la compilación produce el siguiente error en la fase `DOWNLOAD_SOURCE`: `Parameter does not exist`.

Causa posible: el rol de servicio en el que se basa el proyecto de compilación no tiene permiso para llamar a la `ssm:GetParameters` acción o el proyecto de compilación usa un rol de servicio generado por AWS CodeBuild y que permite llamar a la `ssm:GetParameters` acción, pero los parámetros tienen nombres que no comienzan por `/CodeBuild/`.

Soluciones recomendadas:

- Si el rol de servicio no lo generó CodeBuild, actualiza su definición para poder llamar CodeBuild a la `ssm:GetParameters` acción. Por ejemplo, la siguiente instrucción de política permite llamar a la acción `ssm:GetParameters` para obtener parámetros con nombres que empiecen por `/CodeBuild/`:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```

    "Action": "ssm:GetParameters",
    "Effect": "Allow",
    "Resource": "arn:aws:ssm:REGION_ID:ACCOUNT_ID:parameter/CodeBuild/*"
  }
]
}

```

- Si el rol de servicio lo generó CodeBuild, actualice su definición para permitir el acceso CodeBuild a los parámetros del almacén de parámetros de Amazon EC2 con nombres distintos de los que comienzan por. /CodeBuild/ Por ejemplo, la siguiente instrucción de política permite llamar a la acción `ssm:GetParameters` para obtener parámetros con el nombre especificado:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "ssm:GetParameters",
      "Effect": "Allow",
      "Resource": "arn:aws:ssm:REGION_ID:ACCOUNT_ID:parameter/PARAMETER_NAME"
    }
  ]
}

```

No se puede obtener acceso al filtro de ramificaciones en la consola de CodeBuild

Problema: la opción de filtro de ramas no está disponible en la consola al crear o actualizar un AWS CodeBuild proyecto.

Causa posible: la opción del filtro de ramificaciones ha quedado en desuso. Se ha sustituido por grupos de filtros de webhook, que proporcionan más control sobre los eventos de webhook que desencadenan una nueva compilación de CodeBuild.

Solución recomendada: para migrar un filtro de ramificaciones que se creó antes de que aparecieran los filtros de webhooks, cree grupos de filtros de webhooks con un filtro `HEAD_REF` y la expresión regular `^refs/heads/branchName$`. Por ejemplo, si la expresión regular del filtro de ramificaciones era `^branchName$`, la expresión regular actualizada que debe insertar en el filtro `HEAD_REF` es `^refs/heads/branchName$`. Para obtener más información, consulte [Eventos de webhooks de Bitbucket](#) y [Filtra los eventos de GitHub webhook \(consola\)](#).

No se puede ver si la compilación se ha realizado correctamente o no

Problema: al reintentar una compilación, no se puede ver si esta se ha realizado correctamente o no.

Causa posible: la opción de notificar el estado de la compilación no está habilitada.

Soluciones recomendadas: active el estado de creación de informes al crear o actualizar un CodeBuild proyecto. Esta opción indica a CodeBuild que le informe del estado cuando active una compilación. Para obtener más información, consulte [reportBuildStatus](#) en la Referencia de la API de AWS CodeBuild .

No se ha notificado el estado de la compilación al proveedor de fuentes

Problema: Tras permitir que un proveedor de fuentes, como Bitbucket GitHub o Bitbucket, informen sobre el estado de la compilación, el estado de la compilación no se actualiza.

Causa posible: el usuario asociado al proveedor de fuentes no tiene acceso de escritura al repositorio.

Soluciones recomendadas: para poder informar del estado de la compilación al proveedor de fuentes, el usuario asociado al proveedor de fuentes debe tener acceso de escritura al repositorio. Si el usuario no tiene acceso de escritura, no es posible actualizar el estado de compilación. Para obtener más información, consulte [Acceso al proveedor de fuentes](#).

No se puede encontrar ni seleccionar la imagen base de la plataforma Windows Server Core 2019

Problema: no se puede encontrar ni seleccionar la imagen base de la plataforma Windows Server Core 2019.

Causa posible: estás utilizando una AWS región que no admite esta imagen.

Soluciones recomendadas: utilice una de las siguientes regiones de AWS compatibles con la imagen base de la plataforma Windows Server Core 2019:

- Este de EE. UU. (Norte de Virginia)
- Este de EE. UU. (Ohio)
- Oeste de EE. UU. (Oregón)
- Europa (Irlanda)

Comandos anteriores de los archivos buildspec no reconocidos por comandos más recientes

Problema: los resultados de uno o varios comandos del archivo buildspec no los reconocen los comandos posteriores del mismo archivo buildspec. Por ejemplo, un comando podría establecer una variable de entorno local y un comando ejecutado más tarde podría no ser capaz de obtener el valor de esa variable de entorno local.

Causa posible: en la versión de archivo buildspec 0.1, AWS CodeBuild ejecuta cada comando en una instancia distinta del shell predeterminado en el entorno de compilación. Esto significa que cada comando se ejecuta con independencia de los demás. De forma predeterminada, no se puede ejecutar un comando que se base en el estado de un comando anterior.

Soluciones recomendadas: le recomendamos que utilice la versión de especificación de compilación 0.2, que soluciona este problema. Si necesita utilizar la versión de especificación de compilación 0.1, le recomendamos que utilice el operador de encadenamiento de comandos del shell (por ejemplo, `&&` en Linux) para combinar varios comandos en uno solo. También puede incluir un script del shell en el código fuente que contenga varios comandos y después llamar a dicho script desde un solo comando en el archivo buildspec. Para obtener más información, consulte [Intérpretes de comandos y comandos de los entornos de compilación](#) y [Variables de entorno en los entornos de compilación](#).

Error: "Access denied" (Acceso denegado) al intentar descargar la caché

Problema: al intentar descargar la caché en un proyecto de compilación que tiene la caché habilitada, aparece el error `Access denied`.

Causas posibles:

- Ha configurado la caché como parte de su proyecto de compilación.

- La caché se ha invalidado recientemente mediante la API `InvalidateProjectCache`.
- La función de servicio que utiliza CodeBuild no tiene `s3:GetObject` y `s3:PutObject` permisos para acceder al depósito de S3 que contiene la memoria caché.

Solución recomendada: durante el primer uso, es normal que aparezca este error inmediatamente después de actualizar la configuración de la caché. Si el error continúa, debe comprobar si el rol de servicio tiene los permisos `s3:GetObject` y `s3:PutObject` en el bucket de S3 donde se aloja la caché. Para obtener más información, consulte [Especificación de permisos de S3](#) en la Guía del desarrollador de Amazon S3.

Aparece el error

"BUILD_CONTAINER_UNABLE_TO_PULL_IMAGE" al usar una imagen de compilación personalizada

Problema: al intentar ejecutar una compilación que usa una imagen de compilación personalizada, la compilación produce el error `BUILD_CONTAINER_UNABLE_TO_PULL_IMAGE`.

Causa posible: el tamaño total sin comprimir de la imagen de compilación es mayor que el espacio en disco disponible del tipo de computación del entorno de compilación. Para comprobar el tamaño de la imagen de compilación, use Docker para ejecutar el comando `docker images REPOSITORY:TAG`. Para obtener una lista del espacio en disco disponible por tipo de computación, consulte [Modos y tipos de computación del entorno de compilación](#).

Solución recomendada: utilice un tipo de computación mayor con más espacio en disco o reduzca el tamaño de la imagen de compilación personalizada.

Causa posible: AWS CodeBuild no tiene permiso para extraer la imagen de compilación de su Amazon Elastic Container Registry (Amazon ECR).

Solución recomendada: actualice los permisos de su repositorio en Amazon ECR para que CodeBuild pueda incorporar su imagen de compilación personalizada al entorno de compilación. Para obtener más información, consulte [Ejemplo de Amazon ECR](#).

Causa posible: la imagen de Amazon ECR que has solicitado no está disponible en la AWS región que utiliza tu AWS cuenta.

Solución recomendada: utilice una imagen de Amazon ECR que esté en la misma AWS región que la que utiliza su AWS cuenta.

Causa posible: está utilizando un registro privado en una VPC que no tiene acceso público a Internet. CodeBuild no puede extraer una imagen de una dirección IP privada en una VPC. Para obtener más información, consulte [Registro privado con AWS Secrets Manager una muestra para CodeBuild](#).

Solución recomendada: si utiliza un registro privado de una VPC, asegúrese de que la VPC tiene acceso público a Internet.

Causa posible: si el mensaje de error contiene "toomanyrequests" y la imagen se obtiene de Docker Hub, este error significa que se ha alcanzado el límite de recuperación de Docker Hub.

Solución recomendada: utilice un registro privado de Docker Hub u obtenga la imagen de Amazon ECR. Para obtener más información acerca del uso de un registro privado, consulte [Registro privado con AWS Secrets Manager una muestra para CodeBuild](#). Para obtener más información sobre cómo usar Amazon ECR, consulte [Ejemplo de Amazon ECR para CodeBuild](#).

Error: «Se encontró que el contenedor de compilación estaba inactivo antes de completar la compilación. El contenedor de compilación murió porque no tenía memoria suficiente o la imagen de Docker no es compatible. ErrorCode: 500»

Problema: cuando intenta utilizar un contenedor de Microsoft Windows o Linux AWS CodeBuild, este error se produce durante la fase de APROVISIONAMIENTO.

Causas posibles:

- La versión del sistema operativo del contenedor no es compatible CodeBuild con.
- Se ha especificado HTTP_PROXY, HTTPS_PROXY o ambos en el contenedor.

Soluciones recomendadas:

- En Microsoft Windows, utilice un contenedor de Windows con un sistema operativo de contenedor que tenga la versión microsoft/windowsservercore:10.0.x. Por ejemplo, microsoft/windowsservercore:10.0.14393.2125.
- En Linux, desactive las opciones HTTPS_PROXY y HTTP_PROXY en la imagen de Docker o especifique la configuración de la VPC del proyecto de compilación.

Error: "Cannot connect to the Docker daemon" (No se puede conectar al demonio de Docker) al ejecutar una compilación

Problema: se ha producido un error en la compilación y aparece un error similar a `Cannot connect to the Docker daemon at unix:///var/run/docker.sock. Is the docker daemon running?` en el registro de la compilación.

Causa posible: no ha ejecutado la compilación en un modo con privilegios.

Solución recomendada: para corregir este error, debes habilitar el modo privilegiado y actualizar tu especificación de compilación siguiendo las siguientes instrucciones.

Para ejecutar la compilación en modo privilegiado, sigue estos pasos:

1. Abre la CodeBuild consola en <https://console.aws.amazon.com/codebuild/>.
2. En el panel de navegación, selecciona Crear proyectos y, a continuación, elige tu proyecto de compilación.
3. En Edit (Editar), seleccione Environment (Entorno).
4. Elija Configuración adicional.
5. En Privilegiado, selecciona Activar este indicador si quieres crear imágenes de Docker o si deseas que tus compilaciones obtengan privilegios elevados. .
6. Seleccione Update environment (Actualizar entorno).
7. Seleccione Start build (Comenzar compilación) para volver a intentar crear la compilación.

También tendrás que iniciar el daemon de Docker dentro de tu contenedor. La `install` fase de tu especificación de compilación podría tener un aspecto similar a este.

```
phases:
  install:
    commands:
      - nohup /usr/local/bin/dockerd --host=unix:///var/run/docker.sock --
host=tcp://127.0.0.1:2375 --storage-driver=overlay2 &
      - timeout 15 sh -c "until docker info; do echo .; sleep 1; done"
```

Para obtener más información acerca del controlador de almacenamiento de OverlayFS al que se hace referencia en el archivo `buildspec`, consulte [Use the OverlayFS storage driver](#) en el sitio web de Docker.

Note

Si el sistema operativo base es Alpine Linux, en el `buildspec.yml` añade el argumento `-t` a `timeout`:

```
- timeout -t 15 sh -c "until docker info; do echo .; sleep 1; done"
```

Para obtener más información sobre cómo crear y ejecutar una imagen de Docker mediante el uso, consulte. AWS CodeBuild [Ejemplo de imagen personalizada de Docker para CodeBuild](#)

Error: "no CodeBuild está autorizado a ejecutar: sts:AssumeRole" al crear o actualizar un proyecto de compilación

Problema: cuando intenta crear o actualizar un proyecto de compilación, aparece el error `Code:InvalidInputException, Message:CodeBuild is not authorized to perform: sts:AssumeRole on arn:aws:iam::account-ID:role/service-role-name`.

Causas posibles:

- El AWS Security Token Service (AWS STS) se ha desactivado para la AWS región en la que está intentando crear o actualizar el proyecto de construcción.
- El rol de AWS CodeBuild servicio asociado al proyecto de construcción no existe o no tiene permisos suficientes en los que confiar CodeBuild.

Soluciones recomendadas:

- Asegúrese de que AWS STS esté activado en la AWS región en la que está intentando crear o actualizar el proyecto de compilación. Para obtener más información, consulte [Activación y desactivación AWS STS en una AWS región en la](#) Guía del usuario de IAM.
- Asegúrese de que el rol de CodeBuild servicio de destino existe en su cuenta. AWS Si no utiliza la consola, asegúrese de que no haya escrito incorrectamente el nombre de recurso de Amazon (ARN) del rol de servicio cuando creó o actualizó el proyecto de compilación.
- Asegúrese de que el rol CodeBuild de servicio de destino tenga permisos suficientes en los que confiar CodeBuild. Para obtener más información, consulte la instrucción de relación de confianza de políticas en [Crear un rol de servicio de CodeBuild](#).

Error: «Error al llamar GetBucketAcl: el propietario del bucket ha cambiado o el rol de servicio ya no tiene permiso para llamar a s3:GetBucketAcl»

Problema: al ejecutar una compilación, recibe un error acerca de un cambio en la propiedad de un bucket de S3 y los permisos de GetBucketAcl.

Posible causa: ha añadido los permisos de s3:GetBucketAcl y s3:GetBucketLocation a su rol de IAM. Estos permisos protegen el bucket de S3 del proyecto y garantizan que solo usted tenga acceso a él. Después de agregar estos permisos, el propietario del bucket de S3 cambió.

Solución recomendada: compruebe que usted es el propietario del bucket de S3 y agregue de nuevo los permisos a su rol de IAM. Para obtener más información, consulte [Acceso seguro a los buckets de S3](#).

Error: "Failed to upload artifacts: Invalid arn (Error al cargar artefactos: arn no válido)" al ejecutar una compilación

Problema: cuando ejecuta una compilación, se produce el siguiente error en la fase UPLOAD_ARTIFACTS de la compilación: Failed to upload artifacts: Invalid arn.

Causa posible: el depósito de salida de S3 (el depósito en el que se AWS CodeBuild almacena la salida de la compilación) se encuentra en una AWS región distinta a la del proyecto de CodeBuild compilación.

Solución recomendada: actualiza la configuración del proyecto de compilación para que apunte a un depósito de salida que esté en la misma AWS región que el proyecto de compilación.

Error: "Error de clon de Git: no es posible acceder a '**your-repository-URL**': problema con el certificado de SSL: certificado autofirmado"

Problema: cuando intenta ejecutar un proyecto de compilación, se produce este error en la compilación.

Causa posible: el repositorio de código fuente tiene un certificado autofirmado, pero no ha elegido la opción de instalar el certificado desde el bucket de S3 durante el proyecto de compilación.

Soluciones recomendadas:

- Edite el proyecto. En Certificate, elija Install certificate from S3. En Bucket of certificate, elija el bucket de S3 en el que esté almacenado el certificado SSL. En Object key of certificate (Clave de objeto del certificado), escriba el nombre de la clave de objeto de S3.
- Edite el proyecto. Seleccione SSL inseguro para ignorar las advertencias de SSL al conectarse al repositorio de proyectos de GitHub Enterprise Server.

 Note

Le recomendamos que utilice Insecure SSL únicamente para pruebas. No debe utilizarse en un entorno de producción.

Error: "The bucket you are attempting to access must be addressed using the specified endpoint (El bucket al que intenta obtener acceso debe direccionarse utilizando el punto de conexión especificado)" al ejecutar una compilación

Problema: cuando ejecuta una compilación, se produce el siguiente error en la fase `DOWNLOAD_SOURCE` de la compilación: `The bucket you are attempting to access must be addressed using the specified endpoint. Please send all future requests to this endpoint.`

Causa posible: el código fuente prediseñado está almacenado en un depósito de S3 y ese depósito se encuentra en una AWS región distinta a la del proyecto de AWS CodeBuild compilación.

Solución recomendada: actualice la configuración del proyecto de compilación para que apunte a un bucket que contenga el código fuente precompilado. Asegúrese de que el depósito esté en la misma AWS región que el proyecto de compilación.

Error: "Esta imagen de compilación requiere seleccionar al menos una versión de tiempo de ejecución".

Problema: cuando ejecuta una compilación, se produce el siguiente error en la fase `DOWNLOAD_SOURCE` de la compilación: `YAML_FILE_ERROR: This build image requires selecting at least one runtime version.`

Causa posible: la compilación utiliza la versión 1.0 o posterior de la imagen estándar de Amazon Linux 2 (AL2) o la versión 2.0 o posterior de la imagen estándar de Ubuntu y no se especifica un runtime en el archivo `buildspec`.

Solución recomendada: si utilizas la imagen `aws/codebuild/standard:2.0` CodeBuild gestionada, debes especificar una versión en tiempo de ejecución en la `runtime-versions` sección del archivo `buildspec`. Por ejemplo, puede usar el siguiente archivo `buildspec` para un proyecto que usa PHP:

```
version: 0.2

phases:
  install:
    runtime-versions:
      php: 7.3
  build:
    commands:
      - php --version
artifacts:
  files:
    - README.md
```

Note

Si se especifica una sección de `runtime-versions` y se utiliza una imagen distinta de Ubuntu Standard Image 2.0 o posterior, o la imagen estándar de Amazon Linux 2 (AL2) 1.0 o posterior, la compilación mostrará la advertencia "Skipping install of runtimes. Runtime version selection is not supported by this build image".

Para obtener más información, consulte [Specify runtime versions in the buildspec file](#).

Error: aparecer el error "QUEUED: INSUFFICIENT_SUBNET" cuando se produce un error en una compilación de una cola de compilaciones

Problema: una compilación de una cola de compilaciones produce un error similar a QUEUED: INSUFFICIENT_SUBNET.

Causas posibles: el bloque de CIDR de IPv4 especificado para su VPC utiliza una dirección IP reservada. Las cuatro primeras direcciones IP y la última dirección IP de cada bloque de CIDR de las subredes no se pueden utilizar y no se pueden asignar a ninguna instancia. Por ejemplo, en una subred con el bloque de CIDR 10.0.0.0/24, estarán reservadas las cinco direcciones IP siguientes:

- 10.0.0.0:: dirección de red.
- 10.0.0.1: Reservado AWS por el router de VPC.
- 10.0.0.2: Reservado por AWS. La dirección IP del servidor DNS es siempre la base del intervalo de red VPC más dos; sin embargo, también se reserva la base de cada intervalo de subred más dos. En el caso de las VPC con varios bloques de CIDR, la dirección IP del servidor DNS se encuentra en el CIDR principal. Para obtener más información, consulte [Servidor DNS de Amazon](#) en la Guía del usuario de Amazon VPC.
- 10.0.0.3: Reservado AWS para uso futuro.
- 10.0.0.255: dirección de difusión de red. No admitimos la difusión en las VPC. Esta dirección está reservada.

Soluciones recomendadas: compruebe si su VPC utiliza una dirección IP reservada. Reemplace las direcciones IP reservadas por otras que no estén reservadas. Para obtener más información, consulte [Tamaño de subred y VPC](#) en la Guía del usuario de Amazon VPC.

Error: «No se pudo descargar la caché RequestError: se produjo un error en el envío de la solicitud debido a: x509: no se pudieron cargar las raíces del sistema y no se proporcionaron las raíces»

Problema: cuando intenta ejecutar un proyecto de compilación, se produce este error en la compilación.

Causa posible: ha configurado el almacenamiento en caché como parte del proyecto de compilación y está usando una imagen de Docker antigua que incluye un certificado raíz caducado.

Solución recomendada: actualice la imagen de Docker que se está utilizando en su proyecto AWS CodeBuild . Para obtener más información, consulte [Imágenes de Docker proporcionadas por CodeBuild](#).

Error: «No se puede descargar el certificado de S3. AccessDenied»

Problema: cuando intenta ejecutar un proyecto de compilación, se produce este error en la compilación.

Causas posibles:

- Ha elegido el bucket de S3 erróneo para el certificado.
- Ha introducido la clave de objeto errónea para el certificado.

Soluciones recomendadas:

- Edite el proyecto. En Bucket of certificate, elija el bucket de S3 en el que esté almacenado el certificado SSL.
- Edite el proyecto. En Object key of certificate (Clave de objeto del certificado), escriba el nombre de la clave de objeto de S3.

Error: "Unable to locate credentials" (No se encuentran credenciales)

Problema: cuando intentas ejecutar AWS CLI, usar un AWS SDK o llamar a otro componente similar como parte de una compilación, aparecen errores de compilación que están directamente relacionados con el AWS CLI AWS SDK o el componente. Por ejemplo, puede obtener un error de compilación como `Unable to locate credentials`.

Causas posibles:

- La versión del AWS CLI AWS SDK o del componente del entorno de compilación no es compatible con AWS CodeBuild.

- Estás ejecutando un contenedor de Docker en un entorno de compilación que usa Docker y el contenedor no tiene acceso a las AWS credenciales de forma predeterminada.

Soluciones recomendadas:

- Asegúrate de que tu entorno de compilación tenga la siguiente versión o una versión superior del AWS CLI AWS SDK o del componente.
 - AWS CLI: 1.10.47
 - AWS SDK para C++: 0.2.19
 - AWS SDK para Go: 1.2.5
 - AWS SDK para Java: 1.11.16
 - AWS SDK para JavaScript 2.4.7
 - AWS SDK para PHP: 3.18.28
 - AWS SDK para Python (Boto3): 1.4.0
 - AWS SDK para Ruby: 2.3.22
 - Botocore: 1.4.37
 - CoreCLR: 3.2.6-beta
 - Node.js: 2.4.7
- Si necesitas ejecutar un contenedor de Docker en un entorno de compilación y el contenedor requiere AWS credenciales, debes pasar las credenciales del entorno de compilación al contenedor. En el archivo buildspec, incluye un comando `run` de Docker como el siguiente. En este ejemplo se utiliza el comando `aws s3 ls` para mostrar los buckets de S3 disponibles. La `-e` opción pasa por las variables de entorno necesarias para que tu contenedor acceda a AWS las credenciales.

```
docker run -e AWS_DEFAULT_REGION -e AWS_CONTAINER_CREDENTIALS_RELATIVE_URI your-image-tag aws s3 ls
```

- Si está creando una imagen de Docker y la compilación requiere AWS credenciales (por ejemplo, para descargar un archivo de Amazon S3), debe pasar las credenciales del entorno de compilación al proceso de compilación de Docker de la siguiente manera.
 1. En el Dockerfile del código fuente para la imagen de Docker, especifique las siguientes instrucciones ARG.

```
ARG AWS_DEFAULT_REGION
```

```
ARG AWS_CONTAINER_CREDENTIALS_RELATIVE_URI
```

2. En el archivo buildspec, incluya un comando build de Docker como el siguiente. Las `--build-arg` opciones establecen las variables de entorno necesarias para que el proceso de compilación de Docker acceda a las credenciales. AWS

```
docker build --build-arg AWS_DEFAULT_REGION=$AWS_DEFAULT_REGION --build-arg  
AWS_CONTAINER_CREDENTIALS_RELATIVE_URI=$AWS_CONTAINER_CREDENTIALS_RELATIVE_URI -  
t your-image-tag .
```

RequestError error de tiempo de espera cuando se ejecuta CodeBuild en un servidor proxy

Problema: recibe un error RequestError similar a uno de los siguientes:

- RequestError: send request failed caused by: Post `https://logs.<your-region>.amazonaws.com/`: dial tcp 52.46.158.105:443: i/o timeout de los CloudWatch registros.
- Error uploading artifacts: RequestError: send request failed caused by: Put `https://<your-bucket>.s3.<your-aws-region>.amazonaws.com/*`: dial tcp 52.219.96.208:443: connect: connection refused de Amazon S3.

Causas posibles:

- `ssl-bump` no está configurado correctamente.
- La política de seguridad de la organización no permite utilizar `ssl_bump`.
- El archivo buildspec no tiene una configuración de proxy especificada con un elemento proxy.

Soluciones recomendadas:

- Asegúrese de que `ssl-bump` esté configurado correctamente. Si utiliza Squid con el servidor proxy, consulte [Configuración de Squid como un servidor proxy explícito](#).
- Siga estos pasos para usar puntos de enlace privados para Amazon S3 y CloudWatch Logs:

1. En la tabla de ruteo de la subred privada, quite la regla que añadió y que dirige el tráfico destinado a Internet al servidor proxy. Para obtener más información, consulte [Creación de una subred en su VPC](#) en la Guía del usuario de Amazon VPC.
 2. Cree un punto de conexión privado de Amazon S3 y un punto de enlace de CloudWatch Logs y asócielo a la subred privada de su Amazon VPC. Para obtener más información, consulte [Puntos de conexión de VPC](#) en la Guía del usuario de Amazon VPC.
 3. Compruebe que la opción Habilitar nombre de DNS privado de la Amazon VPC esté seleccionada. Para más información, consulte [Creación de un punto de conexión de interfaz](#) en la Guía del usuario de Amazon VPC.
- Si no utiliza `ssl-bump` para un servidor de proxy explícito, añada una configuración de proxy al archivo `buildspec` con un elemento `proxy`. Para obtener más información, consulte [Ejecución de CodeBuild en un servidor proxy explícito](#) y [Sintaxis de buildspec](#).

```
version: 0.2
proxy:
  upload-artifacts: yes
  logs: yes
phases:
  build:
    commands:
```

El shell de Bourne (sh) debe existir en las imágenes de compilación

Problema: está utilizando una imagen de compilación que no ha sido proporcionada por AWS CodeBuild y sus compilaciones fallan y aparece el mensaje. `Build container found dead before completing the build`

Causa posible: la consola Bourne (sh) no está incluida en la imagen de compilación. CodeBuild necesita sh ejecutar comandos y scripts de compilación.

Solución recomendada: si sh no está presente en la imagen de compilación, asegúrese de incluirlo antes de iniciar nuevas compilaciones que usen la imagen. (CodeBuild ya incluye sh en sus imágenes de compilación).

Advertencia: "Skipping install of runtimes. runtime version selection is not supported by this build image" (Si se omite la instalación de los entornos de ejecución, no se podrá seleccionar la versión del entorno de ejecución en esta imagen de compilación) al ejecutar una compilación

Problema: cuando ejecuta una compilación, el registro de compilación contiene esta advertencia.

Causa posible: la compilación no utiliza la versión 1.0 o posterior de la imagen estándar de Amazon Linux 2 (AL2) o la versión 2.0 o posterior de la imagen estándar de Ubuntu y se especifica un entorno de tiempo de ejecución en una sección `runtime-versions` del archivo de especificación de compilación.

Solución recomendada: asegúrese de que el archivo `buildspec` no contenga una sección `runtime-versions`. La sección `runtime-versions` solo es necesaria si se utiliza la imagen estándar de Amazon Linux 2 (AL2) o posterior o la versión 2.0 o posterior de la imagen estándar de Ubuntu.

Error: «No se puede verificar JobWorker la identidad» al abrir la CodeBuild consola

Problema: al abrir la CodeBuild consola, aparece el mensaje de error «No se puede verificar la JobWorker identidad».

Causa posible: el rol de IAM que se utiliza para acceder a la consola tiene una etiqueta con `jobId` como clave. Esta clave de etiqueta está reservada CodeBuild y, si está presente, provocará este error.

Solución recomendada: cambie las etiquetas de rol de IAM personalizadas que tengan la clave `jobId` por otra clave, por ejemplo `jobIdentifier`.

No se ha podido iniciar la compilación

Problema: al iniciar una compilación, se muestra un mensaje de error que indica que No se ha podido iniciar la compilación.

Causa posible: se ha alcanzado el número de compilaciones simultáneas.

Soluciones recomendadas: espere a que se completen otras compilaciones o aumente el límite de compilaciones simultáneas del proyecto y vuelva a iniciar la compilación. Para obtener más información, consulte [Configuración del proyecto](#).

Acceder a GitHub los metadatos en compilaciones almacenadas en caché local

Problema: en algunos casos, el directorio `.git` de una compilación en caché es un archivo de texto y no un directorio.

Causas posibles: cuando el almacenamiento en caché de fuentes locales está habilitado para una compilación, CodeBuild crea un gitlink para el directorio `.git`. Esto significa que el directorio de `.git` es realmente un archivo de texto que contiene la ruta hasta el directorio.

Soluciones recomendadas: en todos los casos, utilice el siguiente comando para obtener el directorio de metadatos de Git. Este comando funcionará independientemente del formato de `.git`:

```
git rev-parse --git-dir
```

AccessDenied: El propietario del bucket del grupo de informes no coincide con el propietario del bucket de S3...

Problema: al cargar datos de prueba en un bucket de Amazon S3, CodeBuild no puede escribir los datos de prueba en el bucket.

Causas posibles:

- El propietario del bucket del grupo de informes no coincide con el propietario del bucket de Amazon S3.
- El rol de servicio no tiene acceso de escritura en el bucket.

Soluciones recomendadas:

- Cambie el propietario del bucket del grupo de informes para que coincida con el propietario del bucket de Amazon S3.
- Modifique el rol de servicio para permitir el acceso de escritura al bucket de Amazon S3.

Cuotas para AWS CodeBuild

En las tablas siguientes, se indican las cuotas actuales de AWS CodeBuild. Estas cuotas se aplican a todas las regiones de AWS admitidas de cada cuenta de AWS, a menos que se especifique lo contrario.

Cuotas de servicio

A continuación se muestran las cuotas predeterminadas para el servicio AWS CodeBuild.

Nombre	Valor predeterminado	Ajuste	Descripción
Etiquetas asociadas por proyecto	Cada región admitida: 50	No	Número máximo de etiquetas que puede asociar a un proyecto de compilación
Proyectos de compilación	Cada región admitida: 5000	Sí	Número máximo de proyectos de compilación
Tiempo de espera de compilación en minutos	Cada región admitida: 480	No	Tiempo de espera máximo de compilación en minutos
Solicitud simultánea de información sobre las compilaciones	Cada región admitida: 100	No	Número máximo de compilaciones sobre las que puede solicitar información en un momento dado mediante la CLI de AWS o un SDK de AWS.
Solicitudes simultáneas de información sobre proyectos de compilación	Cada región admitida: 100	No	Número máximo de proyectos de compilación sobre los que puede solicitar información

Nombre	Valor predeterminado	Ajuste	Descripción
			en un momento dado mediante la CLI de AWS o un SDK de AWS.
Compilaciones en ejecución simultánea en un entorno ARM Lambda / 10 GB	Cada región admitida: 1	Sí	Número máximo de compilaciones en ejecución simultánea en un entorno ARM Lambda / 10 GB
Compilaciones en ejecución simultánea en un entorno ARM Lambda / 1 GB	Cada región admitida: 1	Sí	Número máximo de compilaciones en ejecución simultánea en un entorno ARM Lambda / 1 GB
Compilaciones en ejecución simultánea en un entorno ARM Lambda / 2 GB	Cada región admitida: 1	Sí	Número máximo de compilaciones en ejecución simultánea en un entorno ARM Lambda / 2 GB
Compilaciones en ejecución simultánea en un entorno ARM Lambda / 4 GB	Cada región admitida: 1	Sí	Número máximo de compilaciones en ejecución simultánea en un entorno ARM Lambda / 4 GB
Compilaciones en ejecución simultánea en un entorno ARM Lambda / 8 GB	Cada región admitida: 1	Sí	Número máximo de compilaciones en ejecución simultánea en un entorno ARM Lambda / 8 GB

Nombre	Valor predeterminado	Ajuste	Descripción
Compilaciones en ejecución simultánea en un entorno ARM / grande	Cada región admitida: 1	Sí	Número máximo de compilaciones en ejecución simultánea en un entorno ARM / grande
Compilaciones en ejecución simultánea en un entorno ARM / pequeño	Cada región admitida: 1	Sí	Número máximo de compilaciones en ejecución simultánea en un entorno ARM / pequeño
Compilaciones en ejecución simultánea en un entorno con GPU Linux grande	Cada región admitida: 0	Sí	Número máximo de compilaciones en ejecución simultánea en un entorno con GPU Linux / grande
Compilaciones en ejecución simultánea en un entorno con GPU Linux pequeño	Cada región admitida: 0	Sí	Número máximo de compilaciones en ejecución simultánea en un entorno con GPU Linux / pequeño
Compilaciones en ejecución simultánea en un entorno Linux Lambda / 10 GB	Cada región admitida: 1	Sí	Número máximo de compilaciones en ejecución simultánea en un entorno Linux Lambda / 10 GB
Compilaciones en ejecución simultánea en un entorno Linux Lambda / 1 GB	Cada región admitida: 1	Sí	Número máximo de compilaciones en ejecución simultánea en un entorno Linux Lambda / 1 GB

Nombre	Valor predeterminado	Ajuste	Descripción
Compilaciones en ejecución simultánea en un entorno Linux Lambda / 2 GB	Cada región admitida: 1	Sí	Número máximo de compilaciones en ejecución simultánea en un entorno Linux Lambda / 2 GB
Compilaciones en ejecución simultánea en un entorno Linux Lambda / 4 GB	Cada región admitida: 1	Sí	Número máximo de compilaciones en ejecución simultánea en un entorno Linux Lambda / 4 GB
Compilaciones en ejecución simultánea en un entorno Linux Lambda / 8 GB	Cada región admitida: 1	Sí	Número máximo de compilaciones en ejecución simultánea en un entorno Linux Lambda / 8 GB
Compilaciones en ejecución simultánea en un entorno Linux / 2xLarge	Cada región admitida: 0	Sí	Número máximo de compilaciones en ejecución simultánea en un entorno Linux / 2xLarge
Compilaciones en ejecución simultánea en un entorno Linux / grande	Cada región admitida: 1	Sí	Número máximo de compilaciones en ejecución simultánea en un entorno Linux / grande
Compilaciones en ejecución simultánea en un entorno Linux / mediano	Cada región admitida: 1	Sí	Número máximo de compilaciones en ejecución simultánea en un entorno Linux / mediano

Nombre	Valor predeterminado	Ajuste	Descripción
Compilaciones en ejecución simultánea en un entorno Linux / pequeño	Cada región admitida: 1	Sí	Número máximo de compilaciones en ejecución simultánea en un entorno Linux / pequeño
Compilaciones en ejecución simultánea en un entorno Linux / XLarge	Cada región admitida: 1	Sí	Número máximo de compilaciones en ejecución simultánea en un entorno Linux / XLarge
Compilaciones en ejecución simultánea en un entorno Windows Server 2019 / grande	Cada región admitida: 1	Sí	Número máximo de compilaciones en ejecución simultánea en un entorno Windows Server 2019 / grande
Compilaciones en ejecución simultánea en un entorno Windows Server 2019 / mediano	Cada región admitida: 1	Sí	Número máximo de compilaciones en ejecución simultánea en un entorno Windows Server 2019 / mediano
Compilaciones en ejecución simultánea en un entorno Windows / grande	Cada región admitida: 1	Sí	Número máximo de compilaciones en ejecución simultánea en un entorno Windows / grande
Compilaciones en ejecución simultánea en un entorno Windows / mediano	Cada región admitida: 1	Sí	Número máximo de compilaciones en ejecución simultánea en un entorno Windows / mediano

Nombre	Valor predeterminado	Ajuste	Descripción
Periodo de tiempo de espera mínimo de compilación en minutos	Cada región admitida: 5	No	Tiempo de espera mínimo de compilación en minutos
Grupos de seguridad en una configuración de VPC	Cada región admitida: 5	No	Grupos de seguridad disponibles para la configuración de VPC
Subredes en configuración de VPC	Cada región admitida: 16	No	Subredes disponibles para la configuración de VPC

Note

Las métricas internas determinarán las cuotas predeterminadas para las compilaciones que se ejecutan simultáneamente.

Las cuotas del número máximo de compilaciones en ejecución simultáneas varían en función del tipo de computación. Para algunos tipos de computación y plataformas, el valor predeterminado es 20. Para solicitar un aumento de la cuota de compilaciones simultáneas o si se produce el error "No se pueden tener más de X compilaciones activas en la cuenta", utilice el enlace de arriba para realizar la solicitud. Para obtener más información acerca de los precios, consulte [Precios de AWS CodeBuild](#).

Otros límites

Proyectos de compilación

Resource	Predeterminado
Caracteres permitidos en una descripción de proyecto de compilación	Cualquiera

Resource	Predeterminado
Caracteres permitidos en un nombre de proyecto de compilación	Las letras A-Z y a-z, los números 0-9 y los caracteres especiales - y _
Longitud de un nombre de proyecto de compilación	De 2 a 255 caracteres, inclusive
Longitud máxima de la descripción de un proyecto de compilación	255 caracteres
Número máximo de informes que puede añadir a un proyecto	5
Número de minutos que puede especificar en un proyecto de compilación para el tiempo de espera de la compilación de todas las compilaciones relacionadas	De 5 a 480 (8 horas)

Compilaciones

Resource	Predeterminado
Tiempo máximo que se conserva el historial de una compilación	1 año
Número de minutos que puede especificar para el tiempo de espera de compilación de una sola compilación	De 5 a 480 (8 horas)

Flotas de computación

Resource	Predeterminado
Número simultáneo de flotas de computación	10

Resource	Predeterminado
Instancias en ejecución simultánea para flotas de entornos ARM / pequeños	1
Instancias en ejecución simultánea para flotas de entornos ARM / grandes	1
Instancias en ejecución simultánea para flotas de entornos Linux / pequeños	1
Instancias en ejecución simultánea para flotas de entornos Linux / medianos	1
Instancias en ejecución simultánea para flotas de entornos Linux / grandes	1
Instancias en ejecución simultánea para flotas de entornos Linux / XLarge	1
Instancias en ejecución simultánea para flotas de entornos Linux / 2XLarge	0
Instancias en ejecución simultánea para flotas de entornos con GPU Linux / pequeños	0
Instancias en ejecución simultánea para flotas de entornos con GPU Linux / grandes	0
Instancias en ejecución simultánea para flotas de entornos Windows Server 2019 / medianos	1
Instancias en ejecución simultánea para flotas de entornos Windows Server 2019 / grandes	1
Instancias en ejecución simultánea para flotas de entornos Windows Server 2019 / medianos	1
Instancias en ejecución simultánea para flotas de entornos Windows Server 2019 / grandes	1

Informes

Resource	Predeterminado
La duración máxima de un informe de pruebas está disponible después de su creación	30 días
Longitud máxima del mensaje de un caso de prueba	5000 caracteres
Longitud máxima del nombre de un caso de prueba	1000 caracteres
Número máximo de grupos de informes por cuenta de AWS	5 000
Número máximo de casos de prueba por informe	500

Etiquetas

Los límites de etiquetas se aplican a las etiquetas en proyectos de compilación de CodeBuild y los recursos de grupos de informes de CodeBuild.

Resource	Predeterminado
Nombres de claves de etiquetas de recursos	<p>Cualquier combinación de letras, números, espacios y caracteres permitidos en UTF-8 con una longitud de entre 1 y 127 caracteres. Los caracteres permitidos son + - = . _ : / @</p> <p>Los nombres de clave de etiqueta deben ser únicos y cada clave solo puede tener un valor. Un nombre de clave de etiqueta no puede:</p> <ul style="list-style-type: none"> comenzar por aws : contener únicamente espacios

Resource	Predeterminado
	<ul style="list-style-type: none"> terminar con un espacio contener emojis ni ninguno de los siguientes caracteres: ? ^ * [\ ~ ! # \$ % & * () > < " ' ` [] { } ;
Valores de las etiquetas de recursos	<p>Cualquier combinación de letras, números, espacios y caracteres permitidos en UTF-8 con una longitud de entre 0 y 255 caracteres. Los caracteres permitidos son + - = . _ : / @</p> <p>Una clave solo puede tener un valor, pero muchas claves pueden tener el mismo valor. Un valor de clave de etiqueta no puede contener emojis ni ninguno de los siguientes caracteres: ? ^ * [\ ~ ! # \$ % & * () > < " ' ` [] { } ;</p>

Avisos de terceros AWS CodeBuild para Windows

Cuando las usa CodeBuild para compilaciones de Windows, tiene la opción de usar algunos paquetes y módulos de terceros para permitir que la aplicación compilada se ejecute en los sistemas operativos Microsoft Windows e interactúe con algunos productos de terceros. La lista siguiente contiene los términos legales de terceros aplicables que rigen el uso de los paquetes y módulos de terceros especificados.

Temas

- [1\) Imagen de Docker base - windowsservercore](#)
- [2\) Imagen de Docker base de Windows - Choco](#)
- [3\) Imagen de Docker base de Windows - git --versión 2.16.2](#)
- [4\) Imagen de Docker basada en Windows: versión 15.0.26320.2 microsoft-build-tools](#)
- [5\) Imagen de Docker base de Windows - nuget.commandline --version 4.5.1](#)
- [7\) Imagen de Docker base de Windows - netfx-4.6.2-devpack](#)
- [8\) Imagen de Docker base de Windows - visualfsharpools, v 4.0](#)
- [9\) Imagen de Docker basada en Windows: -4.6 netfx-pcl-reference-assemblies](#)
- [10\) Imagen de Docker base de Windows - visualcppbuildtools, v 14.0.25420.1](#)
- [11\) Imagen de Docker basada en Windows: 3-ondemand-package.cab microsoft-windows-netfx](#)
- [12\) Imagen de Docker base de Windows - dotnet-sdk](#)

1) Imagen de Docker base - windowsservercore

(los términos de licencia están disponibles en: https://hub.docker.com/_/microsoft-windows-servercore)

Licencia: al solicitar y utilizar esta imagen de SO de contenedor para contenedores de Windows, usted acepta, comprende y consiente los siguientes Términos de licencia complementarios:

TÉRMINOS DE LICENCIA COMPLEMENTARIOS DEL SOFTWARE DE MICROSOFT

IMAGEN DE SO DE CONTENEDOR

Microsoft Corporation (o, según dónde viva, una de sus filiales) (a la que se hace referencia como "nosotros", "nos" o "Microsoft") le otorga la licencia de este complemento de imagen de sistema

operativo de contenedor ("Complemento"). Se le concede la licencia para usar este Complemento junto con el software de sistema operativo host subyacente ("Software del host") exclusivamente para ayudar a ejecutar la característica de contenedores en el Software del host. Estos términos de licencia del Software del host se aplican al uso que usted realice del Complemento. No puede usarlo si no tiene una licencia para el Software del host. Puede usar este Complemento con cada copia con licencia válida del Software del host.

REQUISITOS Y/O DERECHOS DE USO DE LICENCIAS ADICIONALES

El uso del Complemento como se especifica en el párrafo anterior podría resultar en la creación o modificación de una imagen de contenedor ("Imagen de contenedor") que incluye algunos componentes del Complemento. Para mayor claridad, una Imagen de contenedor es independiente y distinta de una máquina virtual o de una imagen de aplicación virtual. En virtud de estos términos de licencia, te otorgamos un derecho restringido para redistribuir dichos componentes del Complemento en las siguientes condiciones:

- (i) puede usar los componentes del Complemento solo cuando se usan en y como parte de la Imagen de contenedor,
- (ii) puede usar esos componentes del complemento en la Imagen de contenedor siempre que tenga funcionalidad principal relevante en la Imagen de contenedor que sea materialmente independiente y distinta del Complemento; y
- (iii) acepta incluir estos términos de licencia (o términos similares requeridos por nosotros o un proveedor de servicios de hosting) con la Imagen de contenedor para licenciar correctamente el posible uso de los componentes del Complemento por parte de los usuarios finales.

Nos reservamos todos los demás derechos no otorgados expresamente en este documento.

Al hacer uso de este Complemento, acepta estos términos. Si no los acepta, no utilice este Complemento.

Como parte de los términos de licencia adicionales para esta imagen de sistema operativo de contenedor para los contenedores de Windows, también está sujeto a los términos de licencia subyacentes de software del host de Windows Server, que se encuentran en <https://www.microsoft.com/en-us/useterms>.

2) Imagen de Docker base de Windows - Choco

(los términos de la licencia están disponibles en: <https://github.com/chocolatey/choco/blob/master/LICENSE>)

Derechos de autor 2011 - RealDimensions Present Software, LLC

Con licencia bajo la licencia de Apache, versión 2.0 (la "Licencia"); no puede usar estos archivos excepto de conformidad con la Licencia. Puede obtener una copia de la Licencia en

<http://www.apache.org/licenses/LICENSE-2.0>

A menos que lo exija la legislación vigente o que se acuerde por escrito, el software distribuido bajo la licencia se distribuye "TAL CUAL", SIN GARANTÍAS NI CONDICIONES DE NINGÚN TIPO, ya sean explícitas o implícitas. Consulte la Licencia para conocer los permisos y limitaciones específicos en virtud de la Licencia.

3) Imagen de Docker base de Windows - git --versión 2.16.2

(términos de licencia disponibles en: <https://chocolatey.org/packages/git/2.16.2>)

Autorizado en virtud de GNU General Public License, versión 2, disponible en: <https://www.gnu.org/licenses/old-licenses/gpl-2.0.html>.

4) Imagen de Docker basada en Windows: versión 15.0.26320.2 microsoft-build-tools

(términos de licencia disponibles en: <https://www.visualstudio.com/license-terms/mt171552/>)

EXTENSIONES DE MICROSOFT VISUAL STUDIO 2015, SHELL DE VISUAL STUDIO y C++
REDISTRIBUTABLE

Los presentes términos de licencia constituyen un acuerdo entre Microsoft Corporation (o, en función de donde resida, una de sus filiales) y usted. Se aplican al software citado anteriormente. Los términos también se aplican a cualquier servicio y actualización de Microsoft para el software, excepto en la medida que tengan términos diferentes.

SI CUMPLE ESTOS TÉRMINOS DE LICENCIA, TIENE LOS DERECHOS QUE SE INDICAN A CONTINUACIÓN.

1. DERECHOS DE USO Y DE INSTALACIÓN. Puede instalar y utilizar cualquier número de copias del software.
2. TÉRMINOS PARA COMPONENTES ESPECÍFICOS.
 - a. Utilidades. El software puede contener algunos elementos de la lista de utilidades en <https://docs.microsoft.com/en-us/visualstudio/productinfo/2015-redistribution-vs>. Puede copiar e instalar esos elementos, si están incluidos con el software, en su equipo o en otros equipos de terceros, para depurar e implementar las aplicaciones y bases de datos que haya desarrollado con el software. Tenga en cuenta que las Utilidades se han diseñado para uso temporal, que Microsoft podría no ser capaz de aplicar parches o actualizar las Utilidades por separado del resto del software y que algunas Utilidades por su naturaleza podrían hacer posible a otras personas acceder a los equipos en los que están instalados. En consecuencia, debería eliminar todas las Utilidades que haya instalado después de terminar de depurar o de implementar sus aplicaciones y bases de datos. Microsoft no es responsable del uso por parte de terceros o del acceso a las Utilidades que instale en cualquier equipo.
 - b. Plataformas de Microsoft. El software puede incluir componentes de Microsoft Windows, Microsoft Windows Server, Microsoft SQL Server, Microsoft Exchange, Microsoft Office y Microsoft SharePoint. Estos componentes se rigen por acuerdos independientes y por sus propias políticas de soporte de productos, tal y como se describe en los términos de licencia que se encuentran en el directorio de instalación de dicho componente o en la carpeta "Licencias" que acompaña al software.
 - c. Componentes de terceros. El software puede incluir componentes de terceros con avisos legales independientes o regirse por otros acuerdos, tal y como se describe en el ThirdPartyNotices archivo adjunto al software. Incluso si tales componentes se rigen por otros acuerdos, también se aplican los descargos de responsabilidad y las limitaciones y exclusiones de daños siguientes. El software también podría incluir componentes sujetos a licencias de código abierto con obligaciones de disponibilidad de código fuente. En el ThirdPartyNotices archivo se incluyen copias de esas licencias, si procede. Puede obtener este código fuente correspondiente completo de Microsoft, si se requiere y tal como se requiera en virtud de las licencias de código abierto pertinente enviando un giro postal o cheque por valor de 5,00 USD a: Source Code Compliance Team, Microsoft Corporation, 1 Microsoft Way, Redmond, WA 98052. Escriba código fuente de uno o varios de los componentes indicados a continuación en la línea de memorando de su pago:
 - Herramientas remotas para Visual Studio 2015;

- Generador de perfiles independiente para Visual Studio 2015;
- IntelliTraceCollector para Visual Studio 2015;
- Microsoft VC++ redistribuible 2015;
- Biblioteca Multibyte MFC para Visual Studio 2015;
- Microsoft Build Tools 2015;
- Feedback Client;
- Shell integrado de Visual Studio 2015; o
- Shell aislado de Visual Studio 2015.

También podemos realizar una copia del código fuente disponible en <http://thirdpartysource.microsoft.com>.

3. **DATOS.** El software podría recopilar información acerca de usted y de su uso del software y enviarlo a Microsoft. Microsoft podría utilizar esta información para proporcionar servicios y mejorar nuestros productos y servicios. Es posible darse de baja en muchas de estas situaciones, pero no en todas, tal y como se describe en la documentación del producto. También hay algunas características en el software que le permitirían recopilar datos de los usuarios de sus aplicaciones. Si utiliza estas características para permitir la recopilación de datos en sus aplicaciones, debe cumplir con la legislación aplicable, que incluye la entrega de notificaciones pertinentes a los usuarios de sus aplicaciones. Puede obtener más información sobre la recopilación y el uso de datos en la documentación de ayuda y la declaración de privacidad en <https://privacy.microsoft.com/en-us/privacystatement>. El uso que hace del software funciona como su consentimiento para estas prácticas.
4. **ÁMBITO DE LA LICENCIA.** El software se cede bajo licencia y no es objeto de venta. Este acuerdo solo le otorga algunos derechos de uso del software. Microsoft se reserva todos los demás derechos. A menos que la legislación aplicable le otorgue más derechos a pesar de esta limitación, solo podrá utilizar el software tal como se permite expresamente en este acuerdo. Al hacerlo, deberá ajustarse a las limitaciones técnicas del software que solo permiten utilizarlo de determinadas formas. No podrá
 - eludir las limitaciones técnicas del software;
 - utilizar técnicas de ingeniería inversa, descompilar o desensamblar el software, así como tampoco intentar hacerlo, excepto y en la medida que lo exijan los términos de licencia de terceros que rigen el uso de ciertos componentes de código abierto que se podrían incluir con el software;

- eliminar, minimizar, bloquear o modificar ninguna notificación de Microsoft o sus proveedores en el software;
 - utilizar el software de ninguna manera que esté en contra de la ley; o
 - compartir, publicar, alquilar o dar el software en préstamo, ni entregarlo como solución hospedada independiente para que otros lo utilicen.
5. **RESTRICCIONES EN MATERIA DE EXPORTACIÓN.** Debe cumplir con todas las leyes y reglamentos, nacionales e internacionales, en materia de exportación que sean de aplicación al software, lo que incluye restricciones en cuanto a destino, usuarios finales y uso final. Para obtener más información sobre las restricciones en materia de exportación, visite (aka.ms/exporting).
6. **SERVICIOS DE SOPORTE TÉCNICO.** Dado que este software se suministra "tal cual", es posible que no se proporcionen servicios de soporte técnico para el mismo.
7. **ACUERDO COMPLETO.** Este acuerdo y los términos aplicables a complementos, actualizaciones, servicios basados en Internet y servicios de soporte técnico que utilice constituyen el acuerdo completo para el software y los servicios de soporte técnico.
8. **LEGISLACIÓN APLICABLE.** Si adquirió el software en los Estados Unidos, la legislación del estado de Washington regirá la interpretación de este acuerdo, así como las reclamaciones por incumplimiento, y la legislación del estado donde usted resida regirá todas las demás reclamaciones. Si adquiere el software en otro país, se aplicará la legislación de dicho país.
9. **DERECHOS DEL CONSUMIDOR, VARIACIONES REGIONALES.** En este acuerdo se describen determinados derechos legales. Puede que usted tenga otros derechos, incluidos derechos del consumidor, según la legislación de su estado o país. Por separado e independientemente de su relación con Microsoft, también puede tener derechos con respecto a la parte de la que obtuvo el software. Este acuerdo no modifica los otros derechos de los que dispone en virtud de la legislación de su estado, país o región si dicha legislación no lo permite. Por ejemplo, si adquirió el software en una de las regiones que figuran a continuación, o si se aplican leyes obligatorias del país o la región, las siguientes disposiciones se aplican a su caso:
- a. **Australia.** Tiene garantías legales previstas según la Ley de Consumo australiana y ninguna parte de este acuerdo pretende afectar a dichos derechos.
 - b. **Canadá.** Si adquirió este software en Canadá, para dejar de recibir actualizaciones, puede desactivar la función de actualizaciones automáticas, desconectar su dispositivo de Internet (sin embargo, cuando vuelva a conectarse a Internet el software volverá a comprobar e instalar actualizaciones) o desinstalar el software. La documentación del producto, de haberla, también puede especificar cómo desactivar las actualizaciones para su dispositivo o software específico.

c. Alemania y Austria.

- i. Garantía. El software que disponga de la debida licencia funcionará esencialmente como se describe en la documentación de Microsoft que lo acompaña. Sin embargo, Microsoft no otorga ninguna garantía contractual en relación con el software licenciado.
- ii. Limitación de responsabilidad. En caso de comportamiento intencional, negligencia grave, reclamaciones basadas en la Ley de Responsabilidad por Productos, así como en caso de muerte, daños corporales o lesiones físicas, Microsoft es responsable conforme a la legislación. Sujeto a la cláusula anterior (ii), Microsoft solo será responsable de negligencia leve si Microsoft incumple las obligaciones contractuales materiales, cuyo cumplimiento permite la debida ejecución de este acuerdo, y cuya infracción haría peligrar el propósito de este acuerdo y la confianza de una parte en su cumplimiento (denominadas "obligaciones cardinales"). En otros casos de negligencia leve, Microsoft no será responsable de dicha negligencia.

10 RENUNCIA DE GARANTÍA. EL SOFTWARE SE CONCEDE BAJO LICENCIA "TAL CUAL". USTED ASUME EL RIESGO DE UTILIZARLO. MICROSOFT NO PROPORCIONA NINGUNA GARANTÍA EXPRESA, ASÍ COMO TAMPOCO AVALES O CONDICIONES. EN LA MEDIDA EN QUE LO PERMITE LA LEGISLACIÓN LOCAL, MICROSOFT EXCLUYE LAS GARANTÍAS IMPLÍCITAS DE COMERCIALIZACIÓN, IDONEIDAD PARA UN DETERMINADO FIN Y NO INFRACCIÓN.

11 LIMITACIÓN Y EXCLUSIÓN DE INDEMNIZACIONES. PUEDE RECUPERAR DE MICROSOFT Y SUS PROVEEDORES ONLY DAÑOS DIRECTOS EE. UU. HASTA 5,00 USD. NO PODRÁ OBTENER COMPENSACIÓN POR NINGÚN OTRO DAÑO, INCLUIDOS DAÑOS CONSECUENTES, PÉRDIDAS DE GANANCIAS, DAÑOS ESPECIALES, DAÑOS INDIRECTOS O DAÑOS INCIDENTALES. Esta limitación se aplica a (a) nada relacionado con el software, servicios, contenido, (incluido código) en sitios de Internet o aplicaciones de terceros; y (b) reclamaciones por incumplimiento del contrato, incumplimiento de garantía o condición, responsabilidad estricta, negligencia u otros perjuicios en la medida permitida por la ley vigente.

También se aplicará incluso si Microsoft conocía o debería haber conocido la posibilidad de que se produjesen dichos daños. La limitación o exclusión precedente podría no aplicarse a su caso porque el país podría no admitir la exclusión o limitación de daños accidentales, resultantes o de otra índole.

ID de EULA: ShellsRedist VS2015_Update3_ _ <ENU>

5) Imagen de Docker base de Windows - nuget.commandline --version 4.5.1

(los términos de licencia están disponibles en: <https://github.com/ /Home/blob/dev/LICENSE.txt>)

NuGet

Copyright (c) .NET Foundation. Todos los derechos reservados.

Con licencia bajo la licencia de Apache, versión 2.0 (la "Licencia"); no puede usar estos archivos excepto de conformidad con la Licencia. Puede obtener una copia de la Licencia en

<http://www.apache.org/licenses/LICENSE-2.0>

A menos que lo exija la legislación vigente o que se acuerde por escrito, el software distribuido bajo la licencia se distribuye "TAL CUAL", SIN GARANTÍAS NI CONDICIONES DE NINGÚN TIPO, ya sean explícitas o implícitas. Consulte la Licencia para conocer los permisos y limitaciones específicos en virtud de la Licencia.

7) Imagen de Docker base de Windows - netfx-4.6.2-devpack

TÉRMINOS DE LICENCIA COMPLEMENTARIOS DEL SOFTWARE DE MICROSOFT

.NET FRAMEWORK 4.5 Y PAQUETES DE IDIOMAS ASOCIADOS PARA SISTEMA OPERATIVO
MICROSOFT WINDOWS

Microsoft Corporation (o, en función de donde resida, una de sus filiales) le otorga licencia para este complemento. Si posee licencias para utilizar software de sistema operativo Windows (el "software"), puede utilizar este complemento. No puede utilizarlo si no tiene una licencia para el software. Puede utilizar este complemento con cada copia de licencia válida del software.

Los siguientes términos de licencia describen términos de uso adicionales para este complemento. Los presentes términos y los términos de licencia para el software se aplican al uso que usted haga del complemento. Si hay un conflicto, se aplican estos términos de licencia complementarios.

AL HACER USO DE ESTE COMPLEMENTO, ACEPTA ESTOS TÉRMINOS. SI NO LOS ACEPTA, NO UTILICE ESTE COMPLEMENTO.

Si cumple estos términos de licencia, tiene los derechos que se indican a continuación.

1. **CÓDIGO DISTRIBUIBLE.** El complemento se compone de Código distribuible. El "Código distribuible" es código que puede distribuir en los programas que desarrolle siempre que cumpla con los términos siguientes.
 - a. **Derecho a Utilizar y Distribuir.**
 - Puede copiar y distribuir el código objeto del complemento.
 - Distribución de Terceros. Puede permitir a los distribuidores de sus programas que copien y distribuyan el Código distribuible como parte de esos programas.
 - b. **Requisitos de distribución.** Para cualquier Código distribuible que distribuya, debe
 - agregarle una funcionalidad primaria significativa en sus programas;
 - para cualquier Código distribuible que tenga una extensión de nombre de archivo .lib, distribuya solo los resultados de ejecutar dicho Código distribuible a través de un vinculador con su programa;
 - distribuir Código distribuible incluido en un programa de instalación como parte de ese programa de instalación sin modificarlo;
 - exigir a los distribuidores y usuarios finales externos que acepten términos que lo protejan en la misma medida que este acuerdo;
 - mostrar su aviso de propiedad intelectual válido en sus programas; y
 - indemnizar, defender y eximir de responsabilidad a Microsoft de cualquier reclamación, incluidos honorarios de abogados, relacionados con la distribución o el uso de sus programas.
 - c. **Restricciones a la distribución.** No debe
 - alterar ningún aviso de propiedad intelectual, marca o patente en el Código distribuible;
 - utilizar las marcas de Microsoft en los nombres de sus programas de una forma que sugiera que estos provienen de Microsoft o que esta los respalda;
 - distribuir el Código distribuible para que se ejecute en una plataforma que no sea Windows;
 - incluir Código distribuible en programas malintencionados, engañosos o ilícitos; o
 - modificar o distribuir el código fuente de cualquier Código distribuible para que cualquier parte de él esté sujeta a una Licencia excluida. Una Licencia excluida es una que exige, como condición de uso, modificación o distribución que
 - el código se revele o distribuya en código fuente; u

- otros tengan el derecho a modificarlo.

2. SERVICIOS DE SOPORTE TÉCNICO PARA EL COMPLEMENTO. Microsoft proporciona los servicios de soporte técnico para el presente software que se describen en www.support.microsoft.com/common/international.aspx.

8) Imagen de Docker base de Windows - visualfsharpertools, v 4.0

(los términos de licencia están disponibles en: <https://github.com/dotnet/fsharp/blob/main/License.txt>)

Copyright (c) Microsoft Corporation. Todos los derechos reservados.

Con licencia bajo la licencia de Apache, versión 2.0 (la "Licencia"); no puede usar estos archivos excepto de conformidad con la Licencia. Puede obtener una copia de la Licencia en

<http://www.apache.org/licenses/LICENSE-2.0>

A menos que lo exija la legislación vigente o que se acuerde por escrito, el software distribuido bajo la licencia se distribuye "TAL CUAL", SIN GARANTÍAS NI CONDICIONES DE NINGÚN TIPO, ya sean explícitas o implícitas. Consulte la Licencia para conocer los permisos y limitaciones específicos en virtud de la Licencia.

9) Imagen de Docker basada en Windows: -4.6 netfx-pcl-reference-assemblies

TÉRMINOS DE LICENCIA DEL SOFTWARE DE MICROSOFT

ENSAMBLADOS DE REFERENCIA DE BIBLIOTECA DE CLASES PORTABLES DE MICROSOFT .NET - 4.6

Los presentes términos de licencia constituyen un acuerdo entre Microsoft Corporation (o, en función de donde resida, una de sus filiales) y usted. Por favor, léalos. Se aplican al software citado anteriormente. Los términos también se aplican a

- actualizaciones,
- complementos,

- servicios basados en Internet, y
- servicios de soporte técnico

de Microsoft para este software, a menos que haya otros términos que acompañen a dichos elementos. En tal caso, se aplican dichos términos.

AL HACER USO DEL SOFTWARE, ACEPTA ESTOS TÉRMINOS. SI NO LOS ACEPTA, NO UTILICE EL SOFTWARE.

SI CUMPLE ESTOS TÉRMINOS DE LICENCIA, TIENE LOS DERECHOS PERPETUOS QUE SE INDICAN A CONTINUACIÓN.

1. DERECHOS DE USO Y DE INSTALACIÓN. Puede instalar y utilizar cualquier número de copias del software para diseñar, desarrollar y probar los programas.
2. REQUISITOS Y/O DERECHOS DE USO DE LICENCIAS ADICIONALES
 - a. Código distribuible. Puede distribuir el software en programas de herramienta de desarrollador que usted desarrolle, para permitir a los clientes de sus programas desarrollar bibliotecas portables para utilizarlas en cualquier dispositivo o sistema operativo, siempre que cumpla con los términos siguientes.
 - i. Derecho a Utilizar y Distribuir. El software es "Código distribuible".
 - Código distribuible. Puede copiar y distribuir el código objeto del software.
 - Distribución de Terceros. Puede permitir a los distribuidores de sus programas que copien y distribuyan el Código distribuible como parte de esos programas.
 - ii. Requisitos de distribución. Para cualquier Código distribuible que distribuya, debe
 - agregarle una funcionalidad primaria significativa en sus programas;
 - exigir a los distribuidores y a sus clientes que acepten términos que lo protejan en la misma medida que este acuerdo;
 - mostrar su aviso de propiedad intelectual válido en sus programas; y
 - indemnizar, defender y eximir de responsabilidad a Microsoft de cualquier reclamación, incluidos honorarios de abogados, relacionados con la distribución o el uso de sus programas.
 - iii. Restricciones a la distribución. No debe
 - alterar ningún aviso de propiedad intelectual, marca o patente en el Código distribuible;

- utilizar las marcas de Microsoft en los nombres de sus programas de una forma que sugiera que estos provienen de Microsoft o que esta los respalda;
- incluir Código distribuible en programas malintencionados, engañosos o ilícitos; o
- modificar o distribuir el Código distribuible para que cualquier parte de él esté sujeta a una Licencia excluida. Una Licencia excluida es una que exige, como condición de uso, modificación o distribución que
 - el código se revele o distribuya en código fuente; u
 - otros tengan el derecho a modificarlo.

3. **ÁMBITO DE LA LICENCIA.** El software se cede bajo licencia y no es objeto de venta. Este acuerdo solo le otorga algunos derechos de uso del software. Microsoft se reserva todos los demás derechos. A menos que la legislación aplicable le otorgue más derechos a pesar de esta limitación, solo podrá utilizar el software tal como se permite expresamente en este acuerdo. Al hacerlo, deberá ajustarse a las limitaciones técnicas del software que solo permiten utilizarlo de determinadas formas. No podrá

- eludir las limitaciones técnicas del software;
- utilizar técnicas de ingeniería inversa, descompilar o desensamblar el software, excepto y en la medida que lo permita expresamente la legislación aplicable, a pesar de esta limitación;
- publicar el software para que otros lo copien; o
- alquilar o prestar el software.

4. **COMENTARIOS.** Puede proporcionar comentarios sobre el software. Si envía comentarios acerca de la aplicación a Microsoft, otorga a Microsoft, sin cargo alguno, el derecho de usar, compartir y comercializar sus comentarios de cualquier modo y para cualquier fin. También otorga a terceros, sin cargo alguno, los derechos de patente necesarios para que sus productos, tecnologías y servicios usen o interactúen con cualquier pieza específica de un software de Microsoft o un servicio que incluya los comentarios. Usted no enviará comentarios sujetos a licencia que obliguen a Microsoft a conceder su software o documentación bajo licencia a terceros por el hecho de que se incluyan en ellos sus comentarios. Dichos derechos se mantendrán en vigor tras la finalización del presente acuerdo.

5. **TRANSFERENCIA A TERCEROS.** El primer usuario del software podrá transferirlo y este acuerdo, directamente a un tercero. Antes de la transferencia, dicha parte debe aceptar que este acuerdo se aplica a la transferencia y al uso del software. El primer usuario debe desinstalar el software antes de transferirlo por separado desde el dispositivo. El primer usuario no puede conservar ninguna copia.

6. **RESTRICCIONES EN MATERIA DE EXPORTACIÓN.** El software está sujeto a las leyes de exportación y a los reglamentos de los Estados Unidos. Debe cumplir todas las leyes y los reglamentos de exportación nacionales e internacionales que se aplican al software. Las leyes incluyen restricciones sobre destinos, usuarios finales y uso final. Para obtener información adicional, consulte www.microsoft.com/exporting.
7. **SERVICIOS DE SOPORTE TÉCNICO.** Dado que este software se suministra "tal cual", es posible que no se proporcionen servicios de soporte técnico para el mismo.
8. **ACUERDO COMPLETO.** Este acuerdo y los términos aplicables a complementos, actualizaciones, servicios basados en Internet y servicios de soporte técnico que utilice constituyen el acuerdo completo para el software y todos los servicios de soporte técnico que prestamos.
9. **LEGISLACIÓN APLICABLE.**
 - a. **Estados Unidos.** Si adquirió el software en los Estados Unidos, la legislación del estado de Washington regirá la interpretación de este acuerdo, así como las reclamaciones por incumplimiento, independientemente de los principios de conflicto de leyes. La legislación del estado donde reside rige todas las demás reclamaciones, incluidas las reclamaciones bajo las leyes de protección al consumidor del estado, leyes de competencia desleal o perjuicio.
 - b. **Fuera de los Estados Unidos.** Si adquiere el software en otro país, se aplicará la legislación de dicho país.
10. **EFFECTO LEGAL.** En este acuerdo se describen determinados derechos legales. Es posible que tenga otros derechos en virtud de las leyes de su país. También podría tener derechos con respecto a la parte a la que adquirió el software. Este acuerdo no modifica sus derechos de los que dispone en virtud de la legislación de su país si dicha legislación no lo permite.
11. **RENUNCIA DE GARANTÍA. EL SOFTWARE SE CONCEDE BAJO LICENCIA "TAL CUAL". USTED ASUME EL RIESGO DE UTILIZARLO. MICROSOFT NO PROPORCIONA NINGUNA GARANTÍA EXPRESA, ASÍ COMO TAMPOCO AVALES O CONDICIONES. PUEDE TENER DERECHOS DE CONSUMIDOR O GARANTÍAS LEGALES ADICIONALES SEGÚN SU LEGISLACIÓN LOCAL QUE ESTE ACUERDO NO PUEDE CAMBIAR. EN LA MEDIDA EN QUE LO PERMITE LA LEGISLACIÓN LOCAL, MICROSOFT EXCLUYE LAS GARANTÍAS IMPLÍCITAS DE COMERCIALIZACIÓN, IDONEIDAD PARA UN DETERMINADO FIN Y NO INFRACCIÓN.**

PARA AUSTRALIA: TIENE GARANTÍA LEGALES PREVISTAS SEGÚN LA LEY DE CONSUMO AUSTRALIANA Y NINGUNA PARTE DE ESTOS TÉRMINOS PRETENDE AFECTAR A DICHOS DERECHOS.
12. **LIMITACIÓN Y EXCLUSIÓN DE COMPENSACIONES E INDEMNIZACIONES. PUEDE RECUPERAR DE MICROSOFT Y SUS PROVEEDORES ONLY DAÑOS DIRECTOS EE. UU.**

HASTA 5,00 USD. NO PODRÁ OBTENER COMPENSACIÓN POR NINGÚN OTRO DAÑO, INCLUIDOS DAÑOS CONSECUENTES, PÉRDIDAS DE GANANCIAS, DAÑOS ESPECIALES, DAÑOS INDIRECTOS O DAÑOS INCIDENTALES.

Esta limitación se aplica a

- cualquier aspecto relacionado con el software, los servicios, el contenido (incluido el código) en sitios de Internet de terceros o programas de terceros; y
- reclamaciones por infracción de contrato, infracción de garantía o condición, responsabilidad estricta, negligencia u otros perjuicios en la medida permitida por la legislación aplicable.

También se aplicará incluso si Microsoft conocía o debería haber conocido la posibilidad de que se produjesen dichos daños. La limitación o exclusión precedente podría no aplicarse a su caso porque el país podría no admitir la exclusión o limitación de daños accidentales, resultantes o de otra índole.

10) Imagen de Docker base de Windows - visualcppbuildtools, v 14.0.25420.1

(términos de licencia disponibles en: <https://www.visualstudio.com/license-terms/mt644918/>)

HERRAMIENTAS DE COMPILACIÓN DE MICROSOFT VISUAL C++

TÉRMINOS DE LICENCIA DEL SOFTWARE DE MICROSOFT

HERRAMIENTAS DE COMPILACIÓN DE MICROSOFT VISUAL C++

Los presentes términos de licencia constituyen un acuerdo entre Microsoft Corporation (o, en función de donde resida, una de sus filiales) y usted. Se aplican al software citado anteriormente. Los términos también se aplican a cualquier servicio o actualizaciones de Microsoft para el software, salvo en la medida en que dispongan de distintos términos.

SI CUMPLE ESTOS TÉRMINOS DE LICENCIA, TIENE LOS DERECHOS QUE SE INDICAN A CONTINUACIÓN.

1. DERECHOS DE USO Y DE INSTALACIÓN.

- a. Un usuario puede utilizar copias del software para desarrollar y probar sus aplicaciones.
2. DATOS. El software podría recopilar información acerca de usted y de su uso del software y enviarlo a Microsoft. Microsoft podría utilizar esta información para proporcionar servicios y mejorar nuestros productos y servicios. Es posible darse de baja en muchas de estas situaciones, pero no en todas, tal y como se describe en la documentación del producto. También hay algunas características en el software que le permitirían recopilar datos de los usuarios de sus aplicaciones. Si utiliza estas características para permitir la recopilación de datos en sus aplicaciones, debe cumplir con la legislación aplicable, que incluye la entrega de notificaciones pertinentes a los usuarios de sus aplicaciones. Puede obtener más información sobre la recopilación y el uso de datos en la documentación de ayuda y en la declaración de privacidad en <http://go.microsoft.com/fwlink/?LinkID=528096>. El uso que hace del software funciona como su consentimiento para estas prácticas.
3. TÉRMINOS PARA COMPONENTES ESPECÍFICOS.
 - a. Implementaciones de servidor. El software puede contener algunos componentes del servidor de compilación enumerados en BuildServer archivos.TXT o cualquier archivo que aparezca en la BuildServer lista ubicada siguiendo estos términos de licencia de software de Microsoft. Puede copiar e instalar dichos elementos, si se incluyen en el software, en sus máquinas de compilación. Usted y otros miembros de su organización podrían utilizar estos elementos en las máquinas de compilación únicamente con el fin de compilar, crear, verificar y archivar sus aplicaciones o para llevar a cabo pruebas de control de calidad o rendimiento como parte del proceso de compilación.
 - b. Plataformas de Microsoft. El software puede incluir componentes de Microsoft Windows, Microsoft Windows Server, Microsoft SQL Server, Microsoft Exchange, Microsoft Office y Microsoft SharePoint. Estos componentes se rigen por acuerdos independientes y por sus propias políticas de soporte de productos, tal y como se describe en los términos de licencia que se encuentran en el directorio de instalación de dicho componente o en la carpeta "Licencias" que acompaña al software.
 - c. Componentes de terceros. El software puede incluir componentes de terceros con avisos legales independientes o regirse por otros acuerdos, tal y como se describe en el ThirdPartyNotices archivo adjunto al software. Incluso si tales componentes se rigen por otros acuerdos, también se aplican los descargos de responsabilidad y las limitaciones y exclusiones de daños siguientes.
 - d. Administradores de paquetes. El software podría incluir administradores de paquetes, como NuGet, que le ofrece la opción de descargar otros paquetes de software de Microsoft y de terceros para utilizar con la aplicación. Estos paquetes incluyen sus propias licencias y no son

objeto del presente acuerdo. Microsoft no distribuye, otorga licencia ni proporciona ninguna garantía para ningún paquete de terceros.

4. **ÁMBITO DE LA LICENCIA.** El software se cede bajo licencia y no es objeto de venta. Este acuerdo solo le otorga algunos derechos de uso del software. Microsoft se reserva todos los demás derechos. A menos que la legislación aplicable le otorgue más derechos a pesar de esta limitación, solo podrá utilizar el software tal como se permite expresamente en este acuerdo. Al hacerlo, deberá ajustarse a las limitaciones técnicas del software que solo permiten utilizarlo de determinadas formas. Para obtener más información, consulte <https://docs.microsoft.com/en-us/legal/information-protection/software-license-terms#1-installation-and-use-rights>. No podrá
 - eludir las limitaciones técnicas del software;
 - utilizar técnicas de ingeniería inversa, descompilar o desensamblar el software, así como tampoco intentar hacerlo, excepto y en la medida que lo exijan los términos de licencia de terceros que rigen el uso de ciertos componentes de código abierto que se podrían incluir con el software;
 - eliminar, minimizar, bloquear o modificar ninguna notificación de Microsoft o sus proveedores;
 - utilizar el software de ninguna manera que esté en contra de la ley; o
 - compartir, publicar, alquilar o dar el software en préstamo, ni entregarlo como solución hospedada independiente para que otros lo utilicen.
5. **RESTRICCIONES EN MATERIA DE EXPORTACIÓN.** Debe cumplir con todas las leyes y reglamentos, nacionales e internacionales, en materia de exportación que sean de aplicación al software, lo que incluye restricciones en cuanto a destino, usuarios finales y uso final. Para obtener más información sobre las restricciones en materia de exportación, visite (aka.ms/exporting).
6. **SERVICIOS DE SOPORTE TÉCNICO.** Dado que este software se suministra "tal cual", es posible que no se proporcionen servicios de soporte técnico para el mismo.
7. **ACUERDO COMPLETO.** Este acuerdo y los términos aplicables a complementos, actualizaciones, servicios basados en Internet y servicios de soporte técnico que utilice constituyen el acuerdo completo para el software y los servicios de soporte técnico.
8. **LEGISLACIÓN APLICABLE.** Si adquirió el software en los Estados Unidos, la legislación del estado de Washington regirá la interpretación de este acuerdo, así como las reclamaciones por incumplimiento, y la legislación del estado donde usted resida regirá todas las demás reclamaciones. Si adquiere el software en otro país, se aplicará la legislación de dicho país.
9. **DERECHOS DEL CONSUMIDOR, VARIACIONES REGIONALES.** En este acuerdo se describen determinados derechos legales. Puede que usted tenga otros derechos, incluidos derechos

del consumidor, según la legislación de su estado o país. Por separado e independientemente de su relación con Microsoft, también puede tener derechos con respecto a la parte de la que obtuvo el software. Este acuerdo no modifica los otros derechos de los que dispone en virtud de la legislación de su estado, país o región si dicha legislación no lo permite. Por ejemplo, si adquirió el software en una de las regiones que figuran a continuación, o si se aplican leyes obligatorias del país o la región, las siguientes disposiciones se aplican a su caso:

- **Australia.** Tiene garantías legales previstas según la Ley de Consumo australiana y ninguna parte de este acuerdo pretende afectar a dichos derechos.
- **Canadá.** Si adquirió este software en Canadá, para dejar de recibir actualizaciones, puede desactivar la función de actualizaciones automáticas, desconectar su dispositivo de Internet (sin embargo, cuando vuelva a conectarse a Internet el software volverá a comprobar e instalar actualizaciones) o desinstalar el software. La documentación del producto, de haberla, también puede especificar cómo desactivar las actualizaciones para su dispositivo o software específico.
- **Alemania y Austria.**
 - **Garantía.** El software que disponga de la debida licencia funcionará esencialmente como se describe en la documentación de Microsoft que lo acompaña. Sin embargo, Microsoft no otorga ninguna garantía contractual en relación con el software licenciado.
 - **Limitación de responsabilidad.** En caso de comportamiento intencional, negligencia grave, reclamaciones basadas en la Ley de Responsabilidad por Productos, así como en caso de muerte, daños corporales o lesiones físicas, Microsoft es responsable conforme a la legislación.

Sujeto a la cláusula anterior (ii), Microsoft solo será responsable de negligencia leve si Microsoft incumple las obligaciones contractuales materiales, cuyo cumplimiento permite la debida ejecución de este acuerdo, y cuya infracción haría peligrar el propósito de este acuerdo y la confianza de una parte en su cumplimiento (denominadas "obligaciones cardinales"). En otros casos de negligencia leve, Microsoft no será responsable de dicha negligencia.

10EFECTO LEGAL. En este acuerdo se describen determinados derechos legales. Es posible que tenga otros derechos en virtud de las leyes de su estado o país. Este acuerdo no modifica sus derechos de los que dispone en virtud de la legislación de su estado o país si dicha legislación no lo permite. Sin limitación de lo anterior, para Australia, TIENE GARANTÍA LEGALES PREVISTAS SEGÚN LA LEY DE CONSUMO AUSTRALIANA Y NINGUNA PARTE DE ESTOS TÉRMINOS PRETENDE AFECTAR A DICHOS DERECHOS

- 11 RENUNCIA DE GARANTÍA. EL SOFTWARE SE CONCEDE BAJO LICENCIA "TAL CUAL". USTED ASUME EL RIESGO DE UTILIZARLO. MICROSOFT NO PROPORCIONA NINGUNA GARANTÍA EXPRESA, ASÍ COMO TAMPOCO AVALES O CONDICIONES. EN LA MEDIDA EN QUE LO PERMITE LA LEGISLACIÓN LOCAL, MICROSOFT EXCLUYE LAS GARANTÍAS IMPLÍCITAS DE COMERCIALIZACIÓN, IDONEIDAD PARA UN DETERMINADO FIN Y NO INFRACCIÓN.
- 12 LIMITACIÓN Y EXCLUSIÓN DE INDEMNIZACIONES. PUEDE RECUPERAR DE MICROSOFT Y SUS PROVEEDORES ONLY DAÑOS DIRECTOS EE. UU. HASTA 5,00 USD. NO PODRÁ OBTENER COMPENSACIÓN POR NINGÚN OTRO DAÑO, INCLUIDOS DAÑOS CONSECUENTES, PÉRDIDAS DE GANANCIAS, DAÑOS ESPECIALES, DAÑOS INDIRECTOS O DAÑOS INCIDENTALES.

Esta limitación se aplica a (a) nada relacionado con el software, servicios, contenido, (incluido código) en sitios de Internet o aplicaciones de terceros; y (b) reclamaciones por incumplimiento del contrato, incumplimiento de garantía o condición, responsabilidad estricta, negligencia u otros perjuicios en la medida permitida por la ley vigente.

También se aplicará incluso si Microsoft conocía o debería haber conocido la posibilidad de que se produjesen dichos daños. La limitación o exclusión precedente podría no aplicarse a su caso porque el país podría no admitir la exclusión o limitación de daños accidentales, resultantes o de otra índole.

11) Imagen de Docker basada en Windows: 3-ondemand-package.cab microsoft-windows-netfx

TÉRMINOS DE LICENCIA COMPLEMENTARIOS DEL SOFTWARE DE MICROSOFT

MICROSOFT .NET FRAMEWORK 3.5 SP1 PARA SISTEMA OPERATIVO MICROSOFT WINDOWS

Microsoft Corporation (o, en función de donde resida, una de sus filiales) le otorga licencia para este complemento. Si posee licencias para utilizar software de sistema operativo Windows (al que se aplica este complemento) (el "software"), puede utilizar este complemento. No puede utilizarlo si no tiene una licencia para el software. Puede utilizar una copia de este complemento con cada copia de licencia válida del software.

Los siguientes términos de licencia describen términos de uso adicionales para este complemento. Los presentes términos y los términos de licencia para el software se aplican al uso que usted haga del complemento. Si hay un conflicto, se aplican estos términos de licencia complementarios.

AL HACER USO DE ESTE COMPLEMENTO, ACEPTA ESTOS TÉRMINOS. SI NO LOS ACEPTA, NO UTILICE ESTE COMPLEMENTO.

Si cumple estos términos de licencia, tiene los derechos que se indican a continuación.

1. SERVICIOS DE SOPORTE TÉCNICO PARA EL COMPLEMENTO. Microsoft proporciona los servicios de soporte técnico para el presente software que se describen en www.support.microsoft.com/common/international.aspx.
2. PRUEBAS DE BANCO DE PRUEBAS DE MICROSOFT .NET. Este software incluye los componentes .NET Framework, Windows Communication Foundation, Windows Presentation Foundation y Windows Workflow Foundation de los sistemas operativos Windows (Componentes .NET). Puede realizar pruebas internas de banco de pruebas de los componentes de .NET. Puede divulgar los resultados de cualquier prueba de banco de pruebas de los componentes de .NET, siempre que cumpla con las condiciones establecidas en <http://go.microsoft.com/fwlink/?LinkID=66406>.

Sin perjuicio de cualquier otro acuerdo que puede tener con Microsoft, si divulga los resultados de dichas pruebas de banco de pruebas, Microsoft tendrá el derecho a divulgar los resultados de las pruebas de referencia que realice de sus productos que compitan con el componente de .NET correspondiente, siempre que cumpla con las mismas condiciones establecidas en <http://go.microsoft.com/fwlink/?LinkID=66406>.

12) Imagen de Docker base de Windows - dotnet-sdk

(disponible en <https://github.com/dotnet/core/blob/main/LICENSE.TXT>)

La Licencia MIT (MIT)

Copyright (c) Microsoft Corporation

Se concede permiso, libre de cargos, a cualquier persona que obtenga una copia de este software y de los archivos de documentación asociados (el "Software"), para utilizar el Software sin restricción, incluyendo sin limitación los derechos a usar, copiar, modificar, fusionar, publicar, distribuir,

sublicenciar, y/o vender copias del Software, y a permitir a las personas a las que se les proporcione el Software a hacer lo mismo, sujeto a las siguientes condiciones:

El aviso de copyright anterior y este aviso de permiso se incluirán en todas las copias o partes sustanciales del Software.

EL SOFTWARE SE PROPORCIONA "TAL CUAL", SIN GARANTÍAS DE NINGÚN TIPO, EXPLÍCITAS NI IMPLÍCITAS, LO QUE INCLUYE, SIN QUE SIRVA DE LIMITACIÓN, LAS GARANTÍAS DE COMERCIALIZACIÓN, IDONEIDAD PARA UN FIN DETERMINADO Y AUSENCIA DE INFRACCIÓN. LOS AUTORES O TITULARES DE LOS DERECHOS DE AUTOR (COPYRIGHT) NO SERÁN RESPONSABLES EN NINGÚN CASO ANTE NINGUNA RECLAMACIÓN, DAÑO, PERJUICIO NI RESPONSABILIDAD DE NINGÚN OTRO TIPO, YA SEA CONTRACTUAL, EXTRA CONTRACTUAL O DE OTRA ÍNDOLE, QUE SE DERIVEN DEL SOFTWARE O ESTÉN RELACIONADOS CON ÉL, CON SU UTILIZACIÓN O CON OTROS USOS DE ESTE.

AWS CodeBuild Historial de documentos de la Guía del usuario

En la siguiente tabla se describen los cambios importantes en la documentación desde la última versión de AWS CodeBuild. Para recibir notificaciones sobre los cambios en esta documentación, puede suscribirse a una fuente RSS.

- Última versión de la API: 06-10-2016

Cambio	Descripción	Fecha
Contenido actualizado: Imágenes de computación de Lambda	Agregue compatibilidad con Lambda para .NET 8 (al-lambda/aarch64/dotnet8 y) al-lambda/x86_64/dotnet8	8 de mayo de 2024
Contenido actualizado: políticas AWS gestionadas (predefinidas) para AWS CodeBuild	Las AWSCodeBuildReadOnlyAccess políticas AWSCodeBuildAdminAccess AWSCodeBuildDeveloperAccess, y se han actualizado para reflejar el cambio de AWS CodeConnections marca.	30 de abril de 2024
Contenido nuevo: Contraseña o token de acceso de la aplicación Bitbucket	Añade compatibilidad con los tokens de acceso de Bitbucket	11 de abril de 2024
Contenido nuevo: Detección automática de informes	CodeBuild ahora admite la detección automática de informes.	4 de abril de 2024

Contenido nuevo: configura ejecutores de Actions autohospedados GitHub	Añade contenido nuevo para los ejecutores de Actions autohospedados GitHub	2 de abril de 2024
Contenido nuevo: conexiones GitLab	Añada compatibilidad GitLab y conexiones GitHub autogestionadas.	25 de marzo de 2024
Contenido nuevo: añade nuevos eventos de webhook y tipos de filtros	Añade compatibilidad con nuevos eventos (RELEASEDyPRERELEASED) de webhook y tipos de filtros (TAG_NAMEyRELEASE_NAME).	15 de marzo de 2024
Contenido nuevo: agrega un nuevo evento de webhook: PULL_REQUEST_CLOSED	Añadir soporte para un nuevo evento de webhook: PULL_REQUEST_CLOSED	20 de febrero de 2024
Contenido actualizado: Imágenes de Docker proporcionadas por CodeBuild	Agregue soporte para Windows Server Core 2019 () windows-base:2019-3.0	7 de febrero de 2024
Contenido actualizado: Imágenes de Docker proporcionadas por CodeBuild	Añada compatibilidad con nuevos tiempos de ejecución para Amazon Linux 2023 () a12/aarch64/standard/3.0	29 de enero de 2024
Nuevo contenido: Capacidad reservada	CodeBuild ahora admite flotas de capacidad reservada en CodeBuild	18 de enero de 2024

Nuevo tipo de computación	CodeBuild ahora es compatible con un tipo de cómputo Linux XLarge. Para obtener más información, consulte Tipos de computación del entorno de compilación .	8 de enero de 2024
Contenido actualizado: Imágenes de Docker proporcionadas por CodeBuild	Añade compatibilidad con nuevos tiempos de ejecución para Amazon Linux 2 (a12/standard/5.0) y Ubuntu (ubuntu/standard/7.0)	14 de diciembre de 2023
Contenido actualizado: Imágenes de Docker proporcionadas por CodeBuild	Añade compatibilidad con nuevas imágenes de computación de Lambda	8 de diciembre de 2023
Contenido nuevo: computar AWS Lambda	Agregue contenido nuevo para el AWS Lambda cómputo	6 de noviembre de 2023
Contenido nuevo: uso de la sintaxis GitHub Actions en una especificación de compilación	Agrega contenido nuevo para usar la sintaxis de GitHub Actions en una especificación de compilación	6 de julio de 2023
Contenido actualizado: Imágenes de Docker proporcionadas por CodeBuild	Se ha añadido la compatibilidad con Amazon Linux 2 (a12/standard/5.0)	17 de mayo de 2023
Cambios en las políticas administradas para CodeBuild	Los detalles sobre las actualizaciones de las políticas AWS gestionadas ya CodeBuild están disponibles. Para obtener más información, consulte CodeBuild las actualizaciones de las políticas AWS gestionadas .	16 de mayo de 2023

Contenido actualizado: Imágenes de Docker proporcionadas por CodeBuild	Se ha eliminado la compatibilidad con Amazon Linux 2 (a12/standard/3.0) y se ha añadido la compatibilidad con Amazon Linux 2 (a12/standard/corretto8) y Amazon Linux 2 (a12/standard/corretto11)	9 de mayo de 2023
Contenido actualizado: Imágenes de Docker proporcionadas por CodeBuild	Se ha añadido compatibilidad con Ubuntu 22.04 (ubuntu/standard/7.0)	13 de abril de 2023
Contenido actualizado: Imágenes de Docker proporcionadas por CodeBuild	Se ha eliminado la compatibilidad con Ubuntu 18.04 (ubuntu/standard/4.0) y Amazon Linux 2 (a12/aarch64/standard/1.0)	31 de marzo de 2023
Contenido actualizado: eliminar la limitación de VPC	Eliminar la siguiente limitación: si se configura CodeBuild para trabajar con una VPC, no se admite el almacenamiento en caché local. A partir del 28/02/22, la compilación con VPC tardará más, ya que se utilizará una nueva instancia de Amazon EC2 para cada compilación.	1 de marzo de 2023
Contenido actualizado: Imágenes de Docker proporcionadas por CodeBuild	Se ha eliminado la compatibilidad con Ubuntu 18.04 (ubuntu/standard/3.0) y Amazon Linux 2 (a12/standard/2.0)	30 de junio de 2022

[Ejemplo de Amazon ECR: restringir el acceso a las imágenes](#)

Cuando se utilizan CodeBuild las credenciales para extraer una imagen de Amazon ECR, puede restringir el acceso a la imagen a un CodeBuild proyecto específico. Para obtener más información, consulte el [Ejemplo de Amazon ECR](#).

10 de marzo de 2022

[Se agregó compatibilidad con regiones](#)

El tipo de computación ARM_CONTAINER ya se admite en las regiones adicionales siguientes: Asia Pacífico (Seúl), Canadá (centro), Europa (Londres) y Europa (París). Para obtener más información, consulte [Tipos de computación del entorno de compilación](#).

10 de marzo de 2022

[Nueva limitación de VPC](#)

Si se configura CodeBuild para trabajar con una VPC, no se admite el almacenamiento en caché local. A partir del 28/02/22, la compilación con VPC tardará más, ya que se utilizará una nueva instancia de Amazon EC2 para cada compilación.

25 de febrero de 2022

Modo de informe por lotes	CodeBuild ahora le permite seleccionar cómo se envían los estados de creación por lotes al proveedor de origen de un proyecto. Para obtener más información, consulte Modo de informe por lotes .	4 de octubre de 2021
Nuevo tipo de computación	CodeBuild ahora admite un tipo de cómputo ARM pequeño. Para obtener más información, consulte Tipos de computación del entorno de compilación .	13 de septiembre de 2021
Proyectos de compilación pública	CodeBuild ahora te permite hacer que los resultados de compilación de tus proyectos de compilación estén disponibles para el público sin necesidad de acceder a una AWS cuenta. Para obtener más información, consulte Proyectos de compilación pública .	11 de agosto de 2021
Depuración de sesión para compilaciones por lotes	CodeBuild ahora admite la depuración de sesiones para compilaciones por lotes. Para obtener más información, consulte build-graph y build-list .	3 de marzo de 2021

[Límite de compilaciones simultáneas en el nivel de proyecto](#)

CodeBuild ahora permite limitar el número de compilaciones simultáneas de un proyecto de compilación. Para obtener más información, consulte [Configuración del proyecto](#) y [concurrentBuildLimit](#).

16 de febrero de 2021

[Nueva propiedad de especificaciones de compilación: s3-prefix](#)

CodeBuild ahora proporciona la propiedad `s3-prefix` `buildspec` para los artefactos que le permite especificar un prefijo de ruta para los artefactos que se cargan en Amazon S3. Para obtener más información, consulte [s3-prefix](#).

9 de febrero de 2021

[Nueva propiedad de especificación de compilación: on-failure](#)

CodeBuild ahora proporciona la propiedad `on-failure` `buildspec` para las fases de compilación, que le permite determinar qué ocurre cuando una fase de compilación falla. Para obtener más información, consulte [on-failure](#).

9 de febrero de 2021

[Nueva propiedad de especificación de compilación: exclude-paths](#)

CodeBuild ahora proporciona la propiedad `exclude-paths` `buildspec` para los artefactos, que permite excluir las rutas de los artefactos de construcción. Para obtener más información, consulte [exclude-paths](#).

9 de febrero de 2021

Nueva propiedad de especificación de compilación: <code>enable-symlinks</code>	CodeBuild ahora proporciona la propiedad <code>enable-symlinks</code> en <code>buildspec</code> para los artefactos, que permite conservar los enlaces simbólicos en un artefacto ZIP. Para obtener más información, consulte enable-symlinks .	9 de febrero de 2021
Mejora de los nombres de artefacto en las especificaciones de compilación	CodeBuild ahora permite que la propiedad <code>artifacts/name</code> contenga información sobre la ruta. Para obtener más información, consulte name .	9 de febrero de 2021
Informes de cobertura de código	CodeBuild ahora proporciona informes de cobertura de código. Para obtener más información, consulte Informes de cobertura de código .	30 de julio de 2020
Compilaciones por lotes	CodeBuild ahora admite la ejecución simultánea y coordinada de compilaciones de un proyecto. Para obtener más información, consulte Batch builds in CodeBuild .	30 de julio de 2020
Imagen de Windows Server 2019	CodeBuild ahora proporciona una imagen de compilación de Windows Server Core 2019. Para obtener más información, consulte las imágenes de Docker proporcionadas por CodeBuild .	20 de julio de 2020

[Session Manager](#)

CodeBuild ahora le permite pausar una compilación en ejecución y, a continuación, usar el Administrador de AWS Systems Manager sesiones para conectarse al contenido de compilación y ver su estado. Para obtener más información, consulte [Session Manager](#).

20 de julio de 2020

[Tema actualizado](#)

CodeBuild ahora permite especificar un shell para usarlo en sus entornos de compilación en el archivo buildspec. Para obtener más información, consulte [Referencia de especificaciones de compilación](#).

25 de junio de 2020

[Generación de informes de pruebas con marcos de trabajo de pruebas](#)

Se agregaron varios temas que describen cómo generar informes de CodeBuild prueba con varios marcos de prueba. Para obtener más información, consulte [Generación de informes de pruebas con marcos de pruebas](#).

29 mayo de 2020

[Temas actualizados](#)

CodeBuild ahora admite la adición de etiquetas a los grupos de informes. Para obtener más información, consulte [ReportGroup](#).

21 de mayo de 2020

[Compatibilidad con informes de pruebas](#)

CodeBuild La compatibilidad con la elaboración de informes de pruebas ya está disponible de forma general.

21 de mayo de 2020

[Temas actualizados](#)

CodeBuild ahora permite crear filtros de webhook para Github y Bitbucket que activen las compilaciones solo cuando el mensaje principal de confirmación coincide con la expresión especificada. Para obtener más información, consulta el ejemplo de filtro para [solicitudes de GitHub extracción y webhook y el ejemplo de filtro para solicitudes de extracción y webhook para Bitbucket](#).

6 de mayo de 2020

[Temas nuevos](#)

CodeBuild ahora permite compartir recursos de creación, proyectos y grupos de informes. Para obtener más información, consulte [Uso de proyectos compartidos](#) y [Uso de grupos de informes compartidos](#).

13 de diciembre de 2019

Temas nuevos y actualizados

CodeBuild ahora admite la generación de informes de pruebas durante la ejecución de un proyecto de compilación. Para obtener más información, consulte [Trabajar con informes de pruebas](#), [Crear un informe de prueba](#) y [Crear un informe de prueba con la AWS CLI muestra](#).

25 de noviembre de 2019

Tema actualizado

CodeBuild ahora es compatible con los tipos de entorno Arm y GPU de Linux, así como con el tipo de 2xlarge cómputo. Para obtener más información, consulte [Tipos de compilación del entorno de compilación](#).

19 de noviembre de 2019

Temas actualizados

CodeBuild ahora admite los números de compilación en todas las compilaciones, la exportación de variables de entorno y AWS Secrets Manager la integración. Para obtener más información, consulte la información sobre [variables exportadas](#) y [Secrets Manager](#) en [Sintaxis de la especificación de compilación](#).

6 de noviembre de 2019

[Nuevo tema](#)

CodeBuild ahora es compatible con las reglas de notificación. Puede utilizar las reglas de notificación para notificar a los usuarios los cambios importantes en los proyectos de compilación. Para obtener más información, consulte [Crear una regla de notificación](#).

5 de noviembre de 2019

[Temas actualizados](#)

CodeBuild ahora es compatible con los tiempos de ejecución de Android versión 29 y Go versión 1.13. [Para obtener más información, consulta las imágenes de Docker proporcionadas por Buildspec CodeBuild y la sintaxis](#).

10 de septiembre de 2019

[Temas actualizados](#)

Al crear un proyecto, ahora puede elegir la imagen administrada de Amazon Linux 2 (AL2). Para obtener más información, consulte las [imágenes de Docker proporcionadas por CodeBuild](#) y las [versiones en tiempo de ejecución incluidas en el ejemplo del archivo buildspec](#). CodeBuild

16 de agosto de 2019

Tema actualizado	Cuando cree un proyecto, ahora podrá deshabilitar el cifrado de los registros de S3 y, si utiliza un repositorio de origen basado en Git, incluir submódulos de Git. Para obtener más información, consulte Crear un proyecto de compilación en. CodeBuild	8 de marzo de 2019
Nuevo tema	CodeBuild ahora admite el almacenamiento en caché local. Cuando cree una compilación, podrá especificar uno de los cuatro modos de almacenamiento en caché local. Para obtener más información, consulte Crear almacenamiento en caché . CodeBuild	21 de febrero de 2019
Temas nuevos	CodeBuild ahora admite grupos de filtros de webhook para especificar los eventos que desencadenan una compilación. Para obtener más información, consulta Filtrar eventos de GitHub webhook y Filtrar eventos de webhook de Bitbucket .	8 de febrero de 2019

Nuevo tema	La guía CodeBuild del usuario ahora muestra cómo utilizar los CodeBuild con un servidor proxy. Para obtener más información, consulte Uso CodeBuild con un servidor proxy .	4 de febrero de 2019
Temas actualizados	CodeBuild ahora admite el uso de una imagen de Amazon ECR que esté en otra AWS cuenta. Se han actualizado varios temas para reflejar este cambio, como un ejemplo de Amazon ECR para CodeBuild , Crear un proyecto de compilación y Crear un rol de CodeBuild servicio .	24 de enero de 2019
Compatibilidad con registros de Docker privados	CodeBuild ahora admite el uso de una imagen de Docker almacenada en un registro privado como entorno de ejecución. Para obtener más información, consulta Registro privado con AWS Secrets Manager muestra .	24 de enero de 2019

Tema actualizado	CodeBuild ahora admite el uso de un token de acceso para conectarse a los repositorios GitHub (con un token de acceso personal) y Bitbucket (con una contraseña de aplicación). Para obtener más información, consulte la sección Crear un proyecto de compilación (consola) y Uso de tokens de acceso con su proveedor de código fuente .	6 de diciembre de 2018
Tema actualizado	CodeBuild ahora admite nuevas métricas de compilación que miden la duración de cada fase de una compilación. Para obtener más información, consulta CodeBuild CloudWatch las métricas .	15 de noviembre de 2018
Tema de la política de puntos de conexión de VPC	Por CodeBuild ahora, los puntos de enlace de Amazon VPC admiten políticas. Para obtener más información, consulte Crear una política de punto final de VPC para CodeBuild .	9 de noviembre de 2018
Contenido actualizado	Los temas se han actualizado para reflejar la experiencia con la nueva consola.	30 de octubre de 2018

[Ejemplo de Amazon EFS](#)

CodeBuild puede montar un sistema de archivos Amazon EFS durante una compilación mediante los comandos del archivo buildspec de un proyecto. Para obtener más información, consulte el [ejemplo de Amazon EFS para CodeBuild](#).

26 de octubre de 2018

[Webhooks de Bitbucket](#)

CodeBuild ahora admite webhooks cuando utilizas Bitbucket como repositorio. Para obtener más información, consulta un ejemplo de [solicitud de extracción de Bitbucket para](#) CodeBuild

2 de octubre de 2018

[Registros de S3](#)

CodeBuild ahora admite registros de compilación en un bucket de S3. Anteriormente, solo se podían crear CloudWatch registros con Logs. Para obtener más información, consulte [Creación de un proyecto](#).

17 de septiembre de 2018

[Varios orígenes de entrada y varios artefactos de salida](#)

CodeBuild ahora es compatible con proyectos que utilizan más de una fuente de entrada y publican más de un conjunto de artefactos. Para obtener más información, consulte [Ejemplo de múltiples fuentes de entrada y artefactos de entrada y muestra de CodePipelineintegración con CodeBuild múltiples fuentes de entrada y artefactos de salida](#).

30 de agosto de 2018

[Ejemplo de control semántico de versiones](#)

La Guía del CodeBuild usuario incluye ahora un ejemplo basado en casos de uso que demuestra cómo utilizar el control de versiones semántico para crear nombres de artefactos en el momento de la compilación. Para obtener más información, consulte [Ejemplo para utilizar el control de versiones semánticas para asignar nombre a artefactos de compilación](#).

14 de agosto de 2018

[Ejemplo de nuevo sitio web estático](#)

La guía del CodeBuild usuario incluye ahora un ejemplo basado en casos de uso que demuestra cómo alojar los resultados de la compilación en un bucket de S3. El ejemplo utiliza casos de soporte recientes sobre los artefactos de compilación sin cifrar. Para obtener más información, consulte [Crear un sitio web estático con la salida de compilación alojada en un bucket de Amazon S3](#).

14 de agosto de 2018

[Compatibilidad con la anulación de un nombre de artefacto con el control de versiones semánticas](#)

Ahora puede usar el control de versiones semántico para especificar un formato que se CodeBuild utilice para nombrar los artefactos de compilación. Esto resulta útil porque un artefacto de compilación con un nombre de codificación fija sobrescribe artefactos de compilación anteriores que utilicen el mismo nombre de codificación fija. Por ejemplo, si se activa una compilación varias veces al día, ahora puede añadir una marca de tiempo a su nombre de artefactos. Cada nombre de artefacto de compilación es único y no sobrescribe los artefactos de compilaciones anteriores.

7 de agosto de 2018

[Compatibilidad con artefactos de compilación sin cifrar](#)

CodeBuild ahora admite compilaciones con artefactos de compilación no cifrados. Para obtener más información, consulte [Crear un proyecto de compilación \(consola\)](#).

26 de julio de 2018

[Support for Amazon CloudWatch metrics and alarms](#)

CodeBuild ahora ofrece integración con CloudWatch métricas y alarmas. Puede utilizar la CloudWatch consola CodeBuild o para supervisar las compilaciones a nivel de proyecto y de cuenta. Para obtener más información, consulte este tema sobre la [monitorización de compilaciones](#).

19 de julio de 2018

[Compatibilidad con la notificación del estado de una compilación](#)

CodeBuild ahora puedes informar del estado de inicio y finalización de una compilación a tu proveedor de origen. Para obtener más información, consulte [Crear un proyecto de compilación en CodeBuild](#).

10 de julio de 2018

[Variables de entorno agregadas a la CodeBuild documentación](#)

La página [Variables de entorno en entornos de compilación](#) se ha actualizado con las variables de entorno CODEBUILD_BUILD_ID, CODEBUILD_LOG_PATH y CODEBUILD_START_TIME.

9 de julio de 2018

[Compatibilidad con un bloque finally en el archivo de especificación de compilación](#)

La CodeBuild documentación se actualizó con detalles sobre el finally bloque opcional en un archivo de especificaciones de compilación. Los comandos del bloque finally siempre se ejecutan después de los comandos de su bloque de comandos correspondiente. Para obtener más información, consulte [Sintaxis de la especificación de compilación](#).

20 de junio de 2018

[CodeBuild notificaciones de actualización del agente](#)

La CodeBuild documentación se actualizó con detalles sobre cómo puede utilizar Amazon SNS para recibir notificaciones cuando se publiquen nuevas versiones del CodeBuild agente. Para obtener más información, consulte [Recibir notificaciones sobre las nuevas versiones AWS CodeBuild del agente](#).

15 de junio de 2018

Actualizaciones anteriores

En la siguiente tabla, se describen los cambios importantes de cada versión de la Guía del usuario de AWS CodeBuild anteriores a junio de 2018.

Cambio	Descripción	Fecha
Compatibilidad con compilaciones de Windows	CodeBuild ahora admite compilaciones para la plataforma Microsoft Windows Server, incluido un entorno	25 de mayo de 2018

Cambio	Descripción	Fecha
	de compilación preempaquetado para .NET Core 2.0 en Windows. Para obtener más información, consulte Ejemplos de Microsoft Windows para CodeBuild .	
Compatibilidad con la idempotencia de la compilación	Cuando se ejecuta el comando <code>start-build</code> con la AWS Command Line Interface (AWS CLI), se puede especificar que la compilación sea idempotente. Para obtener más información, consulte Ejecutar una compilación (AWS CLI) .	15 de mayo de 2018
Compatibilidad con la anulación de más opciones de un proyecto de compilación	Ahora se pueden anular más opciones de un proyecto de compilación al crear una compilación. Las anulaciones solo se aplican a esa compilación. Para obtener más información, consulte Ejecutar una compilación en AWS CodeBuild .	15 de mayo de 2018
Compatibilidad con puntos de conexión de VPC	Ahora es posible utilizar puntos de enlace de la VPC para mejorar la seguridad de las compilaciones. Para obtener más información, consulte Uso de puntos de enlace de la VPC .	18 de marzo de 2018

Cambio	Descripción	Fecha
Compatibilidad con los disparadores	Ahora puede crear desencadenadores para programar las compilaciones a intervalos regulares. Para obtener más información, consulte Crear desencadenadores de AWS CodeBuild .	28 de marzo de 2018
Documentación de los puntos de enlace de FIPS	Ahora puede obtener información sobre cómo usar el AWS Command Line Interface (AWS CLI) o un AWS SDK para indicar si debe usar uno de CodeBuild los cuatro puntos finales de los estándares federales de procesamiento de información (FIPS). Para obtener más información, consulte Especificación del punto de conexión de AWS CodeBuild .	28 de marzo de 2018
AWS CodeBuild disponible en Asia Pacífico (Bombay), Europa (París) y Sudamérica (São Paulo)	AWS CodeBuild ahora está disponible en las regiones de Asia Pacífico (Bombay), Europa (París) y Sudamérica (São Paulo). Para obtener más información, consulte AWS CodeBuild en la Referencia general de Amazon Web Services.	28 de marzo de 2018

Cambio	Descripción	Fecha
GitHub Soporte para servidores empresariales	CodeBuild ahora puede compilar a partir del código fuente almacenado en un repositorio de GitHub Enterprise Server. Para obtener más información, consulte GitHub Ejemplo de Enterprise Server .	25 de enero de 2018
Compatibilidad con la profundidad del clon de Git	CodeBuild ahora admite la creación de un clon superficial con un historial truncado al número especificado de confirmaciones. Para obtener más información, consulte Creación de un proyecto de compilación .	25 de enero de 2018
Compatibilidad con VPC	Las compilaciones habilitadas para VPC ahora pueden tener acceso a los recursos de su VPC. Para obtener más información, consulte Compatibilidad con VPC .	27 de noviembre de 2017
Compatibilidad con el almacenamiento en caché de dependencias	CodeBuild ahora admite el almacenamiento en caché de dependencias. Esto permite CodeBuild guardar ciertas partes reutilizables del entorno de compilación en la memoria caché y utilizarlas en todas las compilaciones.	27 de noviembre de 2017

Cambio	Descripción	Fecha
Compatibilidad con las insignias de compilación	CodeBuild ahora admite el uso de insignias de compilación, que proporcionan una imagen (insignia) integrable y generada dinámicamente que muestra el estado de la última compilación de un proyecto. Para obtener más información, consulte Ejemplo de insignias de compilación .	27 de noviembre de 2017
AWS Config integración	AWS Config ahora es compatible CodeBuild como AWS recurso, lo que significa que el servicio puede realizar un seguimiento de sus CodeBuild proyectos. Para obtener más información al respecto AWS Config, consulte AWS Config muestra .	20 de octubre de 2017
Reconstruye automáticamente el código fuente actualizado en los GitHub repositorios	Si el código fuente está almacenado en un GitHub repositorio, puedes habilitar AWS CodeBuild la reconstrucción del código fuente cada vez que se introduzca un cambio de código en el repositorio. Para obtener más información, consulte GitHub ejemplo de filtro pull request y webhook .	21 de septiembre de 2017

Cambio	Descripción	Fecha
Nuevas formas de almacenar y recuperar variables de entorno confidenciales o grandes en el almacén de parámetros de Amazon EC2 Systems Manager	Ahora puede utilizar la AWS CodeBuild consola o el AWS CLI para recuperar variables de entorno sensibles o de gran tamaño almacenadas en el almacén de parámetros de Amazon EC2 Systems Manager. Ahora también puede usar la AWS CodeBuild consola para almacenar estos tipos de variables de entorno en el almacén de parámetros de Amazon EC2 Systems Manager. Anteriormente, solo podía recuperar estos tipos de variables de entorno incluyéndolas en una especificación de compilación o ejecutando comandos de compilación para automatizar la AWS CLI. Solo podía almacenar estos tipos de variables de entornos mediante la consola del almacén de parámetros de Amazon EC2 Systems Manager. Para obtener más información, consulte Creación de un proyecto de compilación , Cambiar la configuración de un proyecto de compilación y Ejecutar una compilación .	14 de septiembre de 2017

Cambio	Descripción	Fecha
Compatibilidad con la eliminación de compilaciones	Ahora puede eliminar compilaciones en AWS CodeBuild. Para obtener más información, consulte Eliminación de compilaciones .	31 de agosto de 2017
Forma actualizada de recuperar variables de entorno confidenciales o grandes almacenadas en el almacén de parámetros de Amazon EC2 Systems Manager mediante una especificación de compilación	AWS CodeBuild ahora facilita el uso de una especificación de compilación para recuperar variables de entorno sensibles o de gran tamaño almacenadas en el almacén de parámetros de Amazon EC2 Systems Manager. Anteriormente, solo podía recuperar estos tipos de variables de entorno ejecutando comandos de compilación para automatizar la AWS CLI. Para obtener más información, consulte la asignación de <code>parameter-store</code> en Sintaxis de buildspec .	10 de agosto de 2017
AWS CodeBuild es compatible con Bitbucket	CodeBuild ahora puede compilar a partir del código fuente almacenado en un repositorio de Bitbucket. Para obtener más información, consulte Creación de un proyecto de compilación y Ejecutar una compilación .	10 de agosto de 2017

Cambio	Descripción	Fecha
AWS CodeBuild disponible en EE. UU. Oeste (norte de California), Europa (Londres) y Canadá (centro)	AWS CodeBuild ahora está disponible en las regiones de EE. UU. Oeste (Norte de California), Europa (Londres) y Canadá (Centro). Para obtener más información, consulte AWS CodeBuild en la Referencia general de Amazon Web Services.	29 de junio de 2017
Nombres y ubicaciones alternativos de archivos buildspec admitidos	Ahora puede especificar un nombre o ubicación alternativos de un archivo buildspec para usarlos en un proyecto de compilación, en lugar de un archivo buildspec predeterminado llamado <code>buildspec.yml</code> en la raíz del código fuente. Para obtener más información, consulte Nombre de archivo y ubicación de almacenamiento de buildspec .	27 de junio de 2017
Ejemplo de notificaciones de compilación actualizado	CodeBuild ahora ofrece soporte integrado para crear notificaciones a través de Amazon CloudWatch Events y Amazon Simple Notification Service (Amazon SNS). El Ejemplo de notificaciones de compilación anterior se ha actualizado para mostrar este nuevo comportamiento.	22 de junio de 2017

Cambio	Descripción	Fecha
Se ha añadido un ejemplo de Docker en imagen personalizada	Se ha añadido un ejemplo que muestra cómo utilizar CodeBuild una imagen de compilación de Docker personalizada para crear y ejecutar una imagen de Docker. Para obtener más información, consulte la Ejemplo de Docker en una imagen personalizada .	7 de junio de 2017
Busca el código fuente para las solicitudes de incorporación de cambios GitHub	Cuando ejecutas una compilación CodeBuild que se basa en el código fuente almacenado en un GitHub repositorio, ahora puedes especificar un ID de solicitud de GitHub extracción para compilar. También puede especificar un ID de confirmación, un nombre de ramificación o un nombre de etiqueta en su lugar. Para obtener más información, consulte el valor de Versión de fuente en Ejecutar una compilación (consola) o el valor de sourceVersion en Ejecutar una compilación (AWS CLI) .	6 de junio de 2017

Cambio	Descripción	Fecha
Versión de especificación de compilación actualizada	Se ha publicado una nueva versión del formato buildspec . La versión 0.2 aborda el problema de CodeBuild ejecutar cada comando de compilación en una instancia independiente del shell predeterminado. También, en la versión 0.2, <code>environment_variables</code> ahora se llama <code>env</code> y <code>plaintext</code> ahora se llama <code>variables</code> . Para obtener más información, consulte Referencia de especificación de compilación para CodeBuild .	9 de mayo de 2017
Los archivos Docker para imágenes de compilación están disponibles en GitHub	Las definiciones de muchas de las imágenes de compilación que proporciona AWS CodeBuild están disponibles en Dockerfiles en GitHub. Para obtener más información, consulte la columna "Definición" de la tabla de Imágenes de Docker proporcionadas por CodeBuild .	2 de mayo de 2017

Cambio	Descripción	Fecha
AWS CodeBuild disponible en Europa (Fráncfort), Asia Pacífico (Singapur), Asia Pacífico (Sídney) y Asia Pacífico (Tokio)	AWS CodeBuild ya está disponible en las regiones de Europa (Fráncfort), Asia Pacífico (Singapur), Asia Pacífico (Sídney) y Asia Pacífico (Tokio). Para obtener más información, consulte AWS CodeBuild en la Referencia general de Amazon Web Services.	21 de marzo de 2017
CodePipeline soporte de acciones de prueba para CodeBuild	Ahora puedes añadir a una canalización CodePipeline una acción de prueba que utilice CodeBuild. Para obtener más información, consulte Agregar una acción de prueba de CodeBuild a una canalización (consola de CodePipeline) .	8 de marzo de 2017
Los archivos buildspec permiten obtener la salida de la compilación de los directorios de nivel superior seleccionados	Los archivos Buildspec ahora le permiten especificar directorios individuales de nivel superior cuyo contenido puede indicar que se incluya en los artefactos de salida de CodeBuild la compilación. Para ello utilice la asignación <code>base-directory</code> . Para obtener más información, consulte Sintaxis de buildspec .	8 de febrero de 2017

Cambio	Descripción	Fecha
VARIABLES DE ENTORNO INTEGRADAS	AWS CodeBuild proporciona variables de entorno integradas adicionales para que las utilices en tus compilaciones. Estas variables de entorno describen la entidad que inició la compilación, la dirección URL del repositorio de código fuente, el ID de versión de código fuente, etc. Para obtener más información, consulte Variables de entorno en los entornos de compilación .	30 de enero de 2017
AWS CodeBuild disponible en US East (Ohio)	AWS CodeBuild ya está disponible en la región EE.UU. Este (Ohio). Para obtener más información, consulte AWS CodeBuild en la Referencia general de Amazon Web Services.	19 de enero de 2017

Cambio	Descripción	Fecha
Información sobre el funcionamiento del shell y los comandos	CodeBuild ejecuta cada comando que especifiques en una instancia independiente del shell predeterminado de un entorno de compilación. Este comportamiento predeterminado puede producir algunos efectos secundarios imprevistos para los comandos. Le recomendamos algunos enfoques para solucionar este comportamiento predeterminado si fuera necesario. Para obtener más información, consulte Intérpretes de comandos y comandos de los entornos de compilación .	9 de diciembre de 2016
Información sobre las variables de entorno	CodeBuild proporciona varias variables de entorno que puedes usar en tus comandos de compilación. También puede definir sus propias variables de entorno. Para obtener más información, consulte Variables de entorno en los entornos de compilación .	7 de diciembre de 2016
Tema sobre solución de problemas	Ahora dispone de información sobre solución de problemas. Para obtener más información, consulte Solución de problemas AWS CodeBuild .	5 de diciembre de 2016

Cambio	Descripción	Fecha
Versión inicial del complemento de Jenkins	Esta es la versión inicial del complemento de CodeBuild Jenkins. Para obtener más información, consulte Uso de AWS CodeBuild con Jenkins .	5 de diciembre de 2016
Versión inicial de la Guía de usuario	Esta es la versión inicial de la Guía del CodeBuild usuario.	1 de diciembre de 2016

Glosario de AWS

Para ver la terminología más reciente de AWS, consulte el [Glosario de AWS](#) en la Referencia de Glosario de AWS.

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