Guía del usuario de

AWS CodeBuild



Versión de API 2016-10-06

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS CodeBuild: Guía del usuario de

Copyright © 2026 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
Inscríbase en una Cuenta de AWS 1
¿Cómo ejecutar CodeBuild 2
Precios para CodeBuild 3
¿Cómo puedo empezar CodeBuild? 3
Conceptos 4
¿Cómo CodeBuild funciona 4
Siguientes pasos 5
Introducción 7
Primeros pasos con la consola 7
Paso 1: Crear el código fuente 8
Paso 2: creación del archivo de especificación de compilación 11
Paso 3: creación de dos buckets de S3 13
Paso 4: cargar el código fuente y el archivo buildspec 14
Paso 5: crear el proyecto de compilación 16
Paso 6: ejecutar la compilación 18
Paso 7: ver información resumida sobre la compilación 18
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
Paso 1: Crear el código fuente 23
Paso 2: creación del archivo de especificación de compilación 26
Paso 3: creación de 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
Ejemplos basados en casos de uso 45

Cross-service muestras 46
Ejemplo de Amazon ECR 47
Ejemplo de Amazon EFS 54
AWS CodePipeline muestras 60
AWS Config muestra 72
Ejemplo de notificaciones de compilación 73
Ejemplo de insignias de compilación 89
Creación de un proyecto de compilación con insignias de compilación 90
Acceso AWS CodeBuild crea insignias 93
Publica insignias de CodeBuild compilación 94
CodeBuild estados de las insignias 94
Ejemplo para probar un informe 94
Ejecución del ejemplo para probar un informe 95
Muestras de Docker para CodeBuild 102
Ejemplo de Docker en una imagen personalizada 102
Ejemplo de servidor de compilación de imagen de Docker 105
Ejemplo de compilación de Windows Docker 108
Ejemplo de publicación de una imagen de Docker en Amazon ECR 111
Registro privado con AWS Secrets Manager muestra 120
Alojar la salida de la compilación en un bucket de S3 124
Ejemplo con varias entradas y salidas 127
Creación de un proyecto de compilación con varias entradas y salidas 128
Creación de un proyecto sin un origen 131
Versiones del entorno de ejecución en el ejemplo del archivo buildspec 131
Actualización de la versión del tiempo de ejecución en el archivo de especificación de compilación 132
Especificación de dos entornos de ejecución 137
Ejemplo de versión de origen 141
Especifique una versión de GitHub repositorio con un ID de confirmación 142
Especifique una versión de GitHub repositorio con una referencia y un ID de confirmación . 144
Third-party muestras de repositorios de origen 145
Ejecución de ejemplo de Bitbucket 146
Ejecute el ejemplo de GitHub Enterprise Server 152
Ejecuta el ejemplo del GitHub filtro de solicitudes de extracción y webhook 158
Tutorial: La firma de código de Apple con Fastlane al CodeBuild usar S3 para almacenar certificados 163

Tutorial: Cómo firmar código de Apple con Fastlane GitHub para CodeBuild almacenar certificados ... 169
Establecimiento de nombres de artefacto en el momento de la compilación ... 176
Ejecución de ejemplos de Windows ... 179
Ejecución de ejemplos de Windows ... 179
Estructura de directorios ... 181
F # y .NET Framework ... 181
Visual Basic y .NET Framework ... 181
Archivos ... 181
F # y .NET Framework ... 181
Visual Basic y .NET Framework ... 186
Planificación de una compilación ... 200
Referencia de la especificación de compilación ... 203
Nombre de archivo y ubicación de almacenamiento de buildspec ... 203
Sintaxis de buildspec ... 204
versión ... 207
run-as ... 208
env ... 208
proxy ... 214
phases ... 214
informes ... 219
artefactos ... 221
cache ... 227
Ejemplo de un archivo buildspec ... 229
Versiones de buildspec ... 232
Referencia de especificaciones de compilación por lotes ... 233
batch ... 233
batch/build-gráfico ... 234
batch/build-lista ... 237
batch/build-matriz ... 240
batch/build-abanico ... 242
Referencia de entornos de compilación ... 244
Imágenes de Docker proporcionadas por CodeBuild ... 245
Obtención de la lista de imágenes de Docker actuales ... 245
Imágenes de computación de EC2 ... 246
Imágenes de computación de Lambda ... 248

CodeBuild Imágenes obsoletas ... 254
Tiempos de ejecución disponibles ... 255
Versiones de tiempo de ejecución ... 281
Modos y tipos de computación del entorno de compilación ... 285
Acerca de computación ... 285
Acerca de los tipos de entorno de capacidad reservada ... 286
Acerca de los tipos de entorno bajo demanda ... 352
Intérpretes de comandos y comandos de los entornos de compilación ... 365
Variables de entorno en los entornos de compilación ... 366
Tareas en segundo plano en entornos de compilación ... 372
Proyectos de compilación ... 374
Creación de un proyecto de compilación ... 374
Requisitos previos ... 375
Creación de un proyecto de compilación (consola) ... 375
Cree un proyecto de compilación (AWS CLI) ... 398
Crea un proyecto de compilación (AWS SDK) ... 419
Cree un proyecto de compilación (CloudFormation) ... 419
Creación de una regla de notificación ... 419
Cambio de la configuración del proyecto de compilación ... 422
Cambiar la configuración de un proyecto de compilación (consola) ... 423
Cambia la configuración de un proyecto de compilación (AWS CLI) ... 448
Cambiar la configuración de un proyecto de construcción (AWS SDK) ... 449
Uso de varios tokens de acceso ... 449
Paso 1: Crear un secreto de Secrets Manager o un AWS CodeConnections conexión ... 450
Paso 2: Otorgar acceso al rol de IAM CodeBuild del proyecto a los secretos de Secrets Manager ... 450
Paso 3: Configurar Secrets Manager o AWS CodeConnections tokens ... 452
Opciones de configuración adicionales ... 457
Eliminación de proyectos de compilación ... 460
Eliminar un proyecto de compilación (consola) ... 460
Eliminar un proyecto de compilación (AWS CLI) ... 461
Eliminar un proyecto de compilación (AWS SDK) ... 461
Obtención de direcciones URL de proyectos de compilación públicos ... 461
Uso compartido de proyectos de compilación ... 463
Uso compartido de un proyecto ... 463
Servicios relacionados ... 466

Acceso a proyectos compartidos 467
Cómo dejar de compartir un proyecto compartido 467
Identificación de un proyecto compartido 468
Permisos de los proyectos compartidos 468
Etiquetado de proyectos de compilación 468
Agregar una etiqueta a un proyecto 469
Consultar etiquetas de un proyecto 471
Editar etiquetas de un proyecto 472
Quitar una etiqueta de un proyecto 473
Uso de ejecutores 474
GitHub Acciones 475
GitLab corredores 499
Ejecutor de Buildkite 514
Uso de webhooks 536
Prácticas recomendadas para utilizar webhooks 537
Eventos de webhooks de Bitbucket 538
GitHub webhooks globales y organizacionales 552
GitHub webhooks manuales 559
GitHub eventos de webhook 561
GitLab webhooks grupales 578
GitLab webhooks manuales 584
GitLab eventos de webhook 585
Webhooks manuales de Buildkite 600
Aprobación de comentarios de solicitudes de extracción 602
Visualización de los detalles de un proyecto de compilación 608
Ver los detalles de un proyecto de compilación (consola) 608
Ver los detalles de un proyecto de compilación (AWS CLI) 609
Ver los detalles de un proyecto de compilación (AWS SDK) 611
Visualización de los nombres de los proyectos de compilación 611
Ver una lista de nombres de proyectos de compilación (consola) 612
Ver una lista de nombres de proyectos de compilación (AWS CLI) 612
Vea una lista de nombres de proyectos de compilación (AWS SDK) 614
Compilaciones 615
Ejecución de compilaciones de forma manual 616
Ejecución de una compilación de forma local 617
Ejecutar una compilación (consola) 620

Ejecute una compilación (AWS CLI) 621
Ejecute una compilación por lotes (AWS CLI) 628
Iniciar la ejecución de compilaciones automáticamente (AWS CLI) 630
Detener la ejecución de compilaciones automáticamente (AWS CLI) 631
Ejecuta una compilación (AWS SDK) 632
Ejecución de compilaciones en la computación de Lambda 632
Qué herramientas y tiempos de ejecución se incluirán en las imágenes seleccionadas de Docker del entorno de ejecución en las que se ejecutan AWS Lambda? 633
¿Qué pasa si la imagen seleccionada no incluye las herramientas que necesito? 633
Qué regiones son compatibles AWS Lambda computar en CodeBuild? 634
Limitaciones de AWS Lambda cálculo 634
Implemente una función de Lambda AWS SAM con CodeBuild Lambda Java 635
Cree una aplicación React de una sola página con CodeBuild Lambda Node.js 639
Actualice la configuración de una función de Lambda con Lambda Python CodeBuild 642
Ejecución de compilaciones en flotas de capacidad reservada 646
Creación de una flota de capacidad reservada 648
Prácticas recomendadas 650
¿Puedo compartir una flota de capacidad reservada en varios CodeBuild proyectos? 650
¿Cómo funciona el trabajo de computación basado en atributos? 651
¿Puedo especificar manualmente una instancia de Amazon EC2 para mi flota? 651
¿Qué regiones admiten flotas con capacidad reservada? 651
¿Cómo se puede configurar una flota de macOS con capacidad reservada? 652
¿Cómo puedo configurar una imagen de máquina de Amazon (AMI) personalizada para una flota de capacidad reservada? 653
Limitaciones de las flotas de capacidad reservada 655
Propiedades de flota con capacidad reservada 655
Muestras de capacidad reservada 660
Ejecución de compilaciones por lotes 662
Rol de seguridad 662
Tipos de compilación por lotes 662
Modo de informe por lotes 667
Más información 667
Ejecución de pruebas paralelas 667
Soporte en AWS CodeBuild 669
Habilitación de la ejecución de pruebas paralelas en compilaciones por lotes 671
Utilice el comando codebuild-tests-run de la CLI 672

Utilice el comando codebuild-glob-search de la CLI 675
Acerca de la división de pruebas 677
Combinación automática de informes de compilación individuales 678
Ejemplos de ejecución de pruebas paralelas 680
Almacenamiento de las compilaciones en caché 691
Almacenamiento en caché de Amazon S3 691
Almacenamiento en la caché local 698
Especificación de una caché local 700
Depuración de compilaciones 702
Depure compilaciones con sandbox CodeBuild 702
Depuración de compilaciones con Administrador de sesiones 703
Depure compilaciones con sandbox CodeBuild 703
Depuración de compilaciones con Administrador de sesiones 733
Eliminación de compilaciones 738
Eliminación de compilaciones (AWS CLI) 738
Eliminar compilaciones (AWS SDK) 739
Cómo reintentar hacer una compilación de forma manual 740
Cómo reintentar hacer una compilación de forma manual (consola) 740
Vuelva a intentar una compilación manualmente (AWS CLI) 740
Vuelva a intentar una compilación manualmente (AWS SDKs) 741
Cómo reintentar hacer una compilación de forma automática 741
Cómo reintentar hacer una compilación de forma automática (consola) 742
Cómo reintentar hacer una compilación de forma automática (AWS CLI) 742
Vuelva a intentar compilar automáticamente (AWS SDK) 743
Detención de compilaciones 743
Detener una compilación (consola) 743
Detener una compilación (AWS CLI) 744
Detener una compilación (AWS SDKs) 745
Detención de las compilaciones por lotes 745
Detener una compilación por lotes (consola) 745
Detener una compilación por lotes (AWS CLI) 746
Detener una compilación por lotes (AWS SDKs) 746
Cómo desencadenar compilaciones de forma automática 746
Crear desencadenadores de compilación 747
Editar desencadenadores de compilación 750
Ver detalles de las compilaciones 754

Ver detalles de las compilaciones (consola) 754
Ver detalles de las compilaciones (AWS CLI) 755
Ver los detalles de la compilación (AWS SDK) 755
Transiciones de fases de compilación 756
Visualización de los ID de compilación 756
Ver una lista de identificadores de compilación (consola) 756
Ver una lista de identificadores de compilación (AWS CLI) 757
Ver una lista de identificadores de compilación por lotes (AWS CLI) 758
Ver una lista de identificadores de compilación (AWS SDK) 760
Visualización de los ID de compilación para un proyecto de compilación 760
Ver una lista de identificadores de compilación de un proyecto de compilación (consola) 760
Consulta una lista de los identificadores de compilación de un proyecto de compilación (AWS CLI) 761
Consulta una lista de los identificadores de compilación por lotes de un proyecto de compilación (AWS CLI) 762
Consulta una lista de los identificadores de compilación de un proyecto de compilación (AWS SDKs) 764
Informes de pruebas 765
Creación de informes de pruebas 766
Creación de informes de cobertura de código 768
Crear un informe de cobertura de código 768
Detección de informes de forma automática 770
Configuración de la detección automática de informes mediante la consola 771
Configuración de la detección automática de informes mediante variables de entorno del proyecto 771
Grupos de informes 772
Crear un grupo de informes 773
Nomenclatura de grupos de informes 779
Uso compartido de grupos de informes 780
Especificar archivos de prueba 786
Especificar comandos de prueba 787
Etiquetar un grupo de informes 787
Actualizar un grupo de informes 794
Marcos de pruebas 797
Configuración de Jasmine 797
Configuración de Jest 800

Configuración de pytest ... 801
Configuración de RSpec ... 802
Consultar los informes de pruebas ... 803
Consultar los informes de pruebas de una compilación ... 803
Consultar los informes de pruebas de un grupo de informes ... 804
Vea los informes de las pruebas en su AWS inscrita ... 804
Permisos de informes de prueba ... 804
Rol de IAM para informes de pruebas ... 805
Permisos para las operaciones de informes de pruebas ... 807
Ejemplos de permisos de informes de pruebas ... 807
Estados de un informe de pruebas ... 808
Compatibilidad con VPC ... 810
Casos de uso ... 810
Prácticas recomendadas para las VPC ... 811
Limitaciones de las VPC ... 812
Permita el acceso de Amazon VPC a sus proyectos CodeBuild ... 812
Solución de problemas de la configuración de la VPC ... 813
Uso de puntos de enlace de la VPC ... 814
Antes de crear los puntos de enlace de la VPC ... 814
Cree puntos de enlace de VPC para CodeBuild ... 815
Cree una política de puntos de conexión de VPC para CodeBuild ... 815
Utilice un servidor proxy CodeBuild gestionado ... 816
Configuración de un proxy administrado para las flotas de capacidad reservada ... 817
Gestione una flota de capacidad CodeBuild reservada ... 818
Uso de un servidor proxy ... 818
Configure los componentes necesarios para ejecutarse CodeBuild en un servidor proxy ... 819
Se ejecuta CodeBuild en un servidor proxy explícito ... 822
Se ejecuta CodeBuild en un servidor proxy transparente ... 827
Ejecución de un administrador de paquetes y otras herramientas en un servidor proxy ... 828
CloudFormation Plantilla de VPC ... 830
Registro y supervisión ... 837
Registra las llamadas a CodeBuild la API ... 837
Acerca de AWS CodeBuild información en CloudTrail ... 837
Acerca de AWS CodeBuild entradas de archivos de registros ... 839
Supervisión de compilaciones ... 841
CloudWatch métricas ... 842

CloudWatch métricas de utilización de recursos 844
CloudWatch dimensiones 846
CloudWatch alarmas 846
Ver CodeBuild métricas 847
Vea las métricas CodeBuild de utilización de los recursos 849
Cree CodeBuild alarmas en CloudWatch 853
Seguridad 855
Protección de datos 856
Cifrado de datos 858
Administración de claves 859
Privacidad de tráfico 859
Identity and Access Management 859
Información general sobre la administración del acceso 860
Uso de políticas basadas en identidades 864
AWS CodeBuild referencia de permisos 894
Uso de etiquetas para controlar el acceso a AWS CodeBuild recursos 901
Visualización de recursos en la consola 905
Validación de conformidad 906
Resiliencia 907
Seguridad de infraestructuras 907
Acceso al proveedor de fuentes 907
Creación y almacenamiento de un token en un secreto de Secrets Manager 908
GitHub y acceso a GitHub Enterprise Server 911
Acceso de Bitbucket 923
GitLab acceder 934
Cross-service prevención confusa de diputados 941
Temas avanzados 944
Permita a los usuarios interactuar con CodeBuild 944
CodeBuild Permitir interactuar con otros AWS servicios 952
Cifrado de las salidas de compilación 960
Interactúe con el CodeBuild uso del AWS CLI 962
Referencia de la línea de comandos 963
AWS Referencia de SDK y herramientas 964
compatible AWS SDK y herramientas para AWS CodeBuild 965
Trabajando con AWS SDK 965
Especifique el CodeBuild punto final 967

Especifique el AWS CodeBuild punto final (AWS CLI) 967
Especifique el AWS CodeBuild punto final (AWS SDK) 968
Úselo CodeBuild con CodePipeline 970
Requisitos previos 971
Crear una canalización (consola) 973
Creación de una canalización (AWS CLI) 978
Adición de una acción de compilación 983
Adición de una acción de prueba 987
Úselo CodeBuild con Codecov 991
Integrar Codecov en un proyecto de compilación 991
Úselo CodeBuild con Jenkins 994
Configuración de Jenkins 995
Instalación del complemento 995
Uso del complemento 995
Úselo CodeBuild con aplicaciones sin servidor 997
Recursos relacionados 998
Avisos de terceros 998
1) Imagen de Docker base - windowsservercore 999
2) Imagen de Docker base de Windows - Choco 1000
3) Imagen de Docker base de Windows - git --versión 2.16.2 1000
4) Imagen de Docker base de Windows - microsoft-build-tools --versión 15.0.26320.2 1001
5) Imagen de Docker base de Windows - nuget.commandline --version 4.5.1 1005
7) Imagen de Docker base de Windows - netfx-4.6.2-devpack 1005
8) Imagen de Docker base de Windows - visualfsharptools, v 4.0 1007
9) Imagen de Docker base de Windows - netfx-pcl-reference-assemblies-4.6 1008
10) Imagen de Docker base de Windows - visualcppbuildtools, v 14.0.25420.1 1011
11) Imagen de Docker base de Windows - microsoft-windows-netfx3-ondemand-package.cab 1016
12) Imagen de Docker base de Windows - dotnet-sdk 1017
Utilice las claves de CodeBuild condición como variables de rol del servicio de IAM 1017
AWS CodeBuild claves de estado 1018
Aplicación de configuración de conectividad de VPC a proyectos y flotas 1019
Evitación de modificaciones no autorizadas en buildspec de un proyecto 1019
Restricción de tipos de computación para compilaciones 1020
Control de configuración de variables de entorno 1021
Uso de variables en nombres de claves de condición 1022

Comprobación de la existencia de atributos en solicitudes de API 1023
Ejemplos de código 1024
Conceptos básicos 1024
Acciones 1025
API de compilación pública 1042
Acciones de construcción pública 1042
DescribeBuildBatchesForPublicProject 1043
DescribeBuildsForPublicProject 1046
GetCloudWatchLogsForPublicBuild 1051
GetPresignedUrlsForPublicBuild 1054
GetPublicBuild 1057
GetPublicBuildBatch 1061
GetPublicProject 1066
Tipos de datos de construcción pública 1069
BuildBatchForDescribeBuildBatchesPublic 1071
PublicBuild 1073
PublicBuildArtifacts 1077
PublicBuildBatch 1078
PublicBuildGroup 1082
PublicBuildSummary 1084
PublicLogsStatus 1086
PublicProject 1087
PublicProjectArtifacts 1090
PublicProjectBuildBatchConfig 1091
PublicProjectEnvironment 1092
PublicProjectSource 1094
PublicWebhook 1096
Descargable para S3 1098
Resolución de problemas 1100
Apache Maven crea artefactos de referencia en un repositorio incorrecto 1102
De forma predeterminada, los comandos de compilación se ejecutan como usuario raíz 1103
Las compilaciones pueden fallar si los nombres de los archivos no tienen el signo «-U.S». caracteres que no pertenecen al inglés (Estados Unidos) 1104
Las compilaciones pueden fallar al obtener parámetros del almacén de parámetros de Amazon EC2 1105
No se puede acceder al filtro de sucursales de la consola CodeBuild 1106

No se puede ver si la compilación se ha realizado correctamente o no 1106
No se ha notificado el estado de la compilación al proveedor de fuentes 1107
No se puede encontrar ni seleccionar la imagen base de la plataforma Windows Server Core 2019 1107
Comandos anteriores de los archivos buildspec no reconocidos por comandos más recientes 1108
Error: "Access denied" (Acceso denegado) al intentar descargar la caché 1108
Aparece el error "BUILD_CONTAINER_UNABLE_TO_PULL_IMAGE" al usar una imagen de compilación personalizada 1109
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" 1110
Error: "Cannot connect to the Docker daemon" (No se puede conectar al demonio de Docker) al ejecutar una compilación 1110
Error: "no CodeBuild está autorizado a ejecutar: sts:AssumeRole" al crear o actualizar un proyecto de compilación 1112
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» 1113
Error: "Failed to upload artifacts: Invalid arn (Error al cargar artefactos: arn no válido)" al ejecutar una compilación 1113
Error: «Falló el clon de Git: no se puede acceder `'URL de tu repositorio'`: Problema con el certificado SSL: certificado autofirmado» 1113
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 1114
Error: "Esta imagen de compilación requiere seleccionar al menos una versión de tiempo de ejecución". 1115
Error: aparecer el error "QUEUED: INSUFFICIENT_SUBNET" cuando se produce un error en una compilación de una cola de compilaciones 1116
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» 1116
Error: «No se puede descargar el certificado de S3. AccessDenied» 1117
Error: "Unable to locate credentials" (No se encuentran credenciales) 1117
RequestError error de tiempo de espera cuando se ejecuta CodeBuild en un servidor proxy .. 1119
El shell de Bourne (sh) debe existir en las imágenes de compilación 1120

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 1121
Error: «No se puede verificar JobWorker la identidad» 1121
No se ha podido iniciar la compilación 1121
Acceder a GitHub los metadatos de las compilaciones almacenadas en caché local 1122
AccessDenied: El propietario del bucket del grupo de informes no coincide con el propietario del bucket de S3... 1122
Error: «Sus credenciales carecen de uno o más ámbitos de privilegio obligatorios» al crear un proyecto con CodeBuild AWS CodeConnections 1123
Error: “Sorry, no terminal at all requested - can't get input” al compilar con el comando de instalación de Ubuntu 1124
Las compilaciones de Bitbucket fallan debido a errores de autenticación tras usar Secrets Manager OAuth 1125
Cuotas 1126
Cuotas de servicio 1126
Otros límites 1133
Proyectos de compilación 1133
Compilaciones 1133
Flotas de computación 1134
Informes 1135
Tags 1135
Historial de revisión 1137
Actualizaciones anteriores 1164
..... mclxxviii

¿Qué es? AWS CodeBuild?

AWS CodeBuild es un servicio de construcción en la nube totalmente gestionado. 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.

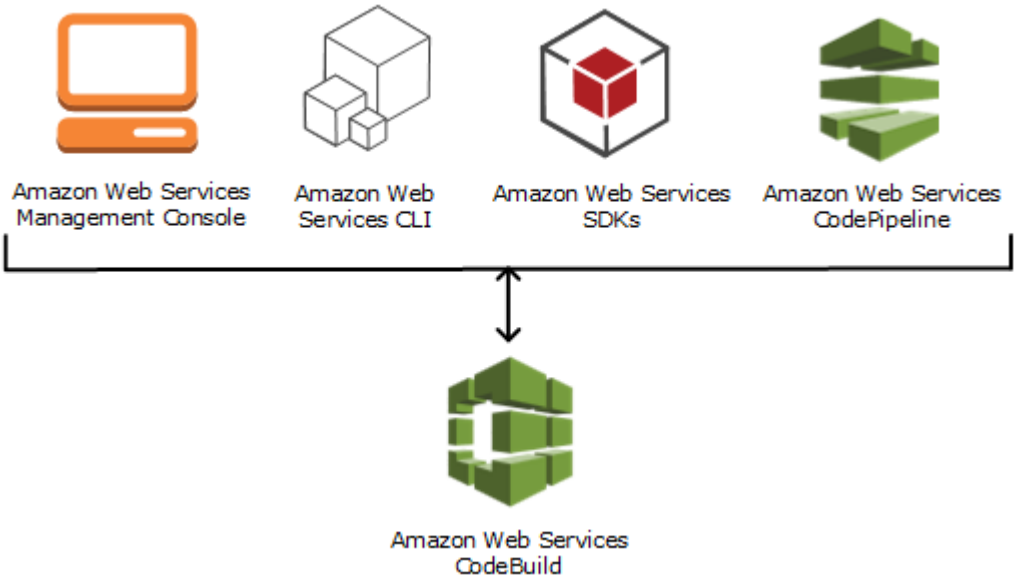
Para obtener más información, consulte AWS CodeBuild.

Inscríbase en una Cuenta de AWS

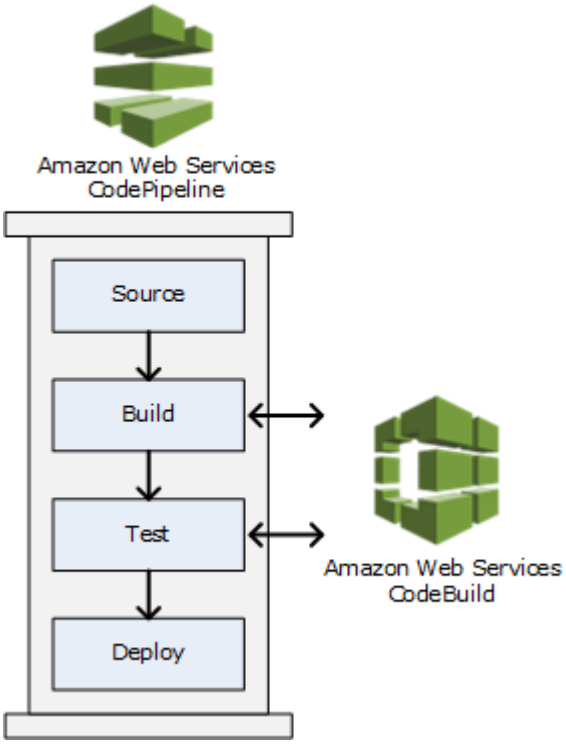
Para empezar AWS, necesitas un Cuenta de AWS. Hay dos formas de inscribirse en AWS. Si desea tener un control total sobre la configuración de la cuenta y el lugar donde se almacena su contenido, o si tiene cargas de trabajo reguladas, como HIPAA o FedRAMP, utilice Sign up for (avanzado). AWS Si es un usuario nuevo y desea utilizar un inicio de sesión que ya tiene para iniciar sesión AWS y crear un AWS entorno con los valores predeterminados preconfigurados en Select, utilice Sign up for. Regiones de AWSAWS

¿Cómo ejecutar CodeBuild

Puedes usar la AWS CodePipeline consola AWS CodeBuild o para correr CodeBuild. También puedes automatizar la ejecución de CodeBuild mediante AWS Command Line Interface (AWS CLI) o los AWS SDK.



Como se muestra en el siguiente diagrama, puedes añadirlas CodeBuild como acción de compilación o prueba a la fase de creación o prueba de una canalización. AWS CodePipeline 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, a continuación, añadir una acción de CodeBuild compilación o prueba, consulteÚselo CodeBuild con CodePipeline. 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 información, consulte Precios de CodeBuild .

¿Cómo puedo empezar CodeBuild?

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

1. Obtenga más información CodeBuild leyendo la información enConceptos.

2. Experimente con CodeBuild un escenario de ejemplo siguiendo las instrucciones dePrimeros pasos con la consola.
3. CodeBuild Úselo en sus propios escenarios siguiendo las instrucciones dePlanificación de una compilación.

AWS CodeBuild conceptos

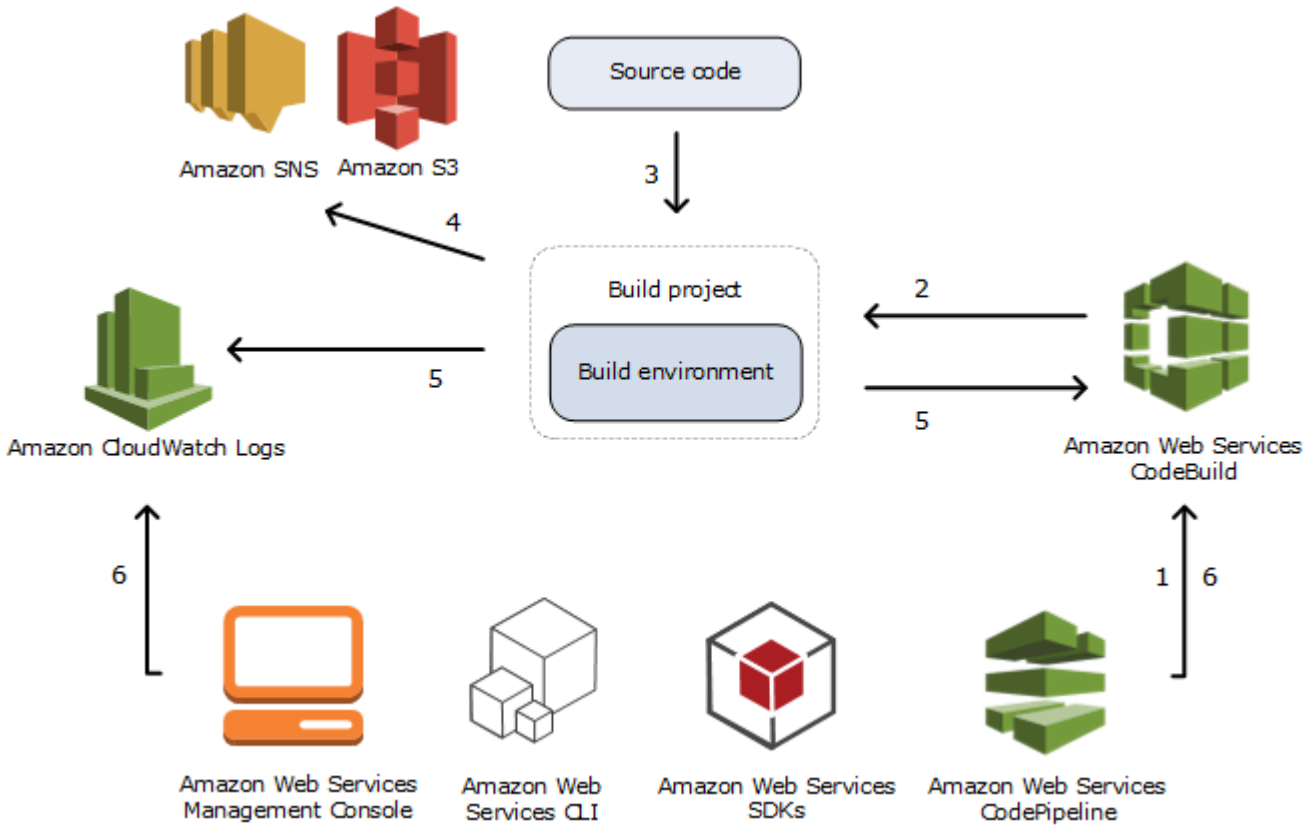
Los siguientes conceptos son importantes para entender cómo CodeBuild funciona.

Temas

- ¿Cómo CodeBuild funciona
- Siguientes pasos

¿Cómo CodeBuild funciona

En el siguiente diagrama se muestra lo que ocurre cuando se ejecuta una compilación con CodeBuild:



1. Como entrada, debes proporcionar 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 del sistema operativo, el tiempo de ejecución del lenguaje de programación y las herramientas que se CodeBuild utilizan para ejecutar una compilación. Para obtener más información, consulte lo siguiente:

   - Creación de un proyecto de compilación
   - Referencia de entornos de compilación
2. CodeBuild usa el proyecto de compilación para crear el entorno de compilación.
3. CodeBuild descarga el código fuente en el entorno de compilación y, a continuación, usa la especificación de compilación (buildspec), tal como se define en el proyecto de compilación o se incluye directamente en el código fuente. Una especificación de compilación es un conjunto de comandos de compilación y configuraciones relacionadas, en formato YAML, que se utiliza para ejecutar una compilación. CodeBuild Para obtener más información, consulte la 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, consulta Ejemplo de notificaciones de compilación.
5. Mientras se ejecuta la compilación, el entorno de compilación envía información a CodeBuild Amazon CloudWatch Logs.
6. Mientras se ejecuta la compilación, puede usar la AWS CodeBuild consola o los AWS SDK para obtener información resumida de la compilación CodeBuild e información detallada de la compilación de Amazon CloudWatch Logs. AWS CLI Si solías AWS CodePipeline ejecutar compilaciones, puedes obtener información limitada sobre las compilaciones CodePipeline.

Siguientes pasos

Ahora que tienes más información al respecto AWS CodeBuild, te recomendamos los siguientes pasos:

1. Experimente con CodeBuild un escenario de ejemplo siguiendo las instrucciones dePrimeros pasos con la consola.

2. CodeBuild Utilízalo en tus propios escenarios siguiendo las instrucciones dePlanificación de una compilación.

Cómo empezar con CodeBuild

En los siguientes tutoriales, se utiliza AWS CodeBuild para crear una colección de archivos de entrada de código fuente de ejemplo en una versión desplegable del código fuente.

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

> **Important**
>
> No te recomendamos que utilices tu cuenta AWS root para completar este tutorial.

Temas

- Introducción al AWS CodeBuild usando la consola
- Introducción al AWS CodeBuild usando el AWS CLI

Introducción al AWS CodeBuild usando la consola

En este tutorial, se utiliza AWS CodeBuild para crear una colección de archivos de entrada de código fuente de ejemplo (crear artefactos de entrada o generar entradas) en una versión desplegable del código fuente (crear artefactos de salida o compilar resultados). Concretamente, enseñas CodeBuild a usar Apache Maven, una herramienta de compilación común, para crear un conjunto de archivos de clases Java en un archivo Java Archive (JAR). No necesita estar familiarizado con Apache Maven o Java para completar este tutorial.

Puedes trabajar con ellos CodeBuild a través de la CodeBuild consola AWS CodePipeline AWS CLI, los SDK o los AWS SDK. En este tutorial se muestra cómo usar la CodeBuild consola. Para obtener más información sobre del uso de CodePipeline, consulte Úselo CodeBuild con CodePipeline.

> **Important**
>
> Los pasos de este tutorial requieren que crees recursos (por ejemplo, un bucket de S3) que puedan generar cargos en tu AWS cuenta. Estos incluyen los posibles cargos por CodeBuild y para AWS los recursos y las acciones relacionados con Amazon S3 y CloudWatch los registros. AWS KMS Para obtener más información, consulte AWS CodeBuild los precios ,

los precios de Amazon S3 y AWS Key Management Service los CloudWatch precios de Amazon.

Temas

- Paso 1: Crear el código fuente
- Paso 2: creación del archivo de especificación de compilación
- Paso 3: creación de 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 al AWS CodeBuild usando la consola)

En este paso, crea el código fuente que desea CodeBuild incorporar al bucket 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
```

Paso 2: creación del 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 una colección de comandos de compilación y ajustes relacionados, en formato YAML, que se CodeBuild usa para ejecutar una compilación. Sin una especificación de compilación, CodeBuild no puedes convertir correctamente la entrada de compilación en una salida de compilación ni localizar el artefacto de salida de compilación en el entorno de compilación para subirlo a tu bucket 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 creación, CodeBuild ejecuta cada comando especificado, de uno en uno, en el orden indicado, de principio a fin.

- `artifacts`representa el conjunto de artefactos de salida de compilación que se CodeBuild cargan en el bucket de salida. `files`representa los archivos que se van a incluir en la salida 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 la 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
```

Paso 3: creación de dos buckets de S3

(Paso anterior: Paso 2: creación del 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`* está la AWS región del depósito y *`account-ID`* el identificador de tu AWS cuenta.
- 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 depósitos deben estar en la misma AWS región que tus compilaciones. Por ejemplo, si CodeBuild ordenas ejecutar una compilación en la región Este de EE. UU. (Ohio), estos cubos también deben estar en la región Este de EE. UU. (Ohio).

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

> **Note**
>
> Aunque CodeBuild también admite la entrada de compilación almacenada en los repositorios de Bitbucket CodeCommit GitHub, y en los repositorios de Bitbucket, en este tutorial no se muestra cómo utilizarlos. Para obtener más información, consulte Planificación de una compilación.

Paso 4: cargar el código fuente y el archivo buildspec

(Paso anterior: Paso 3: creación de 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**
>
> CodeCommitEn GitHub el caso de los repositorios de Bitbucket, por convención, debes almacenar un archivo de especificaciones de compilación con un nombre `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.

Paso 5: crear el proyecto de compilación

(Paso anterior: Paso 4: cargar el código fuente y el archivo buildspec)

En este paso, creas un proyecto de compilación que se AWS CodeBuild utiliza 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 del sistema operativo, el tiempo de ejecución del lenguaje de programación y las 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, debes CodeBuild usar una imagen de Docker que contenga una versión del Java Development Kit (JDK) y Apache Maven.

Para crear el proyecto de compilación

1. Inicia sesión en Consola de administración de AWS y abre la consola en. AWS CodeBuild https://console.aws.amazon.com/codesuite/codebuild/home
2. Usa el selector de AWS regiones para elegir una AWS región en la CodeBuild que sea compatible. Para obtener más información, consulte Puntos de conexión y cuotas de AWS CodeBuild en la Referencia general de Amazon Web Services.
3. Si aparece una página de CodeBuild información, selecciona 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 construcción deben ser únicos en cada AWS cuenta. 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 esto se deba a que iniciaste sesión Consola de administración de AWS como un usuario que no

tiene permisos para crear un proyecto de compilación. Para solucionarlo, cierra sesión en y vuelve a iniciarla con las credenciales de una de las siguientes entidades de IAM: Consola de administración de AWS

- Un usuario administrador en tu AWS cuenta. Para obtener más información, consulte Crear el primer usuario y grupo Cuenta de AWS raíz en la Guía del usuario.
- Un usuario de su AWS cuenta con las `AWSCodeBuildAdminAccess` políticas `IAMFullAccess` administradas y las políticas administradas asociadas a ese usuario o a un grupo de IAM al que pertenezca el usuario. `AmazonS3ReadOnlyAccess` Si no tienes ningún usuario o grupo en tu AWS cuenta con estos permisos y no puedes añadirlos a tu usuario o grupo, ponte en contacto con el administrador de tu AWS cuenta 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. Para Bucket, elige 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 Sistema operativo, elija Amazon Linux.
10. En Runtime(s) (Tiempo de ejecución), elija Standard (Estándar).
11. En Image, elige /amazonlinux-x86_64-standard:corretto11. aws/codebuild
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. Para el nombre del bucket, elige codebuild- - -output-bucket. ***region-ID account-ID***
16. Deje Name (Nombre) y Path (Ruta) en blanco.

17. Elija Crear el proyecto de compilación.

Paso 6: ejecutar la compilación

(Paso anterior: Paso 5: crear el proyecto de compilación)

En este paso, AWS CodeBuild indicas que ejecutes la compilación con la configuración del proyecto de compilación.

Para ejecutar la compilación

1. Abre la AWS CodeBuild consola 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, elija codebuild-demo-project y, a continuación, elija Comenzar compilación. La compilación comienza inmediatamente.

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 no aparece la *<build-ID>* página codebuild-demo-project:, en la barra de navegación, selecciona Historial de compilaciones. A continuación, en la lista de proyectos de compilación, en Proyecto, elija el enlace Ejecución de compilación correspondiente a 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.

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 registros de compilación. CloudWatch

> **Note**
>
> Para proteger la información confidencial, en los CodeBuild registros se oculta 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.

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 todo el registro de compilación en CloudWatch los registros, elija el enlace Ver todo el registro.
2. En el flujo de CloudWatch registros, puedes explorar 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 la sintaxis de filtros y patrones en la Guía del CloudWatch usuario de Amazon.

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 bucket de salida.

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

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

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

Paso 10: eliminar los buckets de S3

(Paso anterior: Paso 9: obtener el artefacto de salida de la compilación)

Para evitar que se sigan cobrando cargos en tu AWS cuenta, puedes eliminar los cubos de entrada y salida 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. Añade la siguiente declaración 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.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:DeleteBucket",
```

```
        "s3:DeleteObject"
      ],
      "Resource": "*"
    }
  ]
}
```

Encapsulación

En este tutorial, AWS CodeBuild solía crear un conjunto de archivos de clases de Java en un archivo JAR. A continuación, ha consultado los resultados de la compilación.

Ahora puede intentar usarlos CodeBuild en sus 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 Utilice muestras basadas en casos para CodeBuild.

Introducción al AWS CodeBuild usando el AWS CLI

En este tutorial, se utiliza AWS CodeBuild para crear una colección de archivos de entrada de código fuente de ejemplo (denominados artefactos de entrada de compilación o entrada de compilación) en una versión desplegable del código fuente (denominada artefacto de salida de compilación o salida de compilación). Concretamente, enseñas CodeBuild a usar Apache Maven, una herramienta de compilación común, para crear un conjunto de archivos de clases de Java en un archivo Java Archive (JAR). No necesita estar familiarizado con Apache Maven o Java para completar este tutorial.

Puedes trabajar con ellos CodeBuild a través de la CodeBuild consola AWS CodePipeline AWS CLI, los SDK o los AWS SDK. En este tutorial se muestra cómo usarlo CodeBuild con. AWS CLI Para obtener información sobre el uso CodePipeline, consulteÚselo CodeBuild con CodePipeline.

> **Important**
>
> Los pasos de este tutorial requieren que cree recursos (por ejemplo, un bucket de S3) que puedan generar cargos en su AWS cuenta. Estos incluyen los posibles cargos por CodeBuild y para AWS los recursos y las acciones relacionados con Amazon S3 y CloudWatch los registros. AWS KMS Para obtener más información, consulte CodeBuild los precios , los

precios de Amazon S3 y AWS Key Management Service los CloudWatch precios de Amazon.

Temas

- Paso 1: Crear el código fuente
- Paso 2: creación del archivo de especificación de compilación
- Paso 3: creación de 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 al AWS CodeBuild usando el AWS CLI)

En este paso, crea el código fuente que desea CodeBuild incorporar al bucket 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
```

Paso 2: creación del 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 una colección de comandos de compilación y ajustes relacionados, en formato YAML, que se CodeBuild usa para ejecutar una compilación. Sin una especificación de compilación, CodeBuild no puedes convertir correctamente la entrada de compilación en una salida de compilación ni localizar el artefacto de salida de compilación en el entorno de compilación para subirlo a tu bucket 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 creación, CodeBuild ejecuta cada comando especificado, de uno en uno, en el orden indicado, de principio a fin.

- `artifacts`representa el conjunto de artefactos de salida de compilación que se CodeBuild cargan en el bucket de salida. `files`representa los archivos que se van a incluir en la salida 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 la 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
```

Paso 3: creación de dos buckets de S3

(Paso anterior: Paso 2: creación del 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`* está la AWS región del depósito y *`account-ID`* el identificador de tu AWS cuenta.
- 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 depósitos deben estar en la misma AWS región que tus compilaciones. Por ejemplo, si CodeBuild ordenas ejecutar una compilación en la región Este de EE. UU. (Ohio), estos cubos también deben estar en la región Este de EE. UU. (Ohio).

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

> **Note**
>
> Aunque CodeBuild también admite la entrada de compilación almacenada en los repositorios de Bitbucket CodeCommit GitHub, y en los repositorios de Bitbucket, en este tutorial no se muestra cómo utilizarlos. Para obtener más información, consulte Planificación de una compilación.

Paso 4: cargar el código fuente y el archivo buildspec

(Paso anterior: Paso 3: creación de 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**
>
> CodeCommitEn GitHub el caso de los repositorios de Bitbucket, por convención, debes almacenar un archivo de especificaciones de compilación con un nombre `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.

Paso 5: crear el proyecto de compilación

(Paso anterior: Paso 4: cargar el código fuente y el archivo buildspec)

En este paso, creas un proyecto de compilación que se AWS CodeBuild utiliza 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 del sistema operativo, el tiempo de ejecución del lenguaje de programación y las 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, debes CodeBuild usar una imagen de Docker que contenga una versión del Java Development Kit (JDK) y Apache Maven.

Para crear el proyecto de compilación

1. Usa el para ejecutar el AWS CLI comando: create-project

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

JSON-formatted los datos aparecen en la salida. Copie los datos a un archivo denominado `create-project.json` en una ubicación del equipo local o instancia en la que AWS CLI esté instalado. 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"
}
```

*serviceIAMRole*Sustitúyalo por el nombre de recurso de Amazon (ARN) de un rol de CodeBuild servicio (por ejemplo,`arn:aws:iam::`*account-ID*`:role/`*role-name*). Para crear uno, consulte CodeBuild Permitir interactuar con otros AWS servicios.

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`).
- For `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 CodeBuild repositorio de imágenes de Docker). `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.

- Forenvironment, computeType es un valor obligatorio que representa los recursos informáticos que se CodeBuild utilizan (en este ejemplo,BUILD_GENERAL1_SMALL).

> **Note**
>
> Otros valores disponibles en los JSON-formatted datos originales, comodescription,buildspec, auth (incluyendo type yresource),path,namespaceType, name (paraartifacts)packaging, environmentVariables (incluyendo name yvalue), timeoutInMinutesencryptionKey, y tags (incluyendo key yvalue) son opcionales. No se utilizan en este tutorial, por lo que no se muestran aquí. Para obtener más información, consulte Cree 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 transcurridos los cuales CodeBuild se detiene la compilación si ésta 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 todas las variables de entorno que se declararon y están disponibles CodeBuild para su uso durante la compilación.
  - `encryptionKey`representa el ARN de la clave gestionada por el cliente que CodeBuild se utilizó para cifrar el artefacto resultante de la compilación.
  - `arn` representa el ARN del proyecto de compilación.

> **Note**
>
> Después de ejecutar el create-project comando, es posible que aparezca un mensaje de error similar al siguiente: Usuario: no ***user-ARN*** está autorizado a ejecutar: codebuild:. CreateProject Lo más probable es que esto se deba a que lo configuraste 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 tu AWS cuenta. Para obtener más información, consulte Crear el primer usuario y grupo Cuenta de AWS raíz en la Guía del usuario.
- Un usuario de su AWS cuenta con las `AWSCodeBuildAdminAccess` políticas `IAMFullAccess` administradas y las políticas administradas asociadas a ese usuario o a un grupo de IAM al que pertenezca el usuario. `AmazonS3ReadOnlyAccess` Si no tienes ningún usuario o grupo en tu AWS cuenta con estos permisos y no puedes añadirlos a tu usuario o grupo, ponte en contacto con el administrador de tu AWS cuenta para obtener ayuda. Para obtener más información, consulte AWS políticas gestionadas (predefinidas) para AWS CodeBuild.

Paso 6: ejecutar la compilación

(Paso anterior: Paso 5: crear el proyecto de compilación)

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

Para ejecutar la compilación

1. Usa el AWS CLI para ejecutar el start-build comando:

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

   *project-name*Sustitúyelo por el nombre del 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.

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

- Usa el AWS CLI para ejecutar el batch-get-builds comando.

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

*id*Sustitúyalo por el `id` valor que apareció en la salida 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 compilación que CodeBuild ejecuta durante el proceso de compilación. 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 el MD5 y los SHA-256 hashes 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 Properties, expanda Metadata y consulte los valores de x-amz-meta-codebuild-content-md5 y x-amz-meta-codebuild-content-sha256. (En la consola de Amazon S3, el valor ETag del artefacto de salida de la compilación no debe interpretarse como el MD5 o el hash). SHA-256
  >
  > Si usa los AWS SDK para obtener estos hashes, los valores se denominan y. `codebuild-content-md5` `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 CodeBuild encabezado denominado `x-amz-meta-codebuild-buildarn` que contiene el encabezado `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.

Paso 8: ver información detallada sobre la compilación

(Paso anterior: Paso 7: ver información resumida sobre la compilación)

En este paso, puede ver información detallada sobre los CloudWatch registros de compilación.

> **Note**
>
> Para proteger la información confidencial, en los CodeBuild registros se oculta 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.

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 la secuencia 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 la 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, CodeBuild completeté correctamente las fases previas, posteriores y posteriores a la compilación. Ha ejecutado las pruebas unitarias y ha compilado correctamente el archivo `messageUtil-1.0.jar`.

Paso 9: obtener el artefacto de salida de la compilación

(Paso anterior: Paso 8: ver información detallada sobre la compilación)

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

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

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

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

   > **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`.

Paso 10: eliminar los buckets de S3

(Paso anterior: Paso 9: obtener el artefacto de salida de la compilación)

Para evitar que se sigan cobrando cargos en tu AWS cuenta, puedes eliminar los cubos de entrada y salida 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. Añade la siguiente declaración 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.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:DeleteBucket",
        "s3:DeleteObject"
      ],
      "Resource": "*"
    }
  ]
}
```

Encapsulación

En este tutorial, AWS CodeBuild solía crear un conjunto de archivos de clases de Java en un archivo JAR. A continuación, ha consultado los resultados de la compilación.

Ahora puede intentar usarlos CodeBuild en sus 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 Utilice muestras basadas en casos para CodeBuild.

Utilice muestras basadas en casos para CodeBuild

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

Cross-service muestras

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

Ejemplo de insignias de compilación

Muestra cómo configurar CodeBuild con insignias de construcción.

Ejemplo para probar un informe

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 con varias entradas y salidas

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

Ejemplos de ejecución de pruebas paralelas

Muestra cómo utilizar el comando `codebuild-tests-run` de la CLI para dividir y ejecutar pruebas en entornos de ejecución en paralelo.

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 CodeBuild compilación.

Third-party muestras de repositorios de origen para CodeBuild

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

Establecimiento de nombres de artefacto en el momento de la compilación con control de versiones semánticas

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

Cross-service muestras para CodeBuild

Puede usar estos ejemplos multiservicio 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. Las instrucciones de ejemplo le mostrarán cómo crear y enviar una imagen de Docker a Amazon ECR, crear un proyecto de Go, construir el proyecto, ejecutar el proyecto y configurar los permisos para poder conectarse CodeBuild a Amazon ECR.

Ejemplo de Amazon EFS

Muestra cómo configurar un archivo de especificaciones de construcción para que un CodeBuild proyecto se monte y se compile en un sistema de archivos Amazon EFS. Las instrucciones de ejemplo le mostrarán cómo crear una instancia de Amazon VPC, crear un sistema de archivos en la instancia de Amazon VPC, crear y compilar un proyecto que use la instancia de Amazon VPC y, después, revisar el archivo y las variables del proyecto generado.

AWS CodePipeline muestras

Muestra cómo usarlo AWS CodePipeline para crear una compilación con compilaciones por lotes, así como con varias fuentes de entrada y varios artefactos de salida. En esta sección se incluyen archivos JSON de ejemplo que muestran estructuras de canalización que crean compilaciones por lotes con artefactos individuales y artefactos combinados. Se proporciona un ejemplo adicional de JSON que muestra la estructura de canalización con varias fuentes de entrada y varios artefactos de salida.

AWS Config muestra

Muestra cómo realizar la configuración AWS Config. Muestra CodeBuild los recursos a los que se hace un seguimiento y describe cómo buscar CodeBuild proyectos en ellos AWS Config. Las instrucciones de ejemplo muestran los requisitos previos para la integración AWS Config, los pasos para configurarlos AWS Config y los pasos para buscar CodeBuild proyectos y datos en AWS Config ellos.

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 Las instrucciones de ejemplo muestran cómo configurar los permisos para CodeBuild poder comunicarse con Amazon SNS y CloudWatch cómo crear e identificar CodeBuild temas en Amazon SNS, cómo suscribir a los destinatarios al tema y cómo configurar las reglas en. CloudWatch

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 posibles cargos por AWS CodeBuild y por AWS los recursos y 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 de Amazon ECR

Ejecución del ejemplo de Amazon ECR

Utilice las siguientes instrucciones para ejecutar el ejemplo de Amazon ECR para CodeBuild.

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 publicación de una imagen de Docker en Amazon ECR de Ejemplo de publicación de una imagen de Docker en 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 GitHub repositorio 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)*.

   b. Cree un proyecto de compilación, ejecute la compilación y vea la información de compilación relacionada.

   Si utilizas el AWS CLI para crear el proyecto de compilación, la JSON-formatted entrada del `create-project` comando 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 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 de AWS 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:GetDownloadUrlForLayer, 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.

  En el siguiente ejemplo de política se utilizan tanto CodeBuild las credenciales como una imagen de Amazon ECR multicuenta.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "CodeBuildAccessPrincipal",
            "Effect": "Allow",
            "Action": [
                "ecr:GetDownloadUrlForLayer",
                "ecr:BatchGetImage",
                "ecr:BatchCheckLayerAvailability"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
```

```
                    "aws:SourceArn": "arn:aws:codebuild:us-
east-1:111122223333:project/MyProject",
                    "aws:SourceAccount": "111122223333"
                }
            }
        },
        {
            "Sid": "CodeBuildAccessCrossAccount",
            "Effect": "Allow",
            "Action": [
                "ecr:GetDownloadUrlForLayer",
                "ecr:BatchGetImage",
                "ecr:BatchCheckLayerAvailability"
            ],
            "Resource": "*"
        }
    ]
}
```

- 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

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "CodeBuildAccessPrincipal",
            "Effect": "Allow",
            "Resource": [
                "arn:aws:codecommit:us-
east-2:111122223333:MySharedDemoRepo"
            ],
            "Action": [
                "ecr:GetDownloadUrlForLayer",
                "ecr:BatchGetImage",
                "ecr:BatchCheckLayerAvailability"
            ]
        },
        {
            "Sid": "CodeBuildAccessCrossAccount",
```

```
            "Effect": "Allow",
            "Resource": [
                "arn:aws:codecommit:us-
east-2:111122223333:MySharedDemoRepo"
            ],
            "Action": [
                "ecr:GetDownloadUrlForLayer",
                "ecr:BatchGetImage",
                "ecr:BatchCheckLayerAvailability"
            ]
        }
    ]
}
```

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

   Si usa la AWS CLI para crear el proyecto de compilación, la JSON-formatted entrada del `create-project` comando 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)
}
```

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 montar y, a continuación, compilar una aplicación Java en un sistema de archivos de Amazon EFS. Antes de empezar, debe tener una aplicación Java lista para su creación y que esté cargada en un bucket 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.

Temas

- Uso AWS CodeBuild con Amazon Elastic File System
- Solución de problemas de integración de Amazon EFS

Uso AWS CodeBuild con Amazon Elastic File System

El ejemplo abarca los cuatro pasos de alto nivel necesarios para usar Amazon EFS con AWS CodeBuild. Son los siguientes:

1. Cree una nube privada virtual (VPC) en su AWS cuenta.
2. Crear un sistema de archivos que utilice esta VPC.
3. Crea y construye un CodeBuild proyecto que utilice la VPC. El CodeBuild proyecto usa 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.
4. Revise el proyecto de compilación para asegurarse de que se han generado las variables y los archivos del proyecto correctos.

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

Temas

- Paso 1: Cree una VPC usando CloudFormation
- Paso 2: creación de un sistema de archivos de Amazon Elastic File System con su VPC
- Paso 3: Cree un CodeBuild proyecto para usarlo con Amazon EFS
- Paso 4: revisión del proyecto de compilación

Paso 1: Cree una VPC usando CloudFormation

Crea tu VPC con una CloudFormation plantilla.

1. Siga las instrucciones de uso CloudFormation Plantilla de VPC para CloudFormation crear una VPC.

> **Note**
>
> La VPC creada con esta CloudFormation plantilla tiene dos subredes privadas y dos públicas. Solo debe utilizar subredes privadas cuando use AWS CodeBuild para montar el sistema de archivos que ha creado en Amazon EFS. Si utiliza alguna de las subredes públicas, la compilación genera un error.

2. Inicie sesión en Consola de administración de AWS y abra la consola de Amazon VPC en. https://console.aws.amazon.com/vpc/
3. Elige la VPC con la que creaste. CloudFormation
4. En la pestaña Description (Descripción), anote el nombre de la VPC y su ID. Ambos serán necesarios cuando cree su proyecto de AWS CodeBuild más adelante en este ejemplo.

Paso 2: creación de 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 Consola de administración de AWS y abra la consola de Amazon EFS en 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.

Paso 3: Cree 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. Abre 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 de un 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, selecciona aws/codebuild /amazonlinux-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, ingresa el nombre del rol que se crea para ti. CodeBuild
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 sin VPC. Si quiere usar contenedores de Docker para las compilaciones de VPC, consulte Runtime Privilege and Linux Capabilities en el sitio web de Docker Docs y habilite 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. Debe tener menos de 129 caracteres y contener sólo caracteres alfanuméricos y guiones bajos. CodeBuild utiliza este identificador para crear una variable de entorno que identifica 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 Directory path (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
      rsize=1048576
      wsize=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 buildspec 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 -Dgpg.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.

Paso 4: revisión del proyecto de compilación

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.

Solución de problemas de integración de Amazon EFS

A continuación se enumeran los 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 inesperado de EC2: 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 la 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 inesperado de EC2: UnauthorizedOperation

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

AWS CodePipeline muestras para CodeBuild

En esta sección se describen ejemplos de integraciones entre CodePipeline y CodeBuild.

Muestra	Description (Descripción)
Ejemplos de CodePipeline/CodeBuild integraciones y compilaciones por lotes	Estos ejemplos muestran cómo usarlos AWS CodePipeline para crear un proyecto de compilación que utilice compilaciones por lotes.
Ejemplo de una CodePipeline/CodeBuild integración con múltiples fuentes de entrada y artefactos de salida	En este ejemplo, se muestra cómo usarlo AWS CodePipeline para crear un proyecto de compilación que utilice varias fuentes de entrada para crear varios artefactos de salida.

Ejemplos de CodePipeline/CodeBuild integraciones y compilaciones por lotes

AWS CodeBuild admite compilaciones por lotes. Los siguientes ejemplos muestran cómo usarlo AWS CodePipeline para crear un proyecto de compilación que utilice compilaciones por lotes.

Puedes usar un JSON-formatted archivo que defina la estructura de tu canalización y, a continuación, usarlo con el 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 la creación por lotes CodePipeline, defina el `BatchEnabled` parámetro del `configuration` objeto en`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
```

```
  }
}
```

A continuación se muestra un ejemplo de un archivo de CodeBuild especificaciones de construcción 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 `build1` la ubicación de. `build1` Para el nombre de salida`build1_artifact1`, CodeBuild cargará el artefacto secundario `artifact1` de `build1` en la ubicación de`build1_artifact1`,

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

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 .

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 la compilación 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
 }
}
```

A continuación se muestra un ejemplo de un archivo de CodeBuild especificaciones de construcción 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 la combinación de artefactos está habilitada para la creación 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 .

Ejemplo de una CodePipeline/CodeBuild integración con múltiples fuentes de entrada y artefactos de salida

Un AWS CodeBuild proyecto puede requerir más de una fuente de entrada. También puede crear más de un artefacto de salida. En este ejemplo se muestra cómo usarlo AWS CodePipeline para 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 JSON-formatted archivo que defina la estructura de tu canalización y, a continuación, usarlo con el 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, consulte la referencia a la estructura de la 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 donde CodeBuild busca y ejecuta tu 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 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 Config muestra con 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 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 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.

Temas

- Úselo CodeBuild con AWS Config
- Paso 3: Ver AWS CodeBuild los datos en la AWS Config consola

Úselo CodeBuild con AWS Config

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

Temas

- Requisitos previos
- Paso 1: Configurar AWS Config
- Paso 2: Busca AWS CodeBuild proyectos

Requisitos previos

Cree su AWS CodeBuild proyecto. Para obtener instrucciones, consulte Creación de un proyecto de compilación.

Paso 1: Configurar AWS Config

- Configuración de AWS Config (consola)
- Configuración de AWS Config (AWS CLI)

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

Paso 2: 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 CodeBuilddel 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.

Paso 3: Ver AWS CodeBuild los datos en la AWS Config consola

Al buscar 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.

Ejemplo de creación de notificaciones para CodeBuild

Amazon CloudWatch Events tiene soporte integrado para AWS CodeBuild. CloudWatch Los eventos son un flujo de eventos del sistema que describen los cambios en sus AWS recursos.

Con CloudWatch los eventos, se escriben reglas declarativas para asociar los eventos de interés con las acciones automatizadas que se deben tomar. En este ejemplo, se utilizan Amazon CloudWatch Events y Amazon Simple Notification Service (Amazon SNS) para enviar notificaciones de compilación a los suscriptores cada vez que las compilaciones se realizan correctamente, fallan, pasan de una fase de creación a otra o cualquier combinación de estos eventos.

> **Important**
>
> La ejecución de este ejemplo podría generar cargos en tu AWS cuenta. Estos incluyen los posibles cargos por CodeBuild y para AWS los recursos y las acciones relacionados con Amazon CloudWatch y Amazon SNS. Para obtener más información, consulte CodeBuild los precios, los precios de Amazon y CloudWatch los precios de Amazon SNS.

Temas

- Ejecución del ejemplo de notificaciones de compilación
- Referencia del formato de entrada de las notificaciones de compilación

Ejecución del ejemplo de notificaciones de compilación

Utilice el siguiente procedimiento para ejecutar el ejemplo de notificaciones de compilación.

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 al grupo de IAM al que esté asociado el usuario). No se recomienda usar una cuenta AWS raíz. 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 ayudarle a encontrar el lugar donde debe añadir la instrucción. No elimine ninguna instrucción ni incluya estos puntos suspensivos en la política existente.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sns:CreateTopic",
                "sns:GetTopicAttributes",
                "sns:List*",
                "sns:Publish",
                "sns:SetTopicAttributes",
                "sns:Subscribe"
            ],
            "Resource": "*"
        }
    ]
}
```

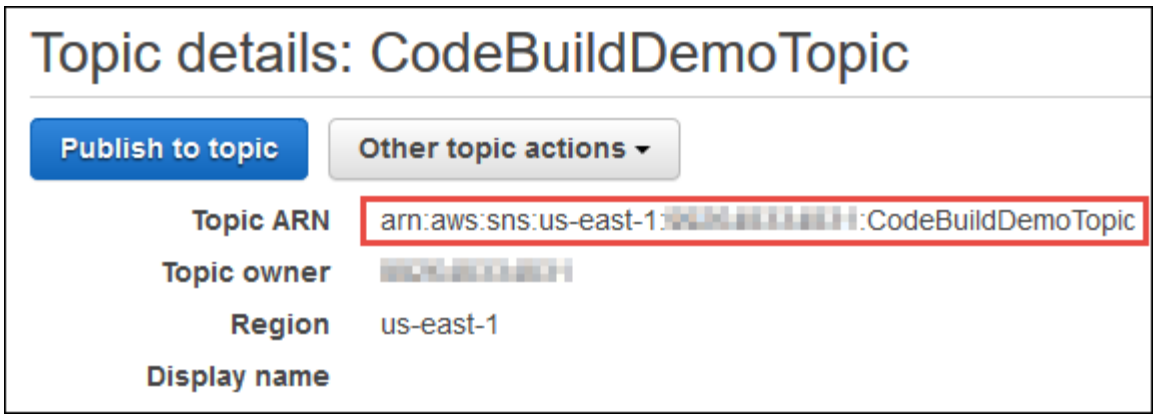
> **Note**
>
> La entidad de IAM que modifica esta política debe tener permiso en IAM para modificar políticas.
> Para obtener más información, consulte Edición de políticas administradas por el cliente o la sección "Para editar o eliminar una política insertada de un grupo, un usuario o un rol" de Uso de políticas insertadas (consola) en la Guía del usuario de IAM.

2. Cree o identifique un tema en Amazon SNS. AWS CodeBuild usa CloudWatch los eventos para enviar notificaciones de creación 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.
   2. Seleccione Crear 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. Seleccione Crear tema.
5. En la CodeBuildDemoTopic página Detalles del tema:, copie el valor ARN del tema. Necesitará este valor para el siguiente paso.

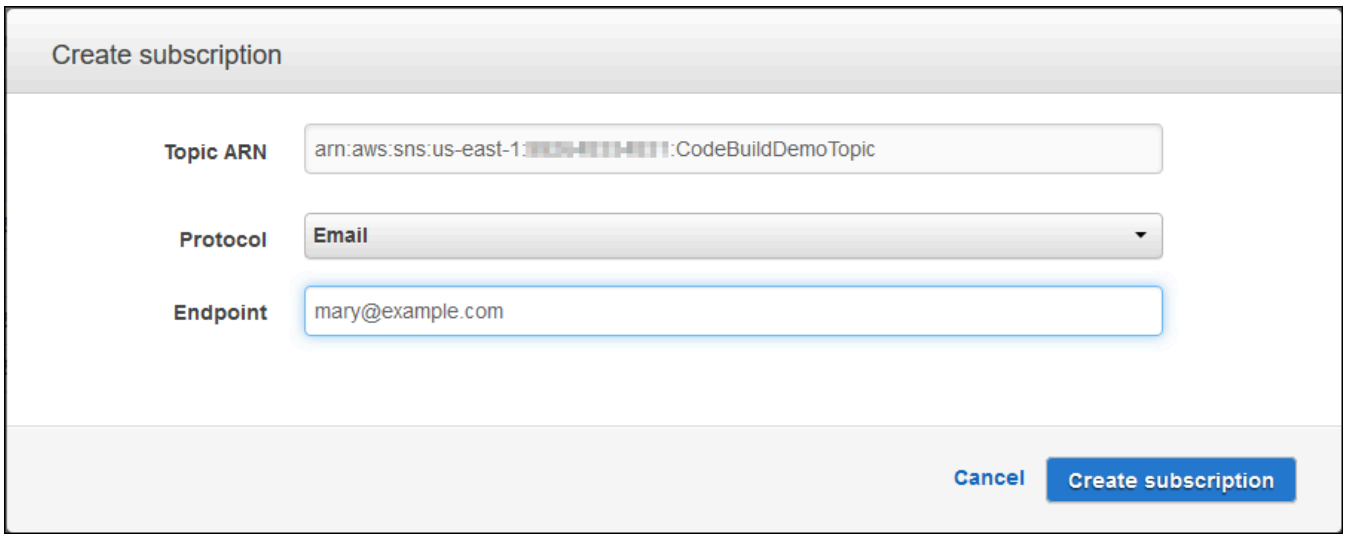


Para obtener más información, consulte Crear un tema en la Guía para desarrolladores de Amazon SNS.

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, elige Correo electrónico.
   4. En Punto de conexión, escriba la dirección de correo electrónico completa del destinatario.



5. Elija Create Subscription.
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.

4. Si utiliza un usuario en lugar de una cuenta AWS raíz o un usuario administrador para trabajar con CloudWatch los eventos, añada la siguiente declaración (entre *### BEGIN ADDING STATEMENT HERE ###* y*### END ADDING STATEMENT HERE ###*) al usuario (o al grupo de IAM al que esté asociado el usuario). No se recomienda usar una cuenta AWS raíz. Esta sentencia se usa 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.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
```

```
            "Effect": "Allow",
            "Action": [
                "events:*",
                "iam:PassRole"
            ],
            "Resource": "arn:aws:iam::*:role/Service*"
        }
    ]
}
```

> **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.

5. Crea una regla en CloudWatch Eventos. Para ello, abre 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 Nombre de servicio, elija 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, consulte los patrones de eventos en la Guía del EventBridge usuario de Amazon.

Para obtener más información sobre el filtrado con patrones de eventos, consulte Content-based Filtrar 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. Seleccione Creación de regla.

20. Cree proyectos de compilación, ejecute las compilaciones y consulte información sobre la compilación.
21. Confirma que las notificaciones de compilación CodeBuild se están enviando correctamente. 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, elige 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, elige Acciones y, a continuación, elige Deshabilitar.

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

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"
  }
}
```

Ejemplo de creación de insignias con CodeBuild

AWS CodeBuild ahora admite el uso de insignias de compilación, que proporcionan una imagen (insignia) que se puede incrustar y se genera 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.

Temas

- Creación de un proyecto de compilación con insignias de compilación

- Acceso AWS CodeBuild crea insignias
- Publica insignias de CodeBuild compilación
- CodeBuild estados de las insignias

Creación de un proyecto de compilación con insignias de compilación

Usa uno de los siguientes procedimientos para crear un proyecto de compilación con las insignias de compilación habilitadas. Puedes usar AWS CLI o el Consola de administración de AWS.

Para crear un proyecto de compilación con las insignias de compilación habilitadas (AWS CLI)

- Para obtener información acerca de cómo crear un proyecto de compilación, consulte Cree un proyecto de compilación (AWS CLI). Para incluir las insignias de compilación en tu AWS CodeBuild proyecto, debes especificarlas *badgeEnabled* con un valor de. `true`

Para crear un proyecto de compilación con insignias de compilación habilitadas (consola)

1. Abre la AWS CodeBuild consola en. https://console.aws.amazon.com/codesuite/codebuild/home
2. Si aparece una página de CodeBuild información, seleccione 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.
3. En Project name (Nombre de proyecto), escriba un nombre para este proyecto de compilación. Los nombres de los proyectos de construcción 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 origen de Amazon S3. Dado que AWS CodePipeline utiliza Amazon S3 para la transferencia de artefactos, las insignias de compilación no son compatibles con los proyectos de construcción que forman parte de un proceso creado en. CodePipeline

- Si eligió CodeCommit, en Repository (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, para el 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, vuelve a la AWS CodeBuild consola y, 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 el sistema operativo, los tiempos de ejecución, la imagen y la 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 eliges Amazon ECR, usa el repositorio de Amazon ECR y la imagen de Amazon ECR para elegir la imagen de Docker en tu 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 usas 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 la 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.

Acceso AWS CodeBuild crea insignias

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

- En la CodeBuild consola, en la lista de proyectos de compilación, en la columna Nombre, elige el enlace que corresponde al proyecto de compilación. En la *project-name* página Crear proyecto:, en Configuración, selecciona Copiar la URL de la 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 insignias de CodeBuild compilación

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 en el archivo readme.md de tu repositorio de origen (por ejemplo, o). GitHub CodeCommit Por ejemplo:

```
![](<build badge URL>)
```

CodeBuild estados de las insignias

La insignia de CodeBuild creación puede tener uno de los siguientes estados.

- 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.

«Informe de prueba con el AWS CLI'muestra

Las pruebas que especifique en su archivo buildspec se ejecutan durante la compilación. En este ejemplo, se muestra cómo usar el AWS CLI para incorporar 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.

Puedes 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

- Ejecución del ejemplo para probar un informe

Ejecución del ejemplo para probar un informe

Siga los pasos siguientes para ejecutar el ejemplo del informe de prueba.

Temas

- Requisitos previos
- Paso 1: creación de un grupo de informes
- Paso 2: configuración de un proyecto con un grupo de informes
- Paso 3: ejecución y visualización de 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)
  - Visual Studio TRX XML (.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.

Paso 1: creación de 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
  }
}
```

Paso 2: configuración de un proyecto con un grupo de informes

Para ejecutar un informe, primero debes crear un proyecto de CodeBuild compilación que esté configurado con tu 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 puedes 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 especificas un nombre en lugar de un ARN, CodeBuild crea un grupo de informes cuando se 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 Creación de informes 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
```

Paso 3: ejecución y visualización de 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": "<amzn-s3-demo-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": []
}
```

4. 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>"
    }
  ]
}
```

Ejemplos de Docker para CodeBuild

En esta sección se describen ejemplos de integraciones entre Docker y. AWS CodeBuild

Muestra	Description (Descripción)
Docker en una muestra de imagen personalizada para CodeBuild	En este ejemplo, se crea y ejecuta una imagen de Docker mediante CodeBuild una imagen de compilación de Docker personalizada (`docker:dind` en Docker Hub).
Ejemplo de servidor de creación de imágenes Docker para CodeBuild	En este ejemplo, se transfieren compilaciones de Docker a un servidor de compilación de imágenes administrado.
Ejemplo de compilación de Windows Docker para CodeBuild	En este ejemplo, se crea y ejecuta una imagen de Docker de Windows mediante. CodeBuild
Ejemplo de «Publicar una imagen de Docker en un repositorio de imágenes de Amazon ECR» 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).
Registro privado con AWS Secrets Manager muestra para CodeBuild	En este ejemplo, se muestra cómo utilizar una imagen de Docker almacenada en un registro privado como entorno de CodeBuild ejecución.

Docker en una muestra de imagen personalizada para CodeBuild

En el siguiente ejemplo, se crea y ejecuta una imagen de Docker mediante AWS CodeBuild una imagen de compilación de Docker personalizada (`docker:dind`en 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 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 de Docker en una imagen personalizada

Ejecución del ejemplo de Docker en una imagen personalizada

Utilice el siguiente procedimiento para ejecutar el ejemplo de Docker en una imagen personalizada. Para obtener más información sobre este ejemplo, consulte Docker en una muestra de imagen personalizada para CodeBuild.

Para ejecutar el ejemplo de Docker en una imagen personalizada

1. Cree los archivos tal y como se describe en las Archivos secciones Estructura de directorios 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)*.

2. Cree un proyecto de compilación, ejecute la compilación y vea la información de compilación relacionada.

   Si utilizas el AWS CLI para crear el proyecto de compilación, la JSON-formatted entrada del `create-project` comando 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 sin VPC. Si quiere usar contenedores de Docker para las compilaciones de VPC, consulte Runtime Privilege and Linux Capabilities en el sitio web de Docker Docs y habilite 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"
```

Ejemplo de servidor de creación de imágenes Docker para CodeBuild

En el ejemplo siguiente se transfieren compilaciones de Docker a un servidor de compilación de imágenes administrado. Puedes adaptar este ejemplo para aprovisionar un servidor de creación de imágenes de Docker dedicado y gestionado en la configuración de tu proyecto. CodeBuild Ten en cuenta que la instancia aprovisionada está activa mientras las compilaciones se ejecutan activamente para el proyecto, y la instancia se detiene cuando las compilaciones no se están ejecutando. La instancia aprovisionada se almacena durante un mes como máximo antes de reciclarla. Para obtener más información, consulte Capacidad del servidor CodeBuild Docker.

> **Important**
>
> La ejecución de este ejemplo puede generar cargos en su AWS cuenta. Estos incluyen los posibles cargos por CodeBuild y por AWS los recursos y las acciones relacionados con Amazon S3 y CloudWatch los registros. AWS KMS Para obtener más información, consulte

CodeBuild los precios , los precios de Amazon S3 y AWS Key Management Service los CloudWatch precios de Amazon.

Temas

- Configuración de un servidor de Docker

Configuración de un servidor de Docker

Utilice el siguiente procedimiento para aprovisionar un entorno informático dedicado para un CodeBuild proyecto que gestione las cargas de trabajo de Docker y almacene las capas de imágenes de Docker.

Cómo configurar un servidor de Docker

1. Crea los archivos tal y como se describe en las Archivos secciones Estructura de directorios y de este tema y, a continuación, súbelos a un depósito de entrada de S3 o a un repositorio AWS CodeCommit de GitHub 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)*.

2. Cree un proyecto de compilación, ejecute la compilación y vea la información de compilación relacionada:

   a. En la sección Entorno de la consola, elija Configuración adicional, vaya a Configuración del servidor Docker y, a continuación, seleccione Habilitar servidor de Docker para este proyecto. A continuación, puede elegir Tipo de computación del servidor de Docker y proporcione una Credencial de registro.

   b. Si utilizas el AWS CLI para crear el proyecto de compilación, la JSON-formatted entrada del `create-project` comando puede 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/DockerServerSample.zip"
  },
  "artifacts": {
    "type": "NO_ARTIFACTS"
  },
  "environment": {
    "type": "LINUX_CONTAINER",
    "image": "aws/codebuild/amazonlinux-x86_64-standard:5.0",
    "computeType": "BUILD_GENERAL1_LARGE",
    "dockerServer": [
         {
            "computeType": "BUILD_GENERAL1_LARGE",
            "securityGroupIds": [ "security-groups-ID" ]
         }
      ]
  },
  "serviceRole": "arn:aws:iam::account-ID:role/role-name"
}
```

> **Note**
>
> Los grupos de seguridad configurados para servidores Docker deben permitir la entrada de tráfico de red desde la VPC configurada en el proyecto. Deberían permitir la entrada por el puerto 9876.

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

`Dockerfile` (in *(root directory name)*)

```
FROM public.ecr.aws/amazonlinux/amazonlinux:latest

RUN echo "Hello World"
```

Ejemplo de compilación de Windows Docker para CodeBuild

En el siguiente ejemplo, se crea y ejecuta una imagen de Docker de Windows mediante. CodeBuild

Temas

- Ejecución del ejemplo de compilación de Windows Docker

Ejecución del ejemplo de compilación de Windows Docker

Utilice el siguiente procedimiento para ejecutar las compilaciones de Windows Docker.

Cómo ejecutar el ejemplo de compilaciones de Windows Docker

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

2. Cree una flota de `WINDOWS_EC2`.

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

   ```
   {
     "name": "fleet-name",
     "baseCapacity": 1,
     "environmentType": "WINDOWS_EC2",
     "computeType": "BUILD_GENERAL1_MEDIUM"
   }
   ```

3. Cree un proyecto de compilación, ejecute la compilación y vea la información de compilación relacionada.

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

   ```
   {
     "name": "project-name",
     "source": {
       "type": "S3",
       "location": "bucket-name/DockerImageSample.zip"
     },
     "artifacts": {
       "type": "NO_ARTIFACTS"
     },
     "environment": {
       "type": "WINDOWS_EC2",
       "image": "Windows",
   ```

```
    "computeType": "BUILD_GENERAL1_MEDIUM",
    "fleet": {
       "fleetArn": "fleet-arn"
    }
  },
  "serviceRole": "arn:aws:iam::account-ID:role/role-name"
}
```

4. 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 `mcr.microsoft.com/windows/servercore:ltsc2022`. 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 powershell -Command "Write-Host 'Hello World!'"
```

`Dockerfile` (in *(root directory name)*)

```
FROM mcr.microsoft.com/windows/servercore:ltsc2022
```

```
RUN powershell -Command "Write-Host 'Hello World'"
```

Ejemplo de «Publicar una imagen de Docker en un repositorio de imágenes de Amazon ECR» 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 de publicación de una imagen de Docker en Amazon ECR para insertar 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. Multi-stage Las compilaciones de imágenes de Docker ayudan a reducir el tamaño de la imagen final de Docker. Para obtener más información, consulte Use multi-stage builds with Docker.

> **Important**
>
> La ejecución de este ejemplo podría generar cargos en tu cuenta. AWS Estos incluyen los posibles cargos por AWS CodeBuild y por 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, los precios de Amazon S3 , AWS Key Management Service los precios de Amazon y CloudWatch los precios de Amazon Elastic Container Registry.

Temas

- Ejecución del ejemplo de publicación de una imagen de Docker en Amazon ECR
- Adaptación del ejemplo de publicación de una imagen de Docker en Amazon ECR para insertar en Docker Hub

Ejecución del ejemplo de publicación de una imagen de Docker en Amazon ECR

Utilice el siguiente procedimiento para ejecutar el ejemplo que publica una imagen de Docker en Amazon ECR. Para obtener más información sobre este ejemplo, consulte Ejemplo de «Publicar una imagen de Docker en un repositorio de imágenes de Amazon ECR» para CodeBuild.

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 al grupo de IAM al que esté asociado el usuario). No se recomienda usar una cuenta AWS raíz. 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 Consola de administración de AWS en la Guía del usuario.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ecr:CreateRepository"
            ],
            "Resource": "*"
        }
    ]
}
```

> **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 su entorno de compilación y en la que ejecuta su 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 instantáneo a imágenes para su función de CodeBuild servicio de IAM.
3. Agregue esta declaración (entre *`### BEGIN ADDING STATEMENT HERE ###`* y *`### END ADDING STATEMENT HERE ###`*) a la política que asignó a su rol de AWS CodeBuild servicio. Esta declaración permite subir 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.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ecr:BatchCheckLayerAvailability",
                "ecr:CompleteLayerUpload",
                "ecr:GetAuthorizationToken",
                "ecr:InitiateLayerUpload",
                "ecr:PutImage",
                "ecr:UploadLayerPart"
            ],
            "Resource": "*"
        }
    ]
```

```
}
```

> **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 Archivos secciones Estructura de directorios y secciones 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. Cree un proyecto de compilación, ejecute la compilación y vea la información de 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 Image, elige 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 *account-ID*
      - IMAGE_TAG con un valor de Latest
      - IMAGE_REPO_NAME con un valor de *Amazon-ECR-repo-name*

Si usas el AWS CLI para crear el proyecto de compilación, la JSON-formatted entrada del `create-project` comando puede 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 ha subido CodeBuild correctamente al 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 de publicación de una imagen de Docker en Amazon ECR para insertar en Docker Hub

Para adaptar el ejemplo de publicación de una imagen de Docker en Amazon ECR de modo que la imagen de Docker se inserte en Docker Hub en lugar de en Amazon ECR, edite el código del ejemplo. Para obtener más información sobre el ejemplo, consulte Ejemplo de «Publicar una imagen de Docker en un repositorio de imágenes de Amazon ECR» para CodeBuild y Ejecución del ejemplo de publicación de una imagen de Docker en Amazon ECR.

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

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

```
...
  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
...
```

Con estas Hub-specific líneas de código de Docker:

```
...
  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 GitHub repositorio 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 JSON-formatted entrada por el `create-project` comando:

```
...
    "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. Cree un entorno de compilación, ejecute la compilación y vea la información de compilación relacionada.
5. Confirma que la imagen de Docker se ha subido AWS CodeBuild correctamente al 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.

Registro privado con AWS Secrets Manager 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.

Temas

- Requisitos del ejemplo de un registro privado
- Crea un CodeBuild proyecto con un registro privado
- Configuración de una credencial de registro privada para ejecutores autoalojados

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 Key/value pares, cree un par clave-valor para su nombre de usuario de Docker Hub y un par clave-valor para su contraseña de Docker Hub.

   c. Siga los pasos de Crear un secreto. AWS Secrets Manager

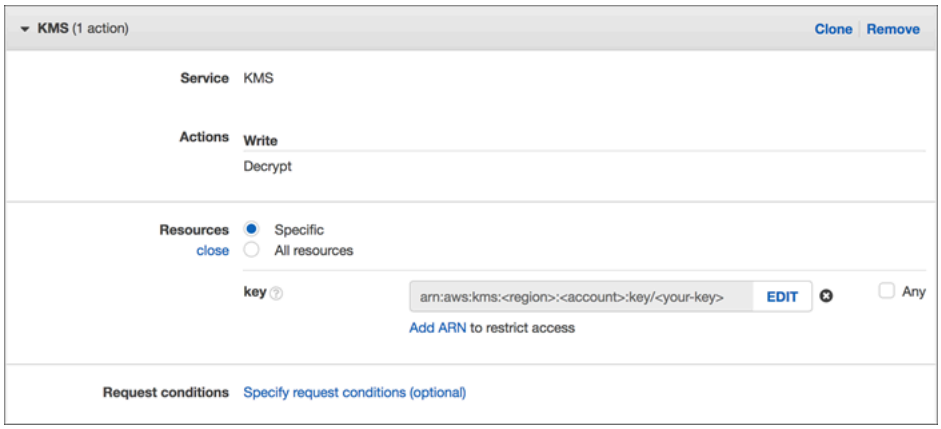
   d. En el paso 5, en la página Configurar la rotación automática, desactívala porque las claves corresponden a sus credenciales de Docker Hub.

   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`.



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 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.

Configuración de una credencial de registro privada para ejecutores autoalojados

Utilice las siguientes instrucciones para configurar una credencial de registro para un ejecutor autoalojado.

> **Note**
>
> Tenga en cuenta que esta credencial solo se utilizará si las imágenes se sustituyen por las de registros privados.

AWS Management Console

1. Abra la AWS CodeBuild consola en. https://console.aws.amazon.com/codesuite/codebuild/home
2. Cree un proyecto de compilación o seleccione uno existente. Para obtener más información, consulte Creación de un proyecto de compilación (consola) y Cambiar la configuración de un proyecto de compilación (consola).
3. En Entorno, elija Configuración adicional.
4. En Configuración adicional, introduzca el nombre o el ARN del secreto de la credencial AWS Secrets Manager de registro (opcional).



AWS CLI

1. Si desea crear un proyecto nuevo, ejecute el comando create-project.

```
aws codebuild create-project \
    --name project-name \
    --source type=source-type,location=source-location \
    --environment "type=environment-type,image=image,computeType=compute-
type,registryCredential={credentialProvider=SECRETS_MANAGER,credential=secret-
name-or-arn},imagePullCredentialsType=CODEBUILD|SERVICE_ROLE" \
    --artifacts type=artifacts-type \
    --service-role arn:aws:iam::account-ID:role/service-role/service-role-name
```

2. Si desea actualizar un proyecto existente, ejecuta el comando update-project.

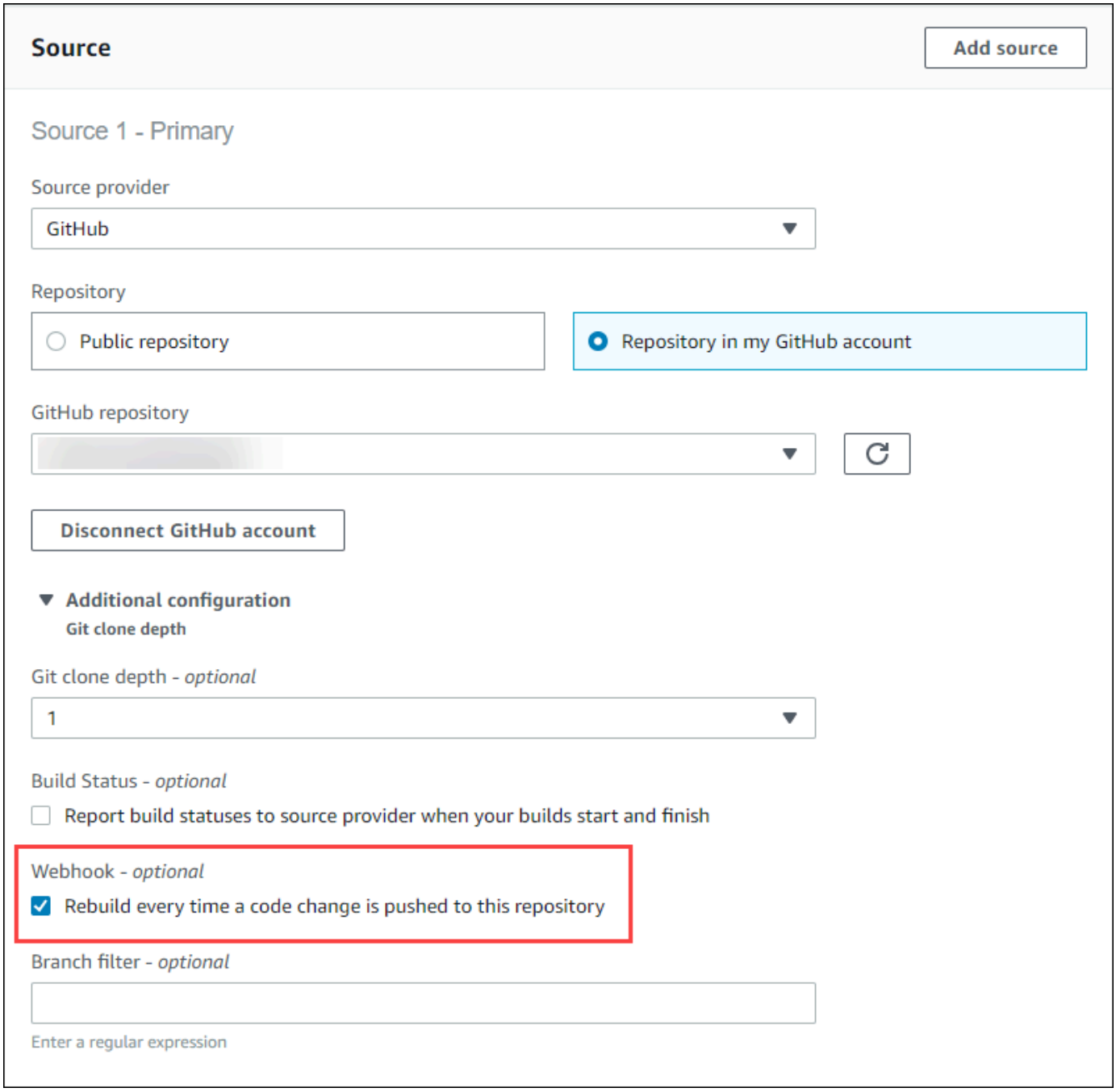
```
aws codebuild update-project \
    --name project-name \
    --environment "type=environment-type,image=image,computeType=compute-
type,registryCredential={credentialProvider=SECRETS_MANAGER,credential=secret-
name-or-arn}"
```

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 se muestra 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 Source (Código fuente), para Source provider (Proveedor de código fuente), 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).



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, 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 .

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 la 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, **`appspec.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 el ejemplo siguiente, 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.

Para obtener información sobre cómo crear una canalización que utilice varias entradas de origen CodeBuild para crear varios artefactos de salida, consulteEjemplo de una CodePipeline/CodeBuild integración con múltiples fuentes de entrada y artefactos de salida.

Temas

- Creación de un proyecto de compilación con varias entradas y salidas
- Creación de un proyecto de compilación sin un origen

Creación de un proyecto de compilación con varias entradas y salidas

Utilice el siguiente procedimiento para crear un proyecto de compilación con varias entradas y salidas.

Para crear un proyecto de compilación con varias entradas y salidas

1. Sube tus fuentes a uno o más repositorios de S3 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. Cree el proyecto de compilación, ejecute la compilación y obtenga información sobre esta.
5. Si utilizas el AWS CLI para crear el proyecto de compilación, la JSON-formatted entrada del `create-project` comando puede 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": [ss
    {
      "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 API de CodeBuild , el valor `artifactIdentifier` de un artefacto secundario es un atributo obligatorio en `CreateProject` y `UpdateProject`. Debe utilizarse para hacer referencia a un artefacto secundario.

Con la JSON-formatted entrada anterior, el archivo buildspec del proyecto podría tener el siguiente 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 JSON-formatted entrada del `start-build` comando en el AWS CLI podría tener el siguiente aspecto:

```
{
   "projectName": "sample-project",
   "secondarySourcesVersionOverride": [
      {
        "sourceIdentifier": "source1",
        "sourceVersion": "codecommit-branch"
```

```
      },
      {
        "sourceIdentifier": "source2",
        "sourceVersion": "github-branch"
      },
   ]
}
```

Creación de un proyecto de compilación sin un origen

Puedes configurar un CodeBuild proyecto eligiendo el tipo **`NO_SOURCE`** de fuente al configurar tu 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, debes especificar una cadena de YAML-formatted especificaciones de construcción en el `buildspec` atributo de la JSON-formatted entrada del 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 Cree un proyecto de compilación (AWS CLI).

Versiones en tiempo de ejecución 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 Ubunto 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.

Los siguientes ejemplos muestran cómo puede cambiar el tiempo de ejecución de su proyecto, especificar más de uno y especificar un tiempo de ejecución que dependa 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 Ejecute AWS CodeBuild compilaciones manualmente y Creación de un proyecto de compilación en AWS CodeBuild.

Temas

- Actualización de la versión del tiempo de ejecución en el archivo de especificación de compilación
- Especificación de dos entornos de ejecución

Actualización de la versión del tiempo de ejecución en el archivo de especificación de compilación

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. Abre 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/amazonlinux-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 compilación. CodeBuild Este proyecto de ejemplo usa dos archivos fuente: uno que usa el motor de ejecución Go y otro que usa el Node.js tiempo de ejecución.

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`. `runtime-versions`En esta sección se especifican los tiempos de ejecución de la Node.js versión 12 y Go de la versión 1.13.

```
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 GitHub de Bitbucket.

> **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/amazonlinux-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 los resultados de Go y los tiempos de ejecución. Node.js También muestra los resultados del Go y de Node.js las aplicaciones.

```
[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 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`).
  - Branch como nombre de sucursal.
  - ID de confirmación.
  - Etiqueta.
  - Referencia y un identificador de confirmación. La referencia puede ser una de las siguientes:
    - 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 GitLab autogestionarse GitLab y autogestionarse, utilice una de las siguientes opciones:
  - Branch como nombre de sucursal.
  - 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.

Temas

- Especifica una versión GitHub del repositorio con un ID de confirmación
- Especifique una GitHub versión del repositorio con una referencia y un ID de confirmación

Especifica una versión GitHub del repositorio con un ID de 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.

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/amazonlinux-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`.

Especifique una GitHub versión del repositorio con una referencia y un ID de confirmación

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 haces, CodeBuild descarga solo la rama especificada para buscar la versión.

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

1. Realice los pasos que se indican en Especifica una versión GitHub del repositorio con un ID de confirmación.
2. En el panel de navegación izquierdo, elija Proyectos de compilación y, a continuación, elija el proyecto que creó anteriormente.
3. Seleccione Iniciar la compilación.

4. 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`*}.
5. Seleccione Iniciar la compilación.
6. 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.

Third-party muestras de repositorios de origen para CodeBuild

En esta sección se describen ejemplos de integraciones entre repositorios de fuentes de terceros y. CodeBuild

Muestra	Description (Descripción)
BitBucket Ejemplo de filtro pull request y webhook: consulte Ejecuta el ejemplo «Filtro de webhooks y solicitudes de extracción de Bitbucket» para CodeBuild	En este ejemplo se muestra cómo crear una solicitud de extracción mediante un repositor io de Bitbucket. También se muestra cómo utilizar un webhook de Bitbucket para activar CodeBuild con el fin de crear una compilación de un proyecto.
GitHub Ejemplo de Enterprise Server: consulte Ejecute el ejemplo de GitHub Enterprise Server para CodeBuild	En este ejemplo, se muestra cómo configurar CodeBuild los proyectos cuando el repositorio de GitHub Enterprise Server tiene un certifica

Muestra	Description (Descripción)
	do instalado. También muestra cómo habilitar los webhooks para que CodeBuild reconstruyan el código fuente cada vez que se introduzca un cambio de código en el repositorio de GitHub Enterprise Server.
GitHub Ejemplo de filtro de webhook y pull request: consulta Ejecuta el ejemplo de filtro de solicitudes de GitHub extracción y webhook para CodeBuild	En este ejemplo, se muestra cómo crear una solicitud de extracción mediante un repositorio de GitHub Enterprise Server. También se muestra cómo habilitar los webhooks para que CodeBuild se reconstruya el código fuente cada vez que se introduce un cambio de código en el repositorio de GitHub Enterprise Server.

Ejecuta el ejemplo «Filtro de webhooks y solicitudes de extracción 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

- Paso 1: creación de un proyecto de compilación con Bitbucket y habilitación de webhooks
- Paso 2: 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.

Paso 1: creación de un proyecto de compilación con Bitbucket 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
2. Si se muestra 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. Para un token de API de Bitbucket, introduce la dirección de correo electrónico y el token de API de tu cuenta de Atlassian en su lugar.

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 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 ¿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 la 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.

Paso 2: 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 consola en AWS CodeBuild . 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 “Paso 1: creación de un proyecto de compilación con Bitbucket 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.

Ejecute el ejemplo de GitHub 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 CodeBuild recompilar el código fuente cada vez que se introduzca un cambio de código en el repositorio de GitHub Enterprise Server.

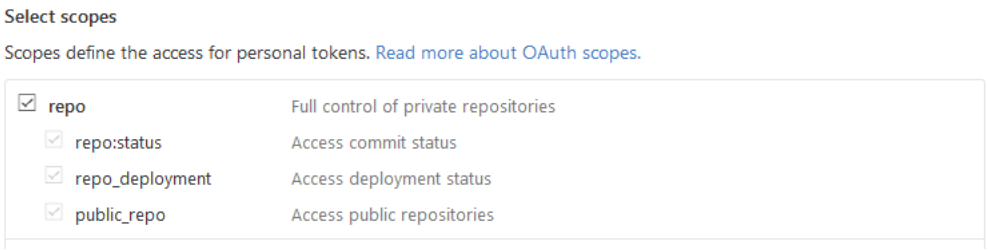
Temas

- Requisitos previos
- Paso 1: Crea un proyecto de compilación con GitHub Enterprise Server y habilita los webhooks

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. Cópielo en su portapapeles para poder 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.

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

   Linux/macOS clientes:

   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 utiliza para conectarse (por ejemplo, 443).

folder. La carpeta en la que descargaste tu certificado.

filename. El nombre de archivo de su archivo de certificado.

> **Important**
>
> Guarde el certificado como un archivo .pem.

Clientes de Windows:

Utilice su navegador para descargar el certificado desde 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.

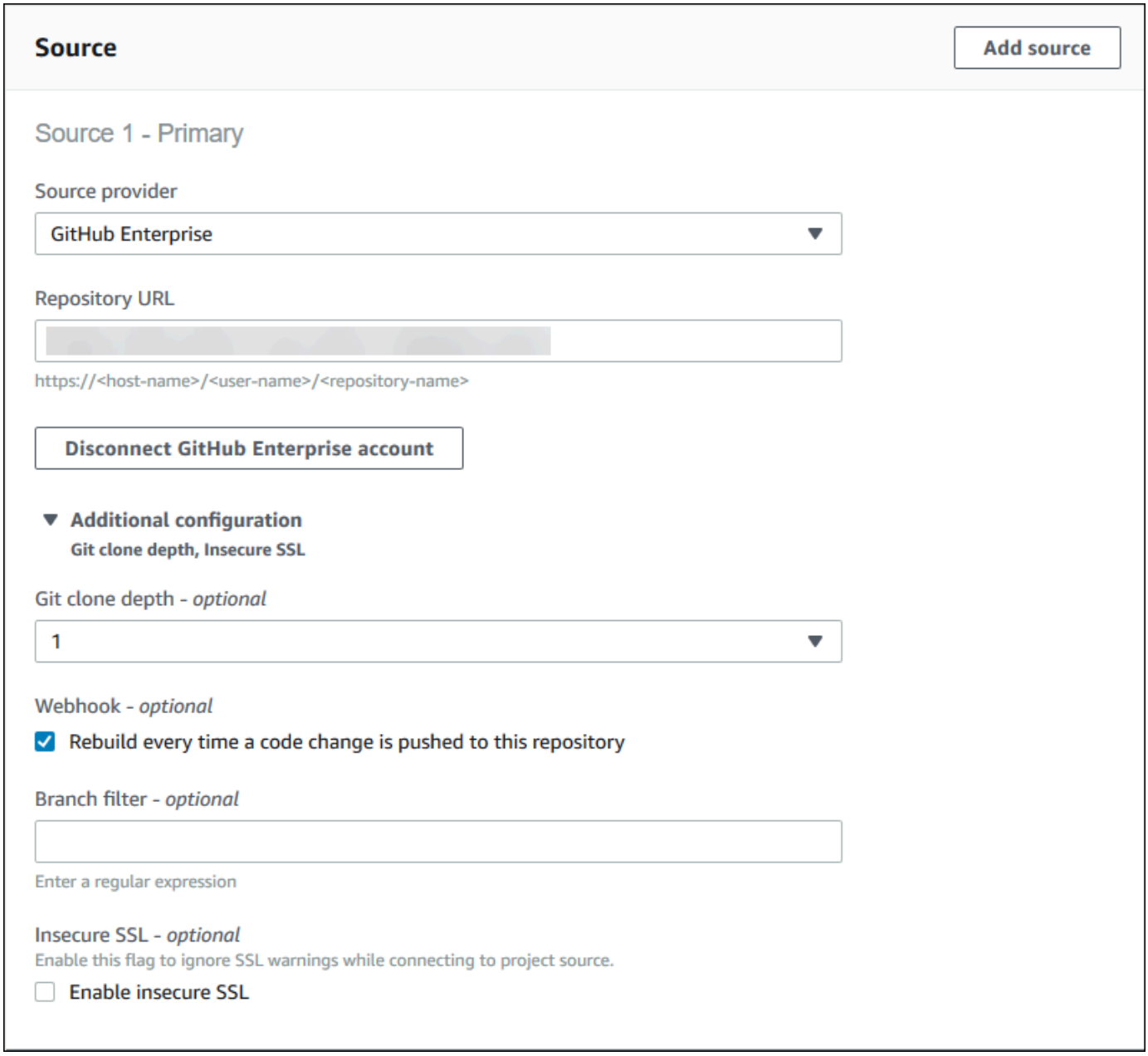
3. Cargue el archivo de certificado en un bucket de S3. Para obtener información acerca de cómo crear un bucket de S3, consulte ¿Cómo puedo crear un bucket de S3? Para obtener información acerca de cómo cargar objetos en un bucket de S3, consulte ¿Cómo puedo cargar archivos y carpetas en un bucket?

> **Note**
>
> Este bucket debe estar en la misma AWS región que tus compilaciones. Por ejemplo, si CodeBuild ordenas ejecutar una compilación en la región Este de EE. UU. (Ohio), el depósito debe estar en la región Este de EE. UU. (Ohio).

Paso 1: Crea un proyecto de compilación con GitHub Enterprise Server y habilita los webhooks

1. Abre la AWS CodeBuild consola en. https://console.aws.amazon.com/codesuite/codebuild/home
2. Si aparece una página de CodeBuild información, seleccione 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.
3. En Project name (Nombre de proyecto), escriba un nombre para este proyecto de compilación. Los nombres de los proyectos de construcción 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, elija GitHub Enterprise Server.
   - Seleccione Administrar credenciales de cuenta y, a continuación, Token de acceso personal. En Servicio, seleccione Secrets Manager (recomendado) y configure su secreto. A continuación, en Token de acceso personal GitHub empresarial, introduce tu token de acceso personal y selecciona Guardar.
   - 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.
   - Selecciona Habilitar el SSL no seguro para ignorar las advertencias de SSL mientras te conectas al repositorio del proyecto 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.



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, seleccione Imagen gestionada y, a continuación, seleccione el sistema operativo, los tiempos de ejecución, la imagen y la 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 eliges Amazon ECR, usa el repositorio de Amazon ECR y la imagen de Amazon ECR para elegir la imagen de Docker en tu 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 usas 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:

   - En el caso de la VPC, elija el ID de VPC que utilice. CodeBuild
   - Para las subredes de VPC, elige las subredes que incluyen los recursos que usa. CodeBuild
   - En el caso de 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 la 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, `appspec.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 de las compilaciones en caché para mejorar el rendimiento.

11. Elija Crear el proyecto de compilación. En la página del proyecto de compilación, elija Start build (Iniciar compilación).

Ejecuta el ejemplo de filtro de solicitudes de GitHub extracción 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.

Temas

- Paso 1: Crea un proyecto de compilación con webhooks GitHub y actívalos
- Paso 2: comprobación de que los webhooks estén habilitados

Paso 1: Crea un proyecto de compilación con webhooks GitHub y actívalos

1. Abre 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

   Elija 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 la 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.

Paso 2: comprobación de que los webhooks estén habilitados

1. Abre 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:

   - 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 y 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.

Tutorial: firma de código de Apple con Fastlane al CodeBuild usar S3 para el almacenamiento de certificados

fastlane es una conocida herramienta de automatización de código abierto para automatizar implementaciones y versiones beta de aplicaciones iOS y Android. Se encarga de todas las tareas tediosas, como generar capturas de pantalla, administrar la firma de código y publicar la aplicación.

Requisitos previos

Para completar este tutorial, debe configurar lo siguiente:

- ¿Un Cuenta de AWS
- Una cuenta de desarrollador de Apple
- Un bucket de S3 para almacenar certificados
- fastlane instalado en el proyecto: Guía para instalar fastlane

Paso 1: configuración de Fastlane Match con S3 en el equipo local

Fastlane Match es una de las herramientas de Fastlane y permite configurar sin problemas la firma de código tanto en su entorno de desarrollo local como en otro. CodeBuild Fastlane Match almacena todos tus certificados de firma de código y perfiles de aprovisionamiento en un almacenamiento en repository/S3 Bucket/Google la nube de Git, y descarga e instala los certificados y perfiles necesarios cuando es necesario.

En este ejemplo de configuración, configurará y usará un bucket de Amazon S3 para almacenamiento.

1. Inicialice match en el proyecto:

```
fastlane match init
```

2. Cuando se le solicite, elija S3 como modo de almacenamiento.
3. Actualice “Matchfile” para usar S3:

```
storage_mode("s3")
   s3_bucket("your-s3-bucket-name")
   s3_region("your-aws-region")
   type("appstore") # The default type, can be: appstore, adhoc, enterprise or
 development
```

Paso 2: configuración del Fastfile

Cree o actualice el “Fastfile” con el siguiente flujo.

Sí CodeBuild, Fastlane Match tendrá que ejecutarse cada vez que crees y firmes tu aplicación. La forma más sencilla de hacerlo es agregar la acción `match` al flujo que compila la aplicación.

```
default_platform(:ios)

platform :ios do
  before_all do
    setup_ci
  end

  desc "Build and sign the app"
  lane :build do
    match(type: "appstore", readonly: true)
    gym(
      scheme: "YourScheme",
      export_method: "app-store"
    )
  end
end
```

> **Note**
>
> Asegúrese de agregar `setup_ci` a la sección `before_all` de `Fastfile` para que la acción de búsqueda de coincidencias funcione correctamente. Esto garantiza el uso de un llavero de Fastlane temporal con los permisos adecuados. Si no lo usa, es posible que se produzcan errores de compilación o resultados incoherentes.

Paso 3: ejecuta el **comando** fastlane match para generar los respectivos certificados y perfiles

El comando fastlane match para el tipo indicado (por ejemplo, desarrollo, tienda de aplicaciones, ad hoc, empresa) generará el certificado y el perfil si no están disponibles en el almacén remoto. Fastlane almacenará los certificados y los perfiles en S3.

```
bundle exec fastlane match appstore
```

La ejecución del comando será interactiva y fastlane solicitará establecer una frase de contraseña para descifrar los certificados.

Paso 4: creación del archivo de aplicación para el proyecto

Cree o agregue el archivo de aplicación que corresponda a su proyecto.

1. Cree o agregue Gymfile, Appfile, Snapfile, Deliverfile en función de los requisitos de creación del proyecto.
2. Confirmación de los cambios al repositorio remoto

Paso 5: creación de variables de entorno en Secrets Manager

Cree dos secretos para almacenar la cookie de sesión de fastlane y la frase de contraseña coincidente. Para obtener información sobre cómo crear un secreto en Secrets Manager, consulte Creación de un secreto de AWS Secrets Manager.

1. Acceda a la cookie de sesión de fastlane de la siguiente forma.

a. Clave del secreto: `FASTLANE_SESSION`

b. Valor secreto: cookie de sesión generada al ejecutar el siguiente comando en el equipo local.

> **Note**
>
> Este valor está disponible después de la autenticación en un archivo local: `~/.fastlane/spaceship/my_appleid_username/cookie`.

```
fastlane spaceauth -u <apple account>
```

2. Contraseña de Fastlane Match: para que Fastlane Match pueda descifrar los certificados y perfiles almacenados en el bucket de S3, es necesario añadir a las variables de entorno del proyecto la contraseña de cifrado que configuraste en el paso de configuración de Match. CodeBuild

   a. Clave del secreto: `MATCH_PASSWORD`

   b. Valor del secreto: *`<match passphrase to decrypt certificates>`* La frase de contraseña se establece al generar los certificados en el paso 3.

> **Note**
>
> Al crear los secretos anteriores en Secrets Manager, recuerde dar un nombre de secreto con el siguiente prefijo: `/CodeBuild/`

Paso 6: creación de una flota de computación

Cree la flota de computación para el proyecto.

1. En la consola, ve a y crea una nueva flota de CodeBuild procesamiento.
2. Elija “macOS” como sistema operativo y seleccione un tipo de computación y una imagen adecuados.

Paso 7: Crea un proyecto en CodeBuild

Crea tu proyecto en CodeBuild.

1. Abra 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).
3. Configure su proveedor de origen (por ejemplo GitHub, CodeCommit). Este es el repositorio de origen de proyectos de iOS y no el repositorio de certificados.
4. En Environment (Entorno):
   - Elija Capacidad reservada.
   - En Flota, seleccione la flota creada anteriormente.
   - Proporcione el nombre del rol de servicio que CodeBuild se creará para usted.
   - Proporcione las siguientes variables de entorno.
     - Nombre:`MATCH_PASSWORD`, Valor:*`<secrets arn>`*, Tipo: Gestor de secretos (el ARN de secretos se creó en el paso 5 para MATCH_PASSWORD)
     - Nombre:`FASTLANE_SESSION`, Valor:*`<secrets arn>`*, Tipo: Gestor de secretos (el ARN de secretos se creó en el paso 5 para FASTLANE_SESSION)
5. En Buildspec, agregue lo siguiente:

```
version: 0.2

phases:
  install:
    commands:
      - gem install bundler
      - bundle install
  build:
    commands:
      - echo "Building and signing the app..."
      - bundle exec fastlane build
  post_build:
    commands:
      - echo "Build completed on date"

artifacts:
```

```
files:
  - '*/.ipa'
name: app-$(date +%Y-%m-%d)
```

Paso 8: configuración del rol de IAM

Una vez creado el proyecto, asegúrate de que la función de servicio de tu CodeBuild proyecto tenga permisos para acceder al bucket de S3 que contiene los certificados. Agregue la siguiente política al rol:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation",
                "s3:ListBucket"
            ],
            "Resource": "arn:aws:s3:::your-s3-bucket-name"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:PutObject",
                "s3:DeleteObject"
            ],
            "Resource": "arn:aws:s3:::your-s3-bucket-name/*"
        }
    ]
}
```

Paso 9: ejecución de la compilación

Ejecute la compilación. Puedes revisar el estado de la compilación e iniciar sesión CodeBuild.

Una vez que se haya completado el trabajo, podrá ver el registro de este.

Resolución de problemas

- Si tiene problemas al obtener certificados, asegúrese de que los permisos de IAM estén configurados correctamente para el acceso a S3.
- Si tiene problemas al descifrar certificados, asegúrese de establecer la frase contraseña correcta en la variable de entorno MATCH_PASSWORD.
- En caso de problemas con la firma de código, compruebe que la cuenta de desarrollador de Apple tiene los certificados y los perfiles necesarios y que el identificador de agrupación del proyecto de Xcode coincide con el del perfil de aprovisionamiento.

Consideraciones de seguridad

Tenga en cuenta los siguientes aspectos de seguridad en este tutorial.

- Asegúrese de que el bucket de S3 tenga la configuración de seguridad adecuada, incluido cifrado en reposo. En concreto, asegúrate de que el bucket no tenga acceso público y restringe el acceso únicamente CodeBuild al sistema que necesite tener acceso.
- Considera usarlo AWS Secrets Manager para almacenar información confidencial, como MATCH_PASSWORD y FASTLANE_SESSION.

Este ejemplo proporciona una configuración para la firma de código de iOS con Fastlane al CodeBuild usar Amazon S3 para el almacenamiento de certificados. Es posible que deba ajustar algunos pasos en función de los requisitos y el CodeBuild entorno específicos de su proyecto. Este enfoque aprovecha los AWS servicios para mejorar la seguridad y la integración dentro del AWS ecosistema.

Tutorial: la firma de código de Apple con Fastlane GitHub para CodeBuild el almacenamiento de certificados

fastlane es una conocida herramienta de automatización de código abierto para automatizar implementaciones y versiones beta de aplicaciones iOS y Android. Se encarga de todas las tareas tediosas, como generar capturas de pantalla, administrar la firma de código y publicar la aplicación.

En este ejemplo, se muestra cómo configurar la firma de código de Apple con Fastlane en un CodeBuild proyecto que se ejecuta en Mac Fleet, GitHub como almacenamiento para los certificados y los perfiles de aprovisionamiento.

Requisitos previos

Para completar este tutorial, debe configurar lo siguiente:

- Un Cuenta de AWS
- Una cuenta de desarrollador de Apple
- Un GitHub repositorio privado para almacenar certificados
- fastlane instalado en el proyecto: Guía para instalar fastlane

Paso 1: Configura Fastlane Match con GitHub en tu máquina local

Fastlane Match es una de las herramientas de Fastlane y permite configurar sin problemas la firma de código tanto en tu entorno de desarrollo local como en otro. CodeBuild Fastlane Match almacena todos tus certificados de firma de código y perfiles de aprovisionamiento en un almacenamiento en repository/S3 Bucket/Google la nube de Git, y descarga e instala los certificados y perfiles necesarios cuando es necesario.

En este ejemplo de configuración, configuraremos y usaremos un repositorio de Git para almacenamiento.

1. Inicialice match en el proyecto:

```
fastlane match init
```

2. Cuando se te pida, elige GitHub como modo de almacenamiento.
3. Actualiza tu `Matchfile` para usar GitHub:

```
git_url("https://github.com/your-username/your-certificate-repo.git")
storage_mode("git")
type("development") # The default type, can be: appstore, adhoc, enterprise or
 development
```

> **Note**
>
> Asegúrate de introducir la URL HTTPS del repositorio de Git para que fastlane se autentique y clone correctamente. De lo contrario, puede aparecer un error de autenticación al intentar utilizar coincidencia.

Paso 2: configuración del Fastfile

Cree o actualice el “Fastfile” con el siguiente flujo.

Sí CodeBuild, Fastlane Match tendrá que ejecutarse cada vez que crees y firmes tu aplicación. La forma más sencilla de hacerlo es agregar la acción `match` al flujo que compila la aplicación.

```
default_platform(:ios)

platform :ios do
  before_all do
    setup_ci
  end

  desc "Build and sign the app"
  lane :build do
    match(type: "appstore", readonly: true)
    gym(
      scheme: "YourScheme",
      export_method: "app-store"
    )
  end
end
```

> **Note**
>
> Asegúrese de agregar `setup_ci` a la sección `before_all` de `Fastfile` para que la acción de búsqueda de coincidencias funcione correctamente. Esto garantiza el uso de un llavero de Fastlane temporal con los permisos adecuados. Si no lo usa, es posible que se produzcan errores de compilación o resultados incoherentes.

Paso 3: ejecuta el **comando** fastlane match para generar los respectivos certificados y perfiles

El comando fastlane match para el tipo indicado (por ejemplo, desarrollo, tienda de aplicaciones, adhoc, empresa) generará el certificado y el perfil si no están disponibles en el almacén remoto. Los certificados y perfiles se almacenarán en GitHub Fastlane.

```
bundle exec fastlane match appstore
```

La ejecución del comando será interactiva y fastlane solicitará establecer una frase de contraseña para descifrar los certificados.

Paso 4: creación del archivo de aplicación para el proyecto

Cree o agregue el archivo de aplicación que corresponda a su proyecto.

1. Cree o agregue Gymfile, Appfile, Snapfile, Deliverfile en función de los requisitos de creación del proyecto.
2. Confirmación de los cambios al repositorio remoto.

Paso 5: creación de variables de entorno en Secrets Manager

Cree tres secretos para almacenar la cookie de sesión de fastlane y la frase de contraseña coincidente. Para obtener información sobre cómo crear un secreto en Secrets Manager, consulte Creación de un secreto de AWS Secrets Manager.

1. Acceda a la cookie de sesión de fastlane de la siguiente forma.
   a. Clave del secreto: `FASTLANE_SESSION`
   b. Valor secreto: cookie de sesión generada al ejecutar el siguiente comando en el equipo local.

> **Note**
>
> Este valor está disponible después de la autenticación en un archivo local: `~/.fastlane/spaceship/my_appleid_username/cookie`.

```
fastlane spaceauth -u <Apple_account>
```

2. Contraseña de Fastlane Match: para que Fastlane Match pueda descifrar los certificados y perfiles almacenados en el repositorio de Git, es necesario añadir a las variables de entorno del proyecto la contraseña de cifrado que configuraste en el paso de configuración de Match. CodeBuild

   a. Clave del secreto: `MATCH_PASSWORD`

   b. Valor del secreto: `<match passphrase to decrypt certificates>` La frase de contraseña se establece al generar los certificados en el paso 3.

3. Fastlane `MATCH_GIT_BASIC_AUTHORIZATION`: establezca una autorización básica para match:

   a. Clave del secreto:

      `MATCH_GIT_BASIC_AUTHORIZATION`

   b. Valor secreto: el valor debe ser una cadena codificada en base64 de su nombre de usuario y su token de acceso personal (PAT) en formato `username:password`. Puede generarlo utilizando el comando siguiente:

      ```
      echo -n your_github_username:your_personal_access_token | base64
      ```

      Puedes generar tu PAT en la GitHub consola en Tu perfil > Ajustes > Ajustes de desarrolladores > Token de acceso personal. Para obtener más información, consulte la siguiente guía: https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/managing-your-personal-access-tokens.

> **Note**
>
> Al crear los secretos anteriores en Secrets Manager, recuerde dar un nombre de secreto con el siguiente prefijo: `/CodeBuild/`

Paso 6: creación de una flota de computación

Cree la flota de computación para el proyecto.

1. En la consola, vaya a CodeBuild y cree una nueva flota de cómputos.
2. Elija `macOS` como sistema operativo y seleccione un tipo de computación y una imagen adecuados.

Paso 7: Crea un proyecto en CodeBuild

Crea tu proyecto en CodeBuild.

1. Abra 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).
3. Configure su proveedor de origen (por ejemplo GitHub, CodeCommit). Este es el repositorio de origen de proyectos de iOS y no el repositorio de certificados.
4. En Environment (Entorno):
   - Elija Capacidad reservada.
   - En Flota, seleccione la flota creada anteriormente.
   - Proporcione el nombre del rol de servicio que CodeBuild se creará para usted.
   - Proporcione las siguientes variables de entorno.
     - Nombre:`MATCH_PASSWORD`, Valor:*`<secrets arn>`*, Tipo: Gestor de secretos (el ARN de secretos se creó en el paso 5 para MATCH_PASSWORD)
     - Nombre:`FASTLANE_SESSION`, Valor:*`<secrets arn>`*, Tipo: Gestor de secretos (el ARN de secretos se creó en el paso 5 para FASTLANE_SESSION)
     - Nombre:`MATCH_GIT_BASIC_AUTHORIZATION`, Valor:*`<secrets ARN>`*, Tipo: Secrets Manager Secrets ARN (creado en el paso 5 para) `MATCH_GIT_BASIC_AUTHORIZATION`
5. En Buildspec, agregue lo siguiente:

```
version: 0.2

phases:
  install:
    commands:
      - gem install bundler
      - bundle install
  build:
```

```
    commands:
      - echo "Building and signing the app..."
      - bundle exec fastlane build
  post_build:
    commands:
      - echo "Build completed on date"

artifacts:
  files:
    - '*/.ipa'
  name: app-$(date +%Y-%m-%d)
```

Paso 8: Ejecuta la compilación

Ejecute la compilación. Puedes revisar el estado de la compilación e iniciar sesión CodeBuild.

Una vez que se haya completado el trabajo, podrá ver el registro de este.

Resolución de problemas

- Si tienes problemas para acceder al GitHub repositorio, comprueba tu token de acceso personal y la variable de entorno MATCH_GIT_BASIC_AUTHORIZATION.
- Si tiene problemas al descifrar certificados, asegúrese de establecer la frase contraseña correcta en la variable de entorno MATCH_PASSWORD.
- En caso de problemas con la firma de código, compruebe que la cuenta de desarrollador de Apple tiene los certificados y los perfiles necesarios y que el identificador de agrupación del proyecto de Xcode coincide con el del perfil de aprovisionamiento.

Consideraciones de seguridad

Tenga en cuenta los siguientes aspectos de seguridad en este tutorial.

- Mantén la privacidad de tu GitHub repositorio de certificados y audita periódicamente el acceso.
- Considera utilizarlos AWS Secrets Manager para almacenar información confidencial, como MATCH_PASSWORD y FASTLANE_SESSION.

En este ejemplo se proporciona una configuración para la firma de código de iOS con Fastlane para el almacenamiento de certificados. CodeBuild GitHub Es posible que tengas que ajustar algunos

pasos en función de los requisitos y el CodeBuild entorno específicos de tu proyecto. Este enfoque aprovecha los AWS servicios para mejorar la seguridad y la integración dentro del AWS ecosistema.

Establecimiento de nombres de artefacto en el momento de la compilación con control de versiones semánticas

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 usa el AWS CLI, establézcalo `overrideArtifactName` en true en el JSON-formatted archivo pasado `acreate-project`. Para obtener más información, consulte Cree 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 se especifica un nombre de artefacto que utiliza una variable de entorno de 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 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.

Ejecute 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 ejemplos de Windows

Utilice el siguiente procedimiento para ejecutar los ejemplos de Windows.

Para ejecutar los ejemplos de Windows

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

   > **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 JSON-formatted entrada del `create-project` comando 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 Ejecución de compilaciones de forma manual.
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.Formatters.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.Formatters.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.EditorBrows
 _
  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.EditorBrows
_
        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>
```

Planifica una construcción AWS CodeBuild

Antes de usarlo AWS CodeBuild, debe responder a estas preguntas:

1. ¿Dónde está almacenado 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	Obligatorio	Documentación
CodeCommit	Nombre del repositorio. (Opcional) ID de confirmación asociado con el código fuente.	Consulte estos temas en la Guía del usuario de AWS CodeCommit : Crea un repositorio CodeCommit Crea una confirmación en CodeCommit
Amazon S3	Nombre del bucket de entrada. Nombre de objeto correspondiente al archivo ZIP de entrada de la compilación que contiene el código fuente. (Opcional) ID de versión asociado al	Consulte estos temas en la Guía de introducción de Amazon S3: Crear un bucket Añadir un objeto a un bucket.

Proveedor de repositorio	Obligatorio	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

2. ¿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 la Referencia de la especificación de compilación.
3. ¿Qué tiempos de ejecución y herramientas necesita para ejecutar la compilación? Por ejemplo, ¿está compilando para Java, Ruby, Python o Node.js? ¿Necesita la compilación Maven o Ant, o un compilador para Java, Ruby o Python? ¿La compilación necesita Git AWS CLI, the 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:

- Utilice la AWS CodeBuild consola o AWS CLI los AWS SDK. Para obtener más información, consulte Ejecución de compilaciones de forma manual.
- Cree o identifique una canalización y AWS 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 CodeBuild con CodePipeline.

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 una colección de comandos de compilación y ajustes relacionados, en formato YAML, que se CodeBuild usa 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 CodeBuild funciona.

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 de especificaciones de construcción 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 Cambio de la configuración del proyecto de compilación.
- Ejecuta el AWS CLI `start-build` comando y establece el `buildspecOverride` valor de la ruta al archivo de especificaciones de construcción 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 Ejecución de compilaciones de forma manual.
- En una AWS CloudFormation plantilla, establece la `BuildSpec` propiedad de `Source` un recurso de tipo como 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, consulta la BuildSpec propiedad en el código 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:
  # build-fanout:

phases:
  install:
    run-as: Linux-user-name
    on-failure: ABORT | CONTINUE | RETRY | RETRY-count | RETRY-regex |
 RETRY-count-regex
    runtime-versions:
      runtime: version
      runtime: version
    commands:
      - command
      - command
    finally:
      - command
      - command

  pre_build:
```

```
    run-as: Linux-user-name
    on-failure: ABORT | CONTINUE | RETRY | RETRY-count | RETRY-regex |
 RETRY-count-regex
    commands:
      - command
      - command
    finally:
      - command
      - command

  build:
    run-as: Linux-user-name
    on-failure: ABORT | CONTINUE | RETRY | RETRY-count | RETRY-regex |
 RETRY-count-regex
    commands:
      - command
      - command
    finally:
      - command
      - command

  post_build:
    run-as: Linux-user-name
    on-failure: ABORT | CONTINUE | RETRY | RETRY-count | RETRY-regex |
 RETRY-count-regex
    commands:
      - command
      - command
    finally:
      - command
      - command

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:
  key: key
  fallback-keys:
    - fallback-key
    - fallback-key
  action: restore | save
  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, en los CodeBuild registros se oculta 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.

> **Note**
>
> El enmascaramiento coincide con el valor exacto almacenado en el almacén de parámetros de Amazon EC2 Systems Manager o. AWS Secrets Manager Si un comando de compilación transforma un secreto antes de que aparezca en los registros, el resultado es una cadena diferente que no está enmascarada. Por ejemplo, al codificarlo con Base64, invertirlo, cambiarlo entre mayúsculas y minúsculas o concatenarlo con otro texto, se crea un valor diferente que puede aparecer en los registros de compilación.

Para mantener un valor enmascarado, guárdalo en la forma que usa tu compilación. Por ejemplo, si su compilación requiere un Base64-encoded valor, almacene la Base64-encoded cadena directamente en AWS Secrets Manager el almacén de parámetros de Amazon EC2 Systems Manager. De este modo, el valor registrado coincide con el valor almacenado.

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 *`value`* escalares*`key`*/, donde cada asignación representa una única variable de entorno personalizada en texto sin formato. *`key`*es el nombre de la variable de entorno personalizada y *`value`* es el valor de esa 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 la 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 Ejecute AWS CodeBuild compilaciones manualmente.
- 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 Cambie la configuración del 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 *`value`* escalares*`key`*/, donde cada asignación representa una única variable de entorno personalizada almacenada en el almacén de parámetros de Amazon EC2 Systems Manager. *`key`*es el nombre que usará más adelante en los comandos de compilación para hacer referencia a esta variable de entorno personalizada y *`value`* 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**
>
> Para poder recuperar CodeBuild las variables de entorno personalizadas almacenadas en el almacén de parámetros de Amazon EC2 Systems Manager, debe añadir la

`ssm:GetParameters` acción a su rol de CodeBuild servicio. Para obtener más información, consulte CodeBuild Permitir interactuar con otros AWS servicios.
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 Ejecute AWS CodeBuild compilaciones manualmente.
- 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 Cambie la configuración del proyecto de compilación en AWS CodeBuild.
- El valor en la declaración de especificación de compilación es el que menos prioridad tiene.

env/secrets-manager

Obligatoria 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 AWS CodePipeline la exportación de variables de entorno de la etapa de creación actual a las etapas posteriores del proceso. 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 ayudante de credenciales de Git para proporcionar las credenciales de Git. `yes`si 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 Se ejecuta 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`Configúralo para tu compilación en un servidor proxy explícito para crear CloudWatch registros. 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 lo `run-as` especifica User-1 globalmente y para la `install` fase solo lo especifica

una `run-as` sentencia User-2, todos los comandos del archivo buildspec se ejecutan como, User-1 excepto los comandos de la `install` fase, 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.
- `RETRY`: vuelve a intentar la compilación hasta 3 veces con un mensaje de error que coincide con la expresión regular `.*`.
- `RETRY-`*`count`*- Vuelva a intentar la compilación un número determinado de veces, como se indica *`count`* con un mensaje de error que coincida con la expresión regular. `.*` Tenga en cuenta que *`count`* debe estar entre 0 y 100. Por ejemplo, los valores válidos incluyen `RETRY-4` y `RETRY-8`.
- `RETRY-`*`regex`*- Vuelva a intentar la compilación hasta 3 veces y utilícela *`regex`* para incluir una expresión regular que coincida con un mensaje de error especificado. Por ejemplo, los valores válidos incluyen `Retry-.*Error: Unable to connect to database.*` y `RETRY-invalid+`.
- `RETRY-`*`count`*`-`*`regex`*- Vuelva a intentar la compilación un número específico de veces, como se representa con. *`count`* Tenga en cuenta que *`count`* debe estar entre 0 y 100. También se puede utilizar *`regex`* para incluir una expresión regular que coincida con el mensaje de error. Por ejemplo, los valores válidos incluyen `Retry-3-.*connection timed out.*` y `RETRY-8-invalid+`.

Si no se especifica esta propiedad, el proceso de fallo sigue las fases de transición, como se muestra en Transiciones de fases de compilación.

> **Important**
>
> El atributo `on-failure` no se admite cuando se utiliza computación Lambda o capacidad reservada. Este atributo solo funciona con las imágenes informáticas de EC2 proporcionadas por CodeBuild.

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 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 `finally` bloque. 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/versiones en tiempo de ejecución

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 con una versión específica, una versión principal seguida de una `.x` para especificar que CodeBuild usa esa versión principal con su versión secundaria más reciente o `latest` para usar la versión principal y secundaria más reciente (por ejemplo, `ruby: 3.2nodejs: 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 elige los tiempos de ejecución predeterminados que están disponibles en la imagen que utilizas. Si especificas uno o más tiempos de ejecución, usa solo esos tiempos de ejecución. CodeBuild Si no se especifica un tiempo de ejecución dependiente, CodeBuild intenta elegir el tiempo de ejecución dependiente automáticamente.

> **Note**
>
> Si no se especifica una versión en tiempo de ejecución, CodeBuild utiliza la versión predeterminada. La versión predeterminada puede cambiar cuando una versión predeterminada anterior llega al final de su vida útil (EOL). Para evitar cambios inesperados en el entorno de compilación, recomendamos especificar una versión en tiempo de ejecución en el archivo buildspec.

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/comandos

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, de uno en uno, 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/comandos

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. CodeBuildejecuta cada comando, de uno en uno, 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/comandos

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, de uno en uno, 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/comandos

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, de uno en uno, 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 de su 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 configurada, 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 de forma recursiva 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 que se indican a continuación.

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. https://github.com/vicentllongo/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 bucket 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 CodeBuild encabezado denominado `x-amz-meta-codebuild-buildarn` que contiene el encabezado `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 valores escalares, en la que cada valor escalar representa una ubicación independiente donde CodeBuild puede encontrar artefactos de salida de la compilación en relación con la ubicación de la compilación original o, si se ha definido, 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 de forma recursiva a partir de un subdirectorio denominado. *my-subdirectory*

Al especificar las ubicaciones de los artefactos de salida de compilación, CodeBuild puede localizar la ubicación de compilación original en el entorno de compilación. 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 la overrideArtifactName marca 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 de especificaciones de construcción 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 anexada.

```
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 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 excluirán de los artefactos de compilación. 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 excude-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 identificador de artefacto de este bloque debe coincidir con un artefacto definido en el `secondaryArtifacts` atributo de tu 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 se CodeBuild pueden preparar los archivos para cargar la caché a un depósito de caché de S3. Esta secuencia no es necesaria si el tipo de caché del proyecto es `No Cache`.

cache/key

Secuencia opcional. Representa la clave principal que se usa al buscar o restaurar una caché. CodeBuild hace una coincidencia exacta con la clave principal.

A continuación se muestra un ejemplo de la clave:

```
key: npm-key-$(codebuild-hash-files package-lock.json) }
```

cache/fallback-keys

Secuencia opcional. Representa una lista de claves de reserva que se utilizan secuencialmente cuando no se puede encontrar una caché con la clave principal. Se admiten hasta cinco claves de reserva y cada una de ellas se compara mediante una búsqueda por prefijo. Esta secuencia se ignorará si no se proporciona key.

A continuación se muestra un ejemplo de claves de reserva:

```
fallback-keys:
    - npm-key-$(codebuild-hash-files package-lock.json) }
    - npm-key-
    - npm-
```

cache/action

Secuencia opcional. Especifica la acción que se va a realizar en la memoria caché. Los valores válidos son:

- `restore` que solo restaura la caché sin guardar actualizaciones.
- `save` que solo guarda la caché sin restaurar una versión anterior.

Si no se proporciona ningún valor, el valor CodeBuild predeterminado es restaurar y guardar.

cache/paths

Secuencia obligatoria. Representa las ubicaciones de la caché. Contiene una secuencia de escalares, en la que cada escalar 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 se ha establecido, 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 de forma recursiva 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 los SDK o los AWS CLI AWS SDK para declarar una especificación de compilación al crear o actualizar un proyecto de compilación, la especificación de construcción debe ser una sola cadena expresada en formato YAML, junto con los espacios en blanco y los caracteres de escape de nueva línea obligatorios. Encontrará un ejemplo en la siguiente sección.
Si utilizas AWS CodePipeline consolas CodeBuild o consolas en lugar de un 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`).
Puedes usar las CodePipeline consolas CodeBuild o consolas en lugar de un 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 en una sola cadena, para usarla con los SDK. AWS CLI AWS

```
"version: 0.2\n\nenv:\n  variables:\n    JAVA_HOME: \"/usr/lib/jvm/java-8-openjdk-
amd64\\"\n  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\nartifacts:\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 archivo `messageUtil-1.0.jar` único. 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.

Versiones de buildspec

En la siguiente tabla se muestran las versiones de especificaciones de compilación y los cambios entre versiones.

Versión	Cambios
0.2	• `environment_variables` ahora se llama `env`.

Versión	Cambios
	• `plaintext` ahora se llama `variables` . • La propiedad `type` de `artifacts` 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-gráfico`
- `batch/build-lista`
- `batch/build-matriz`
- `batch/build-abanico`

batch/build-gráfico

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 Depuración de compilaciones con Administrador de sesiones.

`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.

flota

Identificador de la flota que se va a utilizar para la tarea. Para obtener más información, consulte the section called “Ejecución de compilaciones en flotas de capacidad reservada”.

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.

> **Note**
>
> Tenga en cuenta que el tipo de equipo y la flota no se pueden proporcionar en el mismo identificador de una sola construcción.

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
    - identifier: build4
      env:
        compute-type: ARM_LAMBDA_1GB
    - identifier: build5
      env:
        fleet: fleet_name
```

batch/build-lista

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 Depuración de compilaciones con Administrador de sesiones.

`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.

flota

Identificador de la flota que se va a utilizar para la tarea. Para obtener más información, consulte the section called “Ejecución de compilaciones en flotas de capacidad reservada”.

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.

> **Note**
>
> Tenga en cuenta que el tipo de equipo y la flota no se pueden proporcionar en el mismo identificador de una sola construcción.

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
    - identifier: build3
      env:
        compute-type: ARM_LAMBDA_1GB
    - identifier: build4
      env:
        fleet: fleet_name
```

```
    - identifier: build5
      env:
        compute-type: GENERAL_LINUX_XLAGRE
```

batch/build-matriz

Define una matriz de compilación. Una matriz de compilación define las tareas con diferentes configuraciones que se ejecutan en paralelo. CodeBuild crea una compilación independiente 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.

batch/build-abanico

Define una distribución ramificada de compilación. Un fanout de compilación se usa para definir una tarea que se divide en varias compilaciones que se ejecutan en paralelo. Para obtener más información, consulte Ejecución de pruebas paralelas en compilaciones por lotes.

Este elemento contiene una tarea de compilación que se puede dividir en varias compilaciones. La sección `build-fanout` contiene las siguientes propiedades.

paralelismo

Obligatorio. El número de compilaciones que ejecutarán pruebas paralelas.

ignore-failure

Opcional. Valor booleano que indica si se puede ignorar un error en cualquiera de las tareas de compilación de distribución ramificada. Este valor de ignore-failure se aplicará a todas las compilaciones de distribución ramificada.

false

Valor predeterminado. Si falla alguna tarea de compilación de distribución ramificada, fallará compilación por lotes.

true

Aunque falle alguna tarea de compilación de distribución ramificada, podrá realizarse correctamente la compilación por lotes.

A continuación, se muestra un ejemplo de entrada de buildspec de distribución ramificada de compilación:

```
version: 0.2
```

```
batch:
   fast-fail: false
   build-fanout:
     parallelism: 5
     ignore-failure: false

phases:
  install:
    commands:
      - npm install
   build:
    commands:
      - mkdir -p test-results
      - cd test-results
      - |
        codebuild-tests-run \
         --test-command 'npx jest --runInBand --coverage' \
         --files-search "codebuild-glob-search '**/test/**/*.test.js'" \
         --sharding-strategy 'equal-distribution'
```

Para obtener más información, consulte Distribución ramificada de compilación y Utilice el comando codebuild-tests-run de la CLI .

Cree una referencia de entorno para AWS CodeBuild

Cuando llames AWS CodeBuild para ejecutar una compilación, debes proporcionar información sobre el entorno de compilación. Un entorno de compilación representa una combinación del sistema operativo, el tiempo de ejecución del lenguaje de programación y las herramientas que se CodeBuild utilizan para ejecutar una compilación. Para obtener información sobre cómo funciona un entorno de compilación, consulte ¿Cómo CodeBuild funciona.

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 proporcionas información CodeBuild sobre el entorno de compilación, especificas el identificador de una imagen de Docker en un tipo de repositorio compatible. Estos incluyen el repositorio de imágenes de CodeBuild Docker, las imágenes disponibles públicamente en Docker Hub y los repositorios de Amazon Elastic Container Registry (Amazon ECR) a los que tu AWS cuenta tiene permisos de acceso.

- Le recomendamos que utilice las imágenes de Docker almacenadas en el repositorio de imágenes de CodeBuild Docker, 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 versión principal más reciente de una imagen disponible CodeBuild y se actualiza con actualizaciones secundarias y de parches. CodeBuild optimiza la duración del aprovisionamiento de las compilaciones con imágenes compatibles almacenándolas en caché en las imágenes de máquina de Amazon (AMI) de la máquina. Si quieres aprovechar las ventajas del almacenamiento en caché y minimizar la duración del aprovisionamiento de tu compilación, selecciona 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 usar una versión más granular, como. `aws/codebuild/amazonlinux-x86_64-standard:4.0-1.0.0`

Temas

- Obtención de la lista de imágenes de Docker actuales
- Imágenes de computación de EC2
- Imágenes de computación de Lambda
- CodeBuild Imágenes obsoletas
- Tiempos de ejecución disponibles
- Versiones de tiempo de ejecución

Obtención de la lista de imágenes de Docker actuales

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 creación o en la página Editar proyecto de creación, en Imagen de entorno, seleccione Imagen gestionada. Elija valores en las listas desplegables 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 ello AWS CLI, ejecuta 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 la AWS Referencia de SDK y herramientas.

Imágenes de computación de EC2

AWS CodeBuild admite las siguientes imágenes de Docker que están disponibles para el procesamiento en EC2. CodeBuild

> **Note**
>
> 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)

Plataforma	Identificador de imagen	Definición
Amazon Linux 2	`aws/codebuild/amazonlinux-x86_64-standard:4.0`	al/standard/4.0
Amazon Linux 2023	`aws/codebuild/amazonlinux-x86_64-standard:5.0`	al/standard/5.0
Amazon Linux 2023	`aws/codebuild/amazonlinux-x86_64-standard:6.0`	al/standard/6.0
Amazon Linux 2	`aws/codebuild/amazonlinux-x86_64-standard:corretto8`	al/standard/corretto8

Plataforma	Identificador de imagen	Definición
Amazon Linux 2	`aws/codebuild/amazonlinux-x86_64-standard:corretto11`	al/standard/corretto11
Amazon Linux 2	`aws/codebuild/amazonlinux-aarch64-standard:2.0`	al/aarch64/0standard/2.
Amazon Linux 2023	`aws/codebuild/amazonlinux-aarch64-standard:3.0`	al/aarch64/0standard/3.
Ubuntu 20.04	`aws/codebuild/standard:5.0`	ubuntu/standard/5.0
Ubuntu 22.04	`aws/codebuild/standard:6.0`	ubuntu/standard/6.0
Ubuntu 22.04	`aws/codebuild/standard:7.0`	ubuntu/standard/7.0
Ubuntu 24.04	`aws/codebuild/standard:8.0`	ubuntu/standard/8.0
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

> **Note**
>
> El 22 de noviembre de 2024, los alias de las imágenes de tiempo de ejecución Linux-based estándar se actualizaron de `aamazonlinux2`. `amazonlinux` No es necesaria ninguna actualización manual, ya que los alias anteriores siguen siendo válidos.

Imágenes de computación de Lambda

AWS CodeBuild admite las siguientes imágenes de Docker que están disponibles para su AWS Lambda procesamiento. CodeBuild

> **Important**
>
> El atributo `on-failure` no se admite cuando se utiliza computación Lambda o capacidad reservada. Este atributo solo funciona con las imágenes informáticas de EC2 proporcionadas por. CodeBuild

`Arquitectura aarch64`

Plataforma	Identificador de imagen	Definición
Amazon Linux 2	`aws/codebuild/amazonlinux-aarch64-lambda-standard:dotnet6`	al- /dotnet6 lambda/aarch64
Amazon Linux 2023	`aws/codebuild/amazonlinux-aarch64-lambda-standard:dotnet8`	al- /dotnet8 lambda/aarch64
Amazon Linux 2023	`aws/codebuild/amazonlinux-aarch64-lambda-standard:dotnet10`	al- /dotnet10 lambda/aarch64

Plataforma	Identificador de imagen	Definición
Amazon Linux 2	`aws/codebuild/amazonlinux-aarch64-lambda-standard:go1.21`	al- /go1.21 lambda/aarch64
Amazon Linux 2023	`aws/codebuild/amazonlinux-aarch64-lambda-standard:go1.24`	al- /go1.24 lambda/aarch64
Amazon Linux 2023	`aws/codebuild/amazonlinux-aarch64-lambda-standard:go1.25`	al- /go1.25 lambda/aarch64
Amazon Linux 2	`aws/codebuild/amazonlinux-aarch64-lambda-standard:corretto11`	al- /correto11 lambda/aarch64
Amazon Linux 2	`aws/codebuild/amazonlinux-aarch64-lambda-standard:corretto17`	al- lambda/aarch64 /correto17
Amazon Linux 2023	`aws/codebuild/amazonlinux-aarch64-lambda-standard:corretto21`	al- lambda/aarch64 /correto21
Amazon Linux 2023	`aws/codebuild/amazonlinux-aarch64-lambda-standard:corretto25`	al- lambda/aarch64 /correto25

Plataforma	Identificador de imagen	Definición
Amazon Linux 2	`aws/codebuild/amazonlinux-aarch64-lambda-standard:nodejs18`	al- lambda/aarch64 /nodejs18
Amazon Linux 2023	`aws/codebuild/amazonlinux-aarch64-lambda-standard:nodejs20`	al- lambda/aarch64 /nodejs20
Amazon Linux 2023	`aws/codebuild/amazonlinux-aarch64-lambda-standard:nodejs22`	al- lambda/aarch64 /nodejs22
Amazon Linux 2023	`aws/codebuild/amazonlinux-aarch64-lambda-standard:nodejs24`	al- lambda/aarch64 /nodejs24
Amazon Linux 2	`aws/codebuild/amazonlinux-aarch64-lambda-standard:python3.11`	al- lambda/aarch64 /python3.11
Amazon Linux 2023	`aws/codebuild/amazonlinux-aarch64-lambda-standard:python3.12`	al- lambda/aarch64 /python3.12
Amazon Linux 2023	`aws/codebuild/amazonlinux-aarch64-lambda-standard:python3.13`	al- lambda/aarch64 /python3.13

Plataforma	Identificador de imagen	Definición
Amazon Linux 2023	`aws/codebuild/amazonlinux-aarch64-lambda-standard:python3.14`	al- lambda/aarch64 /pitón3.14
Amazon Linux 2	`aws/codebuild/amazonlinux-aarch64-lambda-standard:ruby3.2`	al- lambda/aarch64 /rubí 3.2
Amazon Linux 2023	`aws/codebuild/amazonlinux-aarch64-lambda-standard:ruby3.4`	al- /rubí 3.4 lambda/aarch64

`Arquitectura` x86_64

Plataforma	Identificador de imagen	Definición
Amazon Linux 2	`aws/codebuild/amazonlinux-x86_64-lambda-standard:dotnet6`	al- /dotnet6 lambda/x86_64
Amazon Linux 2023	`aws/codebuild/amazonlinux-x86_64-lambda-standard:dotnet8`	al- /dotnet8 lambda/x86_64
Amazon Linux 2023	`aws/codebuild/amazonlinux-x86_64-lambda-standard:dotnet10`	al- /dotnet10 lambda/x86_64

Plataforma	Identificador de imagen	Definición
Amazon Linux 2	`aws/codebuild/amazonlinux-x86_64-lambda-standard:go1.21`	al- /go1.21 lambda/x86_64
Amazon Linux 2023	`aws/codebuild/amazonlinux-x86_64-lambda-standard:go1.24`	al- /go1.24 lambda/x86_64
Amazon Linux 2023	`aws/codebuild/amazonlinux-x86_64-lambda-standard:go1.25`	al- /go1.25 lambda/x86_64
Amazon Linux 2	`aws/codebuild/amazonlinux-x86_64-lambda-standard:corretto11`	al- /correto11 lambda/x86_64
Amazon Linux 2	`aws/codebuild/amazonlinux-x86_64-lambda-standard:corretto17`	al- lambda/x86_64 /correto17
Amazon Linux 2023	`aws/codebuild/amazonlinux-x86_64-lambda-standard:corretto21`	al- lambda/x86_64 /correto21
Amazon Linux 2023	`aws/codebuild/amazonlinux-x86_64-lambda-standard:corretto25`	al- lambda/x86_64 /correto25
Amazon Linux 2	`aws/codebuild/amazonlinux-x86_64-lambda-standard:nodejs18`	al- lambda/x86_64 /nodejs18

Plataforma	Identificador de imagen	Definición
Amazon Linux 2023	`aws/codebuild/amazonlinux-x86_64-lambda-standard:nodejs20`	al- lambda/x86_64 /nodejs20
Amazon Linux 2023	`aws/codebuild/amazonlinux-x86_64-lambda-standard:nodejs22`	al- lambda/x86_64 /nodejs22
Amazon Linux 2023	`aws/codebuild/amazonlinux-x86_64-lambda-standard:nodejs24`	al- lambda/x86_64 /nodejs24
Amazon Linux 2	`aws/codebuild/amazonlinux-x86_64-lambda-standard:python3.11`	al- lambda/x86_64 /python3.11
Amazon Linux 2023	`aws/codebuild/amazonlinux-x86_64-lambda-standard:python3.12`	al- lambda/x86_64 /python3.12
Amazon Linux 2023	`aws/codebuild/amazonlinux-x86_64-lambda-standard:python3.13`	al- lambda/x86_64 /python3.13
Amazon Linux 2023	`aws/codebuild/amazonlinux-x86_64-lambda-standard:python3.14`	al- lambda/x86_64 /pitón3.14

Plataforma	Identificador de imagen	Definición
Amazon Linux 2	`aws/codebuild/amazonlinux-x86_64-lambda-standard:ruby3.2`	al- lambda/x86_64 /rubí 3.2
Amazon Linux 2023	`aws/codebuild/amazonlinux-x86_64-lambda-standard:ruby3.4`	al- /rubí 3.4 lambda/x86_64

CodeBuild Imágenes obsoletas

Una imagen obsoleta es una imagen que ya no se almacena en caché ni se actualiza. 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-standard: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

Plataforma	Identificador de imagen	Definición	Fecha de baja
Amazon Linux 2	`aws/codebuild/amazonlinux2-aarch64-standard:1.0`	al2/aarch64/0standard/1.	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-standard: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 elige los tiempos de ejecución predeterminados que están disponibles en la imagen que utilizas. Si especificas uno o más tiempos de ejecución, usa solo esos tiempos de ejecución. CodeBuild Si no se especifica un tiempo de ejecución dependiente, CodeBuild intenta elegir el tiempo de ejecución dependiente por usted.

> **Note**
>
> Si no se especifica una versión en tiempo de ejecución, CodeBuild utiliza la versión predeterminada. La versión predeterminada puede cambiar cuando una versión predeterminada anterior llega al final de su vida útil (EOL). Para evitar cambios inesperados

en el entorno de compilación, recomendamos especificar una versión en tiempo de ejecución en el archivo buildspec.

Para obtener más información, consulte Specify runtime versions in the buildspec file.

Temas

- Motores de ejecución de imágenes de Linux
- Tiempos de ejecución de imágenes de macOS
- 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	6.0	Amazon Linux 2 x86_64 Lambda están dotnet6 Amazon Linux 2 AArch64 Lambda está dotnet6 Amazon Linux 2023 x86_64 estándar: 5.0 Amazon Linux 2023 AArch64 estándar: 3.0 Estándar de Ubuntu 22.04:7.0
	8.0	Amazon Linux 2023 x86_64 estándar: 5.0

Nombre del entorno de tiempo de ejecución	Versión	Imágenes
		Estándar Amazon Linux 2023 x86_64:6.0 Amazon Linux 2023 AArch64 estándar: 3.0 Estándar Amazon Linux 2023 AArch64:4.0 Estándar Ubuntu 22.04:7.0 Estándar Ubuntu 24.04:8.0
	10.0	Amazon Linux 2023 x86_64 estándar: 5.0 Estándar Amazon Linux 2023 x86_64:6.0 Amazon Linux 2023 AArch64 estándar: 3.0 Estándar Amazon Linux 2023 AArch64:4.0 Estándar Ubuntu 22.04:7.0 Estándar Ubuntu 24.04:8.0
golang	1.20	Amazon Linux 2023 x86_64 estándar: 5.0 Amazon Linux 2023 AArch64 estándar: 3.0 Estándar Ubuntu 22.04:7.0

Nombre del entorno de tiempo de ejecución	Versión	Imágenes
	1.21	Amazon Linux 2 x86_64 Lambda están go1.21 Amazon Linux 2 AArch64 Lambda está go1.21 Amazon Linux 2023 x86_64 estándar: 5.0 Amazon Linux 2023 AArch64 estándar: 3.0 Estándar Ubuntu 22.04:7.0
	1.22	Amazon Linux 2023 x86_64 estándar: 5.0 Amazon Linux 2023 AArch64 estándar: 3.0 Estándar Ubuntu 22.04:7.0
	1.23	Amazon Linux 2023 x86_64 estándar: 5.0 Amazon Linux 2023 AArch64 estándar: 3.0 Estándar Ubuntu 22.04:7.0

Nombre del entorno de tiempo de ejecución	Versión	Imágenes
	1.24	Estándar Lambda Amazon Linux 2023 x86_64: go1.24 Estándar Lambda AArch64 de Amazon Linux 2023: go1.24 Amazon Linux 2023 x86_64 estándar: 5.0 Estándar Amazon Linux 2023 x86_64:6.0 Amazon Linux 2023 AArch64 estándar: 3.0 Estándar Amazon Linux 2023 AArch64:4.0 Estándar Ubuntu 22.04:7.0 Estándar Ubuntu 24.04:8.0
	1.25	Amazon Linux 2023 x86_64 estándar: 5.0 Estándar Amazon Linux 2023 x86_64:6.0 Amazon Linux 2023 AArch64 estándar: 3.0 Estándar Amazon Linux 2023 AArch64:4.0 Estándar Ubuntu 22.04:7.0 Estándar Ubuntu 24.04:8.0

Nombre del entorno de tiempo de ejecución	Versión	Imágenes
	1.26	Amazon Linux 2023 x86_64 estándar: 5.0 Estándar Amazon Linux 2023 x86_64:6.0 Amazon Linux 2023 AArch64 estándar: 3.0 Estándar Amazon Linux 2023 AArch64:4.0 Estándar Ubuntu 22.04:7.0 Estándar Ubuntu 24.04:8.0
java	corretto8	Amazon Linux 2 x86_64 estándar: corretto8 Amazon Linux 2023 x86_64 estándar: 5.0 Estándar Amazon Linux 2023 x86_64:6.0 Amazon Linux 2023 AArch64 estándar: 3.0 Estándar Amazon Linux 2023 AArch64:4.0 Estándar Ubuntu 22.04:7.0 Estándar Ubuntu 24.04:8.0

Nombre del entorno de tiempo de ejecución	Versión	Imágenes
	corretto11	Amazon Linux 2 x86_64 estándar: corretto11 Amazon Linux 2 x86_64 Lambda están corretto11 Amazon Linux 2 AArch64 Lambda está corretto11 Amazon Linux 2023 x86_64 estándar: 5.0 Estándar Amazon Linux 2023 x86_64:6.0 Amazon Linux 2023 AArch64 estándar: 3.0 Estándar Amazon Linux 2023 AArch64:4.0 Estándar Ubuntu 22.04:7.0 Estándar Ubuntu 24.04:8.0

Nombre del entorno de tiempo de ejecución	Versión	Imágenes
	corretto17	Amazon Linux 2 x86_64 Lambda están corretto17 Amazon Linux 2 AArch64 Lambda está corretto17 Amazon Linux 2023 x86_64 estándar: 5.0 Estándar Amazon Linux 2023 x86_64:6.0 Amazon Linux 2023 AArch64 estándar: 3.0 Estándar Amazon Linux 2023 AArch64:4.0 Estándar Ubuntu 22.04:7.0 Estándar Ubuntu 24.04:8.0

Nombre del entorno de tiempo de ejecución	Versión	Imágenes
	corretto21	Amazon Linux 2 x86_64 Lambda están corretto21 Amazon Linux 2 AArch64 Lambda está corretto21 Amazon Linux 2023 x86_64 estándar: 5.0 Estándar Amazon Linux 2023 x86_64:6.0 Amazon Linux 2023 AArch64 estándar: 3.0 Estándar Amazon Linux 2023 AArch64:4.0 Estándar Ubuntu 22.04:7.0 Estándar Ubuntu 24.04:8.0
	correcto a 25	Amazon Linux 2023 x86_64 estándar: 5.0 Estándar Amazon Linux 2023 x86_64:6.0 Amazon Linux 2023 AArch64 estándar: 3.0 Estándar Amazon Linux 2023 AArch64:4.0 Estándar Ubuntu 22.04:7.0 Estándar Ubuntu 24.04:8.0

Nombre del entorno de tiempo de ejecución	Versión	Imágenes
nodejs	18	Amazon Linux 2 x86_64 Lambda están nodejs18 Amazon Linux 2 AArch64 Lambda está nodejs18 Amazon Linux 2023 x86_64 estándar: 5.0 Amazon Linux 2023 AArch64 estándar: 3.0 Estándar Ubuntu 22.04:7.0
	20	Amazon Linux 2 x86_64 Lambda están nodejs20 Amazon Linux 2 AArch64 Lambda está nodejs20 Amazon Linux 2023 x86_64 estándar: 5.0 Amazon Linux 2023 AArch64 estándar: 3.0 Estándar Ubuntu 22.04:7.0

Nombre del entorno de tiempo de ejecución	Versión	Imágenes
	22	Amazon Linux 2023 x86_64 Estándar lambda: nodejs22 Estándar Lambda AArch64 de Amazon Linux 2023: nodejs22 Amazon Linux 2023 x86_64 estándar: 5.0 Estándar Amazon Linux 2023 x86_64:6.0 Amazon Linux 2023 AArch64 estándar: 3.0 Estándar Amazon Linux 2023 AArch64:4.0 Estándar Ubuntu 22.04:7.0 Estándar Ubuntu 24.04:8.0
	24	Amazon Linux 2023 x86_64 estándar: 5.0 Estándar Amazon Linux 2023 x86_64:6.0 Amazon Linux 2023 AArch64 estándar: 3.0 Estándar Amazon Linux 2023 AArch64:4.0 Estándar Ubuntu 22.04:7.0 Estándar Ubuntu 24.04:8.0

Nombre del entorno de tiempo de ejecución	Versión	Imágenes
	26	Amazon Linux 2023 x86_64 estándar: 5.0 Estándar Amazon Linux 2023 x86_64:6.0 Estándar Ubuntu 22.04:7.0 Estándar Ubuntu 24.04:8.0
php	8.1	Amazon Linux 2023 AArch64 estándar: 3.0
	8.2	Amazon Linux 2023 x86_64 estándar: 5.0 Estándar Amazon Linux 2023 x86_64:6.0 Amazon Linux 2023 AArch64 estándar: 3.0 Estándar Amazon Linux 2023 AArch64:4.0 Estándar Ubuntu 22.04:7.0 Estándar Ubuntu 24.04:8.0

Nombre del entorno de tiempo de ejecución	Versión	Imágenes
	8.3	Amazon Linux 2023 x86_64 estándar: 5.0 Estándar Amazon Linux 2023 x86_64:6.0 Amazon Linux 2023 AArch64 estándar: 3.0 Estándar Amazon Linux 2023 AArch64:4.0 Estándar Ubuntu 22.04:7.0 Estándar Ubuntu 24.04:8.0
	8.4	Amazon Linux 2023 x86_64 estándar: 5.0 Estándar Amazon Linux 2023 x86_64:6.0 Amazon Linux 2023 AArch64 estándar: 3.0 Estándar Amazon Linux 2023 AArch64:4.0 Estándar Ubuntu 22.04:7.0 Estándar Ubuntu 24.04:8.0

Nombre del entorno de tiempo de ejecución	Versión	Imágenes
	8.5	Amazon Linux 2023 x86_64 estándar: 5.0 Estándar Amazon Linux 2023 x86_64:6.0 Amazon Linux 2023 AArch64 estándar: 3.0 Estándar Amazon Linux 2023 AArch64:4.0 Estándar Ubuntu 22.04:7.0 Estándar Ubuntu 24.04:8.0
python	3.9	Amazon Linux 2023 x86_64 estándar: 5.0 Amazon Linux 2023 AArch64 estándar: 3.0 Estándar Ubuntu 22.04:7.0

Nombre del entorno de tiempo de ejecución	Versión	Imágenes
	3.10	Amazon Linux 2023 x86_64 estándar: 5.0 Estándar Amazon Linux 2023 x86_64:6.0 Amazon Linux 2023 AArch64 estándar: 3.0 Estándar Amazon Linux 2023 AArch64:4.0 Estándar Ubuntu 22.04:7.0 Estándar Ubuntu 24.04:8.0
	3.11	Amazon Linux 2 x86_64 Lambda están python3.11 Amazon Linux 2 AArch64 Lambda está python3.11 Amazon Linux 2023 x86_64 estándar: 5.0 Estándar Amazon Linux 2023 x86_64:6.0 Amazon Linux 2023 AArch64 estándar: 3.0 Estándar Amazon Linux 2023 AArch64:4.0 Estándar Ubuntu 22.04:7.0 Estándar Ubuntu 24.04:8.0

Nombre del entorno de tiempo de ejecución	Versión	Imágenes
	3.12	Amazon Linux 2 x86_64 Lambda están python3.12 Amazon Linux 2 AArch64 Lambda está python3.12 Amazon Linux 2023 x86_64 estándar: 5.0 Estándar Amazon Linux 2023 x86_64:6.0 Amazon Linux 2023 AArch64 estándar: 3.0 Estándar Amazon Linux 2023 AArch64:4.0 Estándar Ubuntu 22.04:7.0 Estándar Ubuntu 24.04:8.0

Nombre del entorno de tiempo de ejecución	Versión	Imágenes
	3.13	Amazon Linux 2023 x86_64 Estándar lambda: python 3.13 Estándar Lambda AArch64 de Amazon Linux 2023: python 3.13 Amazon Linux 2023 x86_64 estándar: 5.0 Estándar Amazon Linux 2023 x86_64:6.0 Amazon Linux 2023 AArch64 estándar: 3.0 Estándar Amazon Linux 2023 AArch64:4.0 Estándar Ubuntu 22.04:7.0 Estándar Ubuntu 24.04:8.0

Nombre del entorno de tiempo de ejecución	Versión	Imágenes
	3.14	Amazon Linux 2023 x86_64 estándar: 5.0 Estándar Amazon Linux 2023 x86_64:6.0 Amazon Linux 2023 AArch64 estándar: 3.0 Estándar Amazon Linux 2023 AArch64:4.0 Estándar Ubuntu 22.04:7.0 Estándar Ubuntu 24.04:8.0
ruby	3.1	Amazon Linux 2023 x86_64 estándar: 5.0 Amazon Linux 2023 AArch64 estándar: 3.0 Estándar Ubuntu 22.04:7.0
	3.2	Amazon Linux 2 x86_64 Lambda están ruby3.2 Amazon Linux 2 AArch64 Lambda está ruby3.2 Amazon Linux 2023 x86_64 estándar: 5.0 Amazon Linux 2023 AArch64 estándar: 3.0 Estándar Ubuntu 22.04:7.0

Nombre del entorno de tiempo de ejecución	Versión	Imágenes
	3.3	Amazon Linux 2023 x86_64 estándar: 5.0 Estándar Amazon Linux 2023 x86_64:6.0 Amazon Linux 2023 AArch64 estándar: 3.0 Estándar Amazon Linux 2023 AArch64:4.0 Estándar Ubuntu 22.04:7.0 Estándar Ubuntu 24.04:8.0
	3.4	Amazon Linux 2023 x86_64 Estándar lambda: ruby3.4 Estándar Lambda AArch64 de Amazon Linux 2023: ruby3.4 Amazon Linux 2023 x86_64 estándar: 5.0 Estándar Amazon Linux 2023 x86_64:6.0 Amazon Linux 2023 AArch64 estándar: 3.0 Estándar Amazon Linux 2023 AArch64:4.0 Estándar Ubuntu 22.04:7.0 Estándar Ubuntu 24.04:8.0

Nombre del entorno de tiempo de ejecución	Versión	Imágenes
	4.0	Amazon Linux 2023 x86_64 estándar: 5.0 Estándar Amazon Linux 2023 x86_64:6.0 Amazon Linux 2023 AArch64 estándar: 3.0 Estándar Amazon Linux 2023 AArch64:4.0 Estándar Ubuntu 22.04:7.0 Estándar Ubuntu 24.04:8.0
rust	1.94	Amazon Linux 2023 AArch64 estándar: 3.0 Estándar Amazon Linux 2023 AArch64:4.0
	1.97	Estándar Amazon Linux 2023 x86_64:6.0 Estándar Ubuntu 24.04:8.0

Tiempos de ejecución de imágenes de macOS

Important

Las imágenes CodeBuild seleccionadas para versiones de Mac contienen macOS y Xcode preinstalados. Al utilizar el software de Xcode, usted acepta, comprende y consiente el Acuerdo de Xcode y los kits de desarrollo de software (SDK) de Apple. Si no acepta los términos y condiciones del acuerdo, no utilice el software de Xcode. En su lugar, proporcione

sus propias imágenes de máquina de Amazon (AMI). Para obtener más información, consulte ¿Cómo se puede configurar una flota de macOS con capacidad reservada?

La siguiente tabla contiene los tiempos de ejecución disponibles compatibles con macOS.

Tiempos de ejecución de la plataforma macOS

Nombre del entorno de tiempo de ejecución	Versión	Imágenes	Notas adicionales
bash	3.2.57	macos-arm-base:15 macos-arm-base:26	
clang	17.0.0	macos-arm-base:15 macos-arm-base:26	
dotnet sdk	8.0.416	macos-arm-base:15	
	8,0417	macos-arm-base:26	
	10.0.101	macos-arm-base:15	
	10.0.102	macos-arm-base:26	
gcc	11.5.0	macos-arm-base:15 macos-arm-base:26	Disponible mediante el alias `gcc-11`
	12.4.0	macos-arm-base:15 macos-arm-base:26	Disponible mediante el alias `gcc-12`
	13.4.0	macos-arm-base:15 macos-arm-base:26	Disponible mediante el alias `gcc-13`
	14.3.0	macos-arm-base:15	Disponible mediante el alias `gcc-14`

Nombre del entorno de tiempo de ejecución	Versión	Imágenes	Notas adicionales
		macos-arm-base:26	
gnu	11.5.0	macos-arm-base:15 macos-arm-base:26	Disponible mediante el alias `gfortran-11`
	12.4.0	macos-arm-base:15 macos-arm-base:26	Disponible mediante el alias `gfortran-12`
	13.4.0	macos-arm-base:15 macos-arm-base:26	Disponible mediante el alias `gfortran-13`
	14.3.0	macos-arm-base:15 macos-arm-base:26	Disponible mediante el alias `gfortran-14`
golang	1.24.11	macos-arm-base:15 macos-arm-base:26	
	1.25.4	macos-arm-base:15 macos-arm-base:26	
java	Corretto8	macos-arm-base:15 macos-arm-base:26	
	Corretto11	macos-arm-base:15 macos-arm-base:26	
	Corretto17	macos-arm-base:15 macos-arm-base:26	

Nombre del entorno de tiempo de ejecución	Versión	Imágenes	Notas adicionales
	Corretto21	macos-arm-base:15 macos-arm-base:26	
	Correcto a 25	macos-arm-base:15 macos-arm-base: 26	
kotlin	2.2.21	macos-arm-base:15	
	2.3.0	macos-arm-base:26	
mono	6.14.1	macos-arm-base:15 macos-arm-base:26	
nodejs	20.19.6	macos-arm-base:15 macos-arm-base:26	
	2.21.1	macos-arm-base:15 macos-arm-base:26	
	24.11.1	macos-arm-base:15 macos-arm-base: 26	
perl	5.34.1	macos-arm-base:15 macos-arm-base:26	
php	8.2.29	macos-arm-base:15	
	8.2.30	macos-arm-base:26	
	8.3.28	macos-arm-base:15	

Nombre del entorno de tiempo de ejecución	Versión	Imágenes	Notas adicionales
	8.3.30	macos-arm-base:26	
	8.4.15	macos-arm-base:15	
	8.4,17	macos-arm-base:26	
	8.5.0	macos-arm-base:15	
	8,5,2	macos-arm-base:26	
python	3.10.19	macos-arm-base:15 macos-arm-base:26	
	3.11.14	macos-arm-base:15 macos-arm-base:26	
	3.12.12	macos-arm-base:15 macos-arm-base:26	
	3.13.10	macos-arm-base:15 macos-arm-base:26	
	3.14.1	macos-arm-base:15 macos-arm-base:26	
ruby	3.2.9	macos-arm-base:15 macos-arm-base:26	
	3.3.10	macos-arm-base:15 macos-arm-base:26	

Nombre del entorno de tiempo de ejecución	Versión	Imágenes	Notas adicionales
	3.4.7	macos-arm-base:15 macos-arm-base:26	
rust	1.91.1	macos-arm-base:15	
	1,92,0	macos-arm-base: 26	
swift	6.2.1	macos-arm-base:15	
	6.2.3	macos-arm-base:26	
Xcode	26.1.1	macos-arm-base:15	
	26.2	macos-arm-base:26	

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	Windows Server Core 20 estándar: versiones 1.0	Windows Server Core 20 estándar: versiones 2.0	Windows Server Core 2019 estándar: versiones 3.0
dotnet	3.1 5.0	3.1 6.0 7.0	8.0
dotnet sdk	3.1 5.0	3.1 6.0	8.0

Nombre del entorno de tiempo de ejecución	Windows Server Core 20 estándar: versiones 1.0	Windows Server Core 20 estándar: versiones 2.0	Windows Server Core 2019 estándar: versiones 3.0
		7.0	
golang	1.14	1,18	1,21 1,22 1,23
gradle	6.7	7.6	8,12
java	Corretto11	Corretto11 Corretto17	Corretto8 Corretto11 Corretto17 Corretto21
maven	3.6	3.8	3.9
nodejs	14,15	16,19	20,18 22,13
php	7.4	8.1	8.3 8.4
powershell	7.1	7.2	7.4
python	3.8	3,10	3,10 3,11 3,12 3.13

Nombre del entorno de tiempo de ejecución	Windows Server Core 20 estándar: versiones 1.0	Windows Server Core 20 estándar: versiones 2.0	Windows Server Core 2019 estándar: versiones 3.0
ruby	2.7	3.1	3.2 3.3 3.4

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 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
dotnet	6.0	`dotnet: 6.0`	`dotnet: 6.x`	`dotnet: latest`
	8.0	`dotnet: 8.0`	`dotnet: 8.x`	
	10.0	`dotnet: 10.0`	`dotnet: 10.x`	
golang	1,20	`golang: 1.20`	`golang: 1.x`	`golang: latest`
	1,21	`golang: 1.21`		
	1,22	`golang: 1.22`		

Nombre del entorno de tiempo de ejecución	Versión	Versión específica	Versión principal específica y última versión secundaria	Última versión
	1,23	`golang: 1.23`		
	1,24	`golang: 1.24`		
	1,25	`golang: 1.25`		
	1,26	`golang: 1.26`		
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`	
	correcto a 25	`java: corretto 5`	`java: corretto 5.x`	
nodejs	18	`nodejs: 18`	`nodejs: 18.x`	`nodejs: latest`
	20	`nodejs: 20`	`nodejs: 20.x`	
	22	`nodejs: 22`	`nodejs: 22.x`	
	24	`nodejs: 24`	`nodejs: 24.x`	
php	8.1	`php: 8.1`	`php: 8.x`	`php: latest`
	8.2	`php: 8.2`		

Nombre del entorno de tiempo de ejecución	Versión	Versión específica	Versión principal específica y última versión secundaria	Última versión
	8.3	`php: 8.3`		
	8.4	`php: 8.4`		
	8.5	`php: 8.5`		
python	3.9	`python: 3.9`	`python: 3.x`	`python: latest`
	3,10	`python: 3.10`		
	3,11	`python: 3.11`		
	3,12	`python: 3.12`		
	3.13	`python: 3.13`		
	3,14	`python: 3.14`		
ruby	3.1	`ruby: 3.1`	`ruby: 3.x`	`ruby: latest`
	3.2	`ruby: 3.2`		
	3.3	`ruby: 3.3`		
	3.4	`ruby: 3.4`		
	4.0	`ruby: 4.0`	`ruby: 4.x`	
rust	1,94	`rust: 1.94`	`rust: 1.x`	`rust: latest`

Puedes 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 usar las versiones de tiempo de ejecución preinstaladas en las CodeBuild-managed imágenes, puedes instalar y usar las versiones personalizadas que prefieras. En la tabla siguiente se muestran los tiempos de ejecución personalizados disponibles y cómo especificarlos.

> **Note**
>
> La selección de versiones de tiempo de ejecución personalizadas solo se admite para imágenes de Ubuntu y Amazon Linux.

Versiones de tiempo de ejecución personalizadas

Nombre del entorno de tiempo de ejecución	Sintaxis	Ejemplo
dotnet	`<major>.<minor>.<patch>`	`5.0.408`
golang	`<major>.<minor>` `<major>.<minor>.<patch>`	`1.19` `1.19.1`
java	`corretto<major>`	`corretto15`

nodejs	`<major>`
	`<major>.<minor>`
	`<major>.<minor>.<patch>`
		`14`
		`14.21`
		`14.21.3`

php	`<major>.<minor>.<patch>`	`8.0.30`

python	`<major>`
	`<major>.<minor>`
	`<major>.<minor>.<patch>`
		`3`
		`3.7`
		`3.7.16`

ruby	`<major>.<minor>.<patch>`	`3.0.6`

Ejemplo de especificación de compilación de tiempo de ejecución personalizada

Este es un ejemplo de una especificación de compilación que indica las 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, puedes especificar la imagen del entorno de procesamiento y ejecución que se CodeBuild utiliza para ejecutar tus compilaciones. La computación hace referencia 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 computación
- Acerca de los tipos de entorno de capacidad reservada
- Acerca de los tipos de entorno bajo demanda

Acerca de computación

CodeBuild ofrece los modos EC2 y de AWS Lambda computación. EC2 ofrece una flexibilidad optimizada durante la construcción y AWS Lambda velocidades de arranque optimizadas. AWS Lambda admite compilaciones más rápidas debido a una menor latencia de inicio. AWS Lambda también se escala automáticamente, por lo que las compilaciones no esperan en la cola para ejecutarse. Para obtener más información, consulte Ejecutar se basa en AWS Lambda cálculo.

Con el modo de computación EC2, puede ejecutar compilaciones con flotas de capacidad reservada o bajo demanda. En el caso de las flotas bajo demanda, puedes seleccionar tipos de procesamiento predefinidos, como o. `BUILD_GENERAL1_SMALL` `BUILD_GENERAL1_LARGE` Para obtener más información, consulte Acerca de los tipos de entorno bajo demanda. Para flotas de capacidad reservada, puede seleccionar configuraciones de computación, incluidas vCPU, memoria y espacio en disco. Después de especificar las configuraciones, CodeBuild elegirá un tipo de procesamiento compatible que se adapte a sus requisitos. Para obtener más información, consulte Acerca de los tipos de entorno de capacidad reservada.

Acerca de los tipos de entorno de capacidad reservada

AWS CodeBuild proporciona los tipos de entorno Linux x86, Arm, GPU, Windows y macOS para flotas de capacidad reservada. En la siguiente tabla se muestran el tipo de equipo, memoria, vCPU y espacio en disco disponibles, ordenados por región:

US East (N. Virginia)

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
ARM	2	4 GiB	64 GB	`GENERAL`	`reserved.arm.2cpu.4gib`
ARM	4	8 GiB	128 GB	`GENERAL`	`reserved.arm.4cpu.8gib`
ARM	8	16 GiB	128 GB	`GENERAL`	`reserved.arm.8cpu.16gib`
ARM	16	32 GiB	256 GB	`GENERAL`	`reserved.arm.16cpu.32gib`

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
ARM	32	64 GiB	256 GB	GENERAL	reserved.arm.32cpu.64gib
ARM	48	96 GiB	512 GB	GENERAL	reserved.arm.48cpu.96gib
ARM	64	128 GiB	824 GB	GENERAL	reserved.arm.64cpu.128gib
ARM EC2	2	4 GiB	64 GB	GENERAL	reserved.arm.2cpu.4gib
ARM EC2	4	8 GiB	128 GB	GENERAL	reserved.arm.4cpu.8gib
ARM EC2	8	16 GiB	128 GB	GENERAL	reserved.arm.8cpu.16gib
Linux	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib
Linux	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
Linux	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib
Linux	16	32 GiB	256 GB	GENERAL	reserved.x86-64.16cpu.32gib
Linux	36	72 GiB	256 GB	GENERAL	reserved.x86-64.36cpu.72gib
Linux	48	96 GiB	512 GB	GENERAL	reserved.x86-64.48cpu.96gib
Linux	72	144 GiB	824 GB	GENERAL	reserved.x86-64.72cpu.144gib
Linux	96	192 GiB	824 GB	GENERAL	reserved.x86-64.96cpu.192gib
Linux	48	96 GiB	824 GB (SSD)	NVME	reserved.x86-64.48cpu.96gib.nvme

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
Linux	72	144 GiB	824 GB (SSD)	NVME	reserved.x86-64.72cpu.144gib.nvme
Linux EC2	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib
Linux EC2	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Linux EC2	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib
Linux GPU	4	16 GiB	235 GB (SSD)	NVME	reserved.gpu.4cpu.16gib.nvme
Linux GPU	8	32 GiB	435 GB (SSD)	NVME	reserved.gpu.8cpu.32gib.nvme
Linux GPU	16	64 GiB	585 GB (SSD)	NVME	reserved.gpu.16cpu.64gib.nvme

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
Linux GPU	32	128 GiB	85 GB (SSD)	NVME	reserved.gpu.32cpu.128gib.nvme
Linux GPU	48	192 GiB	3785 GB (SSD)	NVME	reserved.gpu.48cpu.192gib.nvme
Linux GPU	64	256 GiB	185 GB (SSD)	NVME	reserved.gpu.64cpu.256gib.nvme
Linux GPU	96	384 GiB	3785 GB (SSD)	NVME	reserved.gpu.96cpu.384gib.nvme
macOS	8	24 GiB	128 GB	GENERAL	reserved.arm.m2.8cpu.24gib
macOS	12	32 GiB	256 GB	GENERAL	reserved.arm.m2.12cpu.32gib
Windows	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
Windows	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Windows	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib
Windows	16	32 GiB	256 GB	GENERAL	reserved.x86-64.16cpu.32gib
Windows	32	64 GiB	256 GB	GENERAL	reserved.x86-64.32cpu.64gib
Windows	36	72 GiB	256 GB	GENERAL	reserved.x86-64.36cpu.72gib
Windows	48	96 GiB	512 GB	GENERAL	reserved.x86-64.48cpu.96gib
Windows	64	128 GiB	512 GB	GENERAL	reserved.x86-64.64cpu.128gib

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
Windows	72	144 GiB	824 GB	GENERAL	`reserved.x86-64.72cpu.144gib`
Windows	96	192 GiB	824 GB	GENERAL	`reserved.x86-64.96cpu.192gib`
Windows EC2	4	8 GiB	128 GB	GENERAL	`reserved.x86-64.4cpu.8gib`
Windows EC2	8	16 GiB	128 GB	GENERAL	`reserved.x86-64.8cpu.16gib`

Para obtener más información sobre el identificador de precios, consulte https://aws.amazon.com/codebuild/pricing/.

US East (Ohio)

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
ARM	2	4 GiB	64 GB	GENERAL	`reserved.arm.2cpu.4gib`

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
ARM	4	8 GiB	128 GB	`GENERAL`	`reserved.arm.4cpu.8gib`
ARM	8	16 GiB	128 GB	`GENERAL`	`reserved.arm.8cpu.16gib`
ARM	16	32 GiB	256 GB	`GENERAL`	`reserved.arm.16cpu.32gib`
ARM	32	64 GiB	256 GB	`GENERAL`	`reserved.arm.32cpu.64gib`
ARM	48	96 GiB	512 GB	`GENERAL`	`reserved.arm.48cpu.96gib`
ARM	64	128 GiB	824 GB	`GENERAL`	`reserved.arm.64cpu.128gib`
ARM EC2	2	4 GiB	64 GB	`GENERAL`	`reserved.arm.2cpu.4gib`
ARM EC2	4	8 GiB	128 GB	`GENERAL`	`reserved.arm.4cpu.8gib`

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
ARM EC2	8	16 GiB	128 GB	`GENERAL`	`reserved.arm.8cpu.16gib`
Linux	2	4 GiB	64 GB	`GENERAL`	`reserved.x86-64.2cpu.4gib`
Linux	4	8 GiB	128 GB	`GENERAL`	`reserved.x86-64.4cpu.8gib`
Linux	8	16 GiB	128 GB	`GENERAL`	`reserved.x86-64.8cpu.16gib`
Linux	16	32 GiB	256 GB	`GENERAL`	`reserved.x86-64.16cpu.32gib`
Linux	36	72 GiB	256 GB	`GENERAL`	`reserved.x86-64.36cpu.72gib`
Linux	48	96 GiB	512 GB	`GENERAL`	`reserved.x86-64.48cpu.96gib`

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
Linux	96	192 GiB	824 GB	GENERAL	reserved.x86-64.96cpu.192gib
Linux	48	96 GiB	824 GB (SSD)	NVME	reserved.x86-64.48cpu.96gib.nvme
Linux	72	144 GiB	824 GB (SSD)	NVME	reserved.x86-64.72cpu.144gib.nvme
Linux EC2	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib
Linux EC2	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Linux EC2	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib
Linux GPU	4	16 GiB	235 GB (SSD)	NVME	reserved.gpu.4cpu.16gib.nvme

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
Linux GPU	8	32 GiB	435 GB (SSD)	NVME	reserved.gpu.8cpu.32gib.nvme
Linux GPU	16	64 GiB	585 GB (SSD)	NVME	reserved.gpu.16cpu.64gib.nvme
Linux GPU	32	128 GiB	85 GB (SSD)	NVME	reserved.gpu.32cpu.128gib.nvme
Linux GPU	48	192 GiB	3785 GB (SSD)	NVME	reserved.gpu.48cpu.192gib.nvme
macOS	8	24 GiB	128 GB	GENERAL	reserved.arm.m2.8cpu.24gib
macOS	12	32 GiB	256 GB	GENERAL	reserved.arm.m2.12cpu.32gib
Windows	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
Windows	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Windows	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib
Windows	16	32 GiB	256 GB	GENERAL	reserved.x86-64.16cpu.32gib
Windows	32	64 GiB	256 GB	GENERAL	reserved.x86-64.32cpu.64gib
Windows	36	72 GiB	256 GB	GENERAL	reserved.x86-64.36cpu.72gib
Windows	48	96 GiB	512 GB	GENERAL	reserved.x86-64.48cpu.96gib
Windows	64	128 GiB	512 GB	GENERAL	reserved.x86-64.64cpu.128gib

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
Windows	96	192 GiB	824 GB	GENERAL	reserved.x86-64.96cpu.192gib
Windows EC2	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Windows EC2	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib

Para obtener más información sobre el identificador de precios, consulte https://aws.amazon.com/codebuild/pricing/.

US West (N. California)

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
ARM	2	4 GiB	64 GB	GENERAL	reserved.arm.2cpu.4gib
ARM	4	8 GiB	128 GB	GENERAL	reserved.arm.4cpu.8gib
ARM	8	16 GiB	128 GB	GENERAL	reserved.arm.8cpu.16gib

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
ARM	16	32 GiB	256 GB	`GENERAL`	`reserved.arm.16cpu.32gib`
ARM	32	64 GiB	256 GB	`GENERAL`	`reserved.arm.32cpu.64gib`
ARM	48	96 GiB	824 GB	`GENERAL`	`reserved.arm.48cpu.96gib`
ARM	64	128 GiB	824 GB	`GENERAL`	`reserved.arm.64cpu.128gib`
ARM EC2	2	4 GiB	64 GB	`GENERAL`	`reserved.arm.2cpu.4gib`
ARM EC2	4	8 GiB	128 GB	`GENERAL`	`reserved.arm.4cpu.8gib`
ARM EC2	8	16 GiB	128 GB	`GENERAL`	`reserved.arm.8cpu.16gib`
Linux	2	4 GiB	64 GB	`GENERAL`	`reserved.x86-64.2cpu.4gib`

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
Linux	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Linux	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib
Linux	16	32 GiB	256 GB	GENERAL	reserved.x86-64.16cpu.32gib
Linux	32	64 GiB	256 GB	GENERAL	reserved.x86-64.32cpu.64gib
Linux	36	72 GiB	256 GB	GENERAL	reserved.x86-64.36cpu.72gib
Linux	48	96 GiB	512 GB	GENERAL	reserved.x86-64.48cpu.96gib
Linux	64	128 GiB	512 GB	GENERAL	reserved.x86-64.64cpu.128gib

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
Linux	72	144 GiB	824 GB	GENERAL	reserved.x86-64.72cpu.144gib
Linux	96	192 GiB	824 GB	GENERAL	reserved.x86-64.96cpu.192gib
Linux	72	144 GiB	824 GB (SSD)	NVME	reserved.x86-64.72cpu.144gib.nvme
Linux EC2	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib
Linux EC2	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Linux EC2	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib
Windows	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib
Windows	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
Windows	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib
Windows	16	32 GiB	256 GB	GENERAL	reserved.x86-64.16cpu.32gib
Windows	32	64 GiB	256 GB	GENERAL	reserved.x86-64.32cpu.64gib
Windows	36	72 GiB	256 GB	GENERAL	reserved.x86-64.36cpu.72gib
Windows	48	96 GiB	512 GB	GENERAL	reserved.x86-64.48cpu.96gib
Windows	64	128 GiB	512 GB	GENERAL	reserved.x86-64.64cpu.128gib
Windows	72	144 GiB	824 GB	GENERAL	reserved.x86-64.72cpu.144gib

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
Windows	96	192 GiB	824 GB	GENERAL	reserved.x86-64.96cpu.192gib
Windows EC2	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Windows EC2	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib

Para obtener más información sobre el identificador de precios, consulte https://aws.amazon.com/codebuild/pricing/.

US West (Oregon)

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
ARM	2	4 GiB	64 GB	GENERAL	reserved.arm.2cpu.4gib
ARM	4	8 GiB	128 GB	GENERAL	reserved.arm.4cpu.8gib
ARM	8	16 GiB	128 GB	GENERAL	reserved.arm.8cpu.16gib

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
ARM	16	32 GiB	256 GB	GENERAL	reserved.arm.16cpu.32gib
ARM	32	64 GiB	256 GB	GENERAL	reserved.arm.32cpu.64gib
ARM	48	96 GiB	512 GB	GENERAL	reserved.arm.48cpu.96gib
ARM	64	128 GiB	824 GB	GENERAL	reserved.arm.64cpu.128gib
ARM EC2	2	4 GiB	64 GB	GENERAL	reserved.arm.2cpu.4gib
ARM EC2	4	8 GiB	128 GB	GENERAL	reserved.arm.4cpu.8gib
ARM EC2	8	16 GiB	128 GB	GENERAL	reserved.arm.8cpu.16gib
Linux	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
Linux	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Linux	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib
Linux	16	32 GiB	256 GB	GENERAL	reserved.x86-64.16cpu.32gib
Linux	36	72 GiB	256 GB	GENERAL	reserved.x86-64.36cpu.72gib
Linux	48	96 GiB	512 GB	GENERAL	reserved.x86-64.48cpu.96gib
Linux	72	144 GiB	824 GB	GENERAL	reserved.x86-64.72cpu.144gib
Linux	96	192 GiB	824 GB	GENERAL	reserved.x86-64.96cpu.192gib

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
Linux	48	96 GiB	824 GB (SSD)	NVME	`reserved.x86-64.48cpu.96gib.nvme`
Linux	72	144 GiB	824 GB (SSD)	NVME	`reserved.x86-64.72cpu.144gib.nvme`
Linux EC2	2	4 GiB	64 GB	GENERAL	`reserved.x86-64.2cpu.4gib`
Linux EC2	4	8 GiB	128 GB	GENERAL	`reserved.x86-64.4cpu.8gib`
Linux EC2	8	16 GiB	128 GB	GENERAL	`reserved.x86-64.8cpu.16gib`
Linux GPU	4	16 GiB	235 GB (SSD)	NVME	`reserved.gpu.4cpu.16gib.nvme`
Linux GPU	8	32 GiB	435 GB (SSD)	NVME	`reserved.gpu.8cpu.32gib.nvme`

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
Linux GPU	16	64 GiB	585 GB (SSD)	NVME	reserved.gpu.16cpu.64gib.nvme
Linux GPU	32	128 GiB	85 GB (SSD)	NVME	reserved.gpu.32cpu.128gib.nvme
Linux GPU	48	192 GiB	3785 GB (SSD)	NVME	reserved.gpu.48cpu.192gib.nvme
Linux GPU	64	256 GiB	185 GB (SSD)	NVME	reserved.gpu.64cpu.256gib.nvme
macOS	8	24 GiB	128 GB	GENERAL	reserved.arm.m2.8cpu.24gib
macOS	12	32 GiB	256 GB	GENERAL	reserved.arm.m2.12cpu.32gib
Windows	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
Windows	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Windows	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib
Windows	16	32 GiB	256 GB	GENERAL	reserved.x86-64.16cpu.32gib
Windows	32	64 GiB	256 GB	GENERAL	reserved.x86-64.32cpu.64gib
Windows	36	72 GiB	256 GB	GENERAL	reserved.x86-64.36cpu.72gib
Windows	48	96 GiB	512 GB	GENERAL	reserved.x86-64.48cpu.96gib
Windows	64	128 GiB	512 GB	GENERAL	reserved.x86-64.64cpu.128gib

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
Windows	72	144 GiB	824 GB	GENERAL	reserved.x86-64.72cpu.144gib
Windows	96	192 GiB	824 GB	GENERAL	reserved.x86-64.96cpu.192gib
Windows EC2	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Windows EC2	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib

Para obtener más información sobre el identificador de precios, consulte https://aws.amazon.com/codebuild/pricing/.

Asia Pacific (Tokyo)

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
ARM	2	4 GiB	64 GB	GENERAL	reserved.arm.2cpu.4gib

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
ARM	4	8 GiB	128 GB	`GENERAL`	`reserved.arm.4cpu.8gib`
ARM	8	16 GiB	128 GB	`GENERAL`	`reserved.arm.8cpu.16gib`
ARM	16	32 GiB	256 GB	`GENERAL`	`reserved.arm.16cpu.32gib`
ARM	32	64 GiB	256 GB	`GENERAL`	`reserved.arm.32cpu.64gib`
ARM	48	96 GiB	512 GB	`GENERAL`	`reserved.arm.48cpu.96gib`
ARM	64	128 GiB	824 GB	`GENERAL`	`reserved.arm.64cpu.128gib`
ARM EC2	2	4 GiB	64 GB	`GENERAL`	`reserved.arm.2cpu.4gib`
ARM EC2	4	8 GiB	128 GB	`GENERAL`	`reserved.arm.4cpu.8gib`

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
ARM EC2	8	16 GiB	128 GB	GENERAL	reserved.arm.8cpu.16gib
Linux	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib
Linux	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Linux	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib
Linux	16	32 GiB	256 GB	GENERAL	reserved.x86-64.16cpu.32gib
Linux	36	72 GiB	256 GB	GENERAL	reserved.x86-64.36cpu.72gib
Linux	48	96 GiB	512 GB	GENERAL	reserved.x86-64.48cpu.96gib

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
Linux	72	144 GiB	824 GB	GENERAL	reserved.x86-64.72cpu.144gib
Linux	96	192 GiB	824 GB	GENERAL	reserved.x86-64.96cpu.192gib
Linux	48	96 GiB	824 GB (SSD)	NVME	reserved.x86-64.48cpu.96gib.nvme
Linux	72	144 GiB	824 GB (SSD)	NVME	reserved.x86-64.72cpu.144gib.nvme
Linux EC2	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib
Linux EC2	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Linux EC2	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
Linux GPU	4	16 GiB	235 GB (SSD)	NVME	`reserved.gpu.4cpu.16gib.nvme`
Linux GPU	8	32 GiB	435 GB (SSD)	NVME	`reserved.gpu.8cpu.32gib.nvme`
Linux GPU	48	192 GiB	3785 GB (SSD)	NVME	`reserved.gpu.48cpu.192gib.nvme`
Windows	2	4 GiB	64 GB	GENERAL	`reserved.x86-64.2cpu.4gib`
Windows	4	8 GiB	128 GB	GENERAL	`reserved.x86-64.4cpu.8gib`
Windows	8	16 GiB	128 GB	GENERAL	`reserved.x86-64.8cpu.16gib`
Windows	16	32 GiB	256 GB	GENERAL	`reserved.x86-64.16cpu.32gib`

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
Windows	32	64 GiB	256 GB	GENERAL	reserved.x86-64.32cpu.64gib
Windows	36	72 GiB	256 GB	GENERAL	reserved.x86-64.36cpu.72gib
Windows	48	96 GiB	512 GB	GENERAL	reserved.x86-64.48cpu.96gib
Windows	64	128 GiB	512 GB	GENERAL	reserved.x86-64.64cpu.128gib
Windows	72	144 GiB	824 GB	GENERAL	reserved.x86-64.72cpu.144gib
Windows	96	192 GiB	824 GB	GENERAL	reserved.x86-64.96cpu.192gib
Windows EC2	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
Windows EC2	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib

Para obtener más información sobre el identificador de precios, consulte https://aws.amazon.com/codebuild/pricing/.

Asia Pacific (Malaysia)

Tipo de entorno	CPU virtuales	Memoria	Tipo de máquina	Tipo de instancia de computación
ARM	2	4 GiB	GENERAL	reserved:arm.g1.small
ARM	4	8 GiB	GENERAL	reserved:arm.g1.medium
ARM	8	16 GiB	GENERAL	reserved:arm.g1.large
ARM	32	64 GiB	GENERAL	reserved:arm.g1.xlarge
ARM	48	96 GiB	GENERAL	reserved:arm.g1.2xlarge

Tipo de entorno	CPU virtuales	Memoria	Tipo de máquina	Tipo de instancia de computación
Linux	2	4 GiB	`GENERAL`	`reserved: x86-64.g1 .small`
Linux	4	8 GiB	`GENERAL`	`reserved: x86-64.g1 .medium`
Linux	8	16 GiB	`GENERAL`	`reserved: x86-64.g1 .large`

Para obtener más información sobre el identificador de precios, consulte https://aws.amazon.com/codebuild/pricing/.

Asia Pacific (New Zealand)

Tipo de entorno	CPU virtuales	Memoria	Tipo de máquina	Tipo de instancia de computación
ARM	2	4 GiB	`GENERAL`	`reserved: arm.g1.sm all`
ARM	4	8 GiB	`GENERAL`	`reserved: arm.g1.me dium`
ARM	8	16 GiB	`GENERAL`	`reserved: arm.g1.la rge`

Tipo de entorno	CPU virtuales	Memoria	Tipo de máquina	Tipo de instancia de computación
ARM	32	64 GiB	`GENERAL`	`reserved: arm.g1.xlarge`
ARM	48	96 GiB	`GENERAL`	`reserved: arm.g1.2xlarge`
Linux	2	4 GiB	`GENERAL`	`reserved: x86-64.g1.small`
Linux	4	8 GiB	`GENERAL`	`reserved: x86-64.g1.medium`
Linux	8	16 GiB	`GENERAL`	`reserved: x86-64.g1.large`

Para obtener más información sobre el identificador de precios, consulte https://aws.amazon.com/codebuild/pricing/.

Asia Pacific (Thailand)

Tipo de entorno	CPU virtuales	Memoria	Tipo de máquina	Tipo de instancia de computación
ARM	2	4 GiB	`GENERAL`	`reserved: arm.g1.small`

Tipo de entorno	CPU virtuales	Memoria	Tipo de máquina	Tipo de instancia de computación
ARM	4	8 GiB	GENERAL	reserved: arm.g1.medium
ARM	8	16 GiB	GENERAL	reserved: arm.g1.large
ARM	32	64 GiB	GENERAL	reserved: arm.g1.xlarge
ARM	48	96 GiB	GENERAL	reserved: arm.g1.2xlarge
Linux	2	4 GiB	GENERAL	reserved: x86-64.g1.small
Linux	4	8 GiB	GENERAL	reserved: x86-64.g1.medium
Linux	8	16 GiB	GENERAL	reserved: x86-64.g1.large

Para obtener más información sobre el identificador de precios, consulte https://aws.amazon.com/codebuild/pricing/.

Asia Pacific (Mumbai)

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
ARM	2	4 GiB	64 GB	GENERAL	reserved.arm.2cpu.4gib
ARM	4	8 GiB	128 GB	GENERAL	reserved.arm.4cpu.8gib
ARM	8	16 GiB	128 GB	GENERAL	reserved.arm.8cpu.16gib
ARM	16	32 GiB	256 GB	GENERAL	reserved.arm.16cpu.32gib
ARM	32	64 GiB	256 GB	GENERAL	reserved.arm.32cpu.64gib
ARM	48	96 GiB	512 GB	GENERAL	reserved.arm.48cpu.96gib
ARM	64	128 GiB	824 GB	GENERAL	reserved.arm.64cpu.128gib
ARM EC2	2	4 GiB	64 GB	GENERAL	reserved.arm.2cpu.4gib

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
ARM EC2	4	8 GiB	128 GB	GENERAL	reserved.arm.4cpu.8gib
ARM EC2	8	16 GiB	128 GB	GENERAL	reserved.arm.8cpu.16gib
Linux	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib
Linux	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Linux	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib
Linux	16	32 GiB	256 GB	GENERAL	reserved.x86-64.16cpu.32gib
Linux	36	72 GiB	256 GB	GENERAL	reserved.x86-64.36cpu.72gib
Linux	48	96 GiB	512 GB	GENERAL	reserved.x86-64.48cpu.96gib

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
Linux	72	144 GiB	824 GB	GENERAL	reserved.x86-64.72cpu.144gib
Linux	96	192 GiB	824 GB	GENERAL	reserved.x86-64.96cpu.192gib
Linux	72	144 GiB	824 GB (SSD)	NVME	reserved.x86-64.72cpu.144gib.nvme
Linux EC2	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib
Linux EC2	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Linux EC2	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib
Linux GPU	4	16 GiB	235 GB (SSD)	NVME	reserved.gpu.4cpu.16gib.nvme

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
Linux GPU	8	32 GiB	435 GB (SSD)	NVME	`reserved.gpu.8cpu.32gib.nvme`
Linux GPU	16	64 GiB	585 GB (SSD)	NVME	`reserved.gpu.16cpu.64gib.nvme`
Windows	2	4 GiB	64 GB	GENERAL	`reserved.x86-64.2cpu.4gib`
Windows	4	8 GiB	128 GB	GENERAL	`reserved.x86-64.4cpu.8gib`
Windows	8	16 GiB	128 GB	GENERAL	`reserved.x86-64.8cpu.16gib`
Windows	16	32 GiB	256 GB	GENERAL	`reserved.x86-64.16cpu.32gib`
Windows	32	64 GiB	256 GB	GENERAL	`reserved.x86-64.32cpu.64gib`

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
Windows	36	72 GiB	256 GB	GENERAL	reserved.x86-64.36cpu.72gib
Windows	48	96 GiB	512 GB	GENERAL	reserved.x86-64.48cpu.96gib
Windows	64	128 GiB	512 GB	GENERAL	reserved.x86-64.64cpu.128gib
Windows	72	144 GiB	824 GB	GENERAL	reserved.x86-64.72cpu.144gib
Windows	96	192 GiB	824 GB	GENERAL	reserved.x86-64.96cpu.192gib
Windows EC2	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Windows EC2	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib

Para obtener más información sobre el identificador de precios, consulte https://aws.amazon.com/codebuild/pricing/.

Asia Pacific (Singapore)

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
ARM	2	4 GiB	64 GB	`GENERAL`	`reserved.arm.2cpu.4gib`
ARM	4	8 GiB	128 GB	`GENERAL`	`reserved.arm.4cpu.8gib`
ARM	8	16 GiB	128 GB	`GENERAL`	`reserved.arm.8cpu.16gib`
ARM	16	32 GiB	256 GB	`GENERAL`	`reserved.arm.16cpu.32gib`
ARM	32	64 GiB	256 GB	`GENERAL`	`reserved.arm.32cpu.64gib`
ARM	48	96 GiB	512 GB	`GENERAL`	`reserved.arm.48cpu.96gib`
ARM	64	128 GiB	824 GB	`GENERAL`	`reserved.arm.64cpu.128gib`

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
ARM EC2	2	4 GiB	64 GB	GENERAL	reserved.arm.2cpu.4gib
ARM EC2	4	8 GiB	128 GB	GENERAL	reserved.arm.4cpu.8gib
ARM EC2	8	16 GiB	128 GB	GENERAL	reserved.arm.8cpu.16gib
Linux	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib
Linux	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Linux	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib
Linux	16	32 GiB	256 GB	GENERAL	reserved.x86-64.16cpu.32gib
Linux	36	72 GiB	256 GB	GENERAL	reserved.x86-64.36cpu.72gib

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
Linux	48	96 GiB	512 GB	GENERAL	reserved.x86-64.48cpu.96gib
Linux	72	144 GiB	824 GB	GENERAL	reserved.x86-64.72cpu.144gib
Linux	96	192 GiB	824 GB	GENERAL	reserved.x86-64.96cpu.192gib
Linux	48	96 GiB	824 GB (SSD)	NVME	reserved.x86-64.48cpu.96gib.nvme
Linux	72	144 GiB	824 GB (SSD)	NVME	reserved.x86-64.72cpu.144gib.nvme
Linux EC2	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib
Linux EC2	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
Linux EC2	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib
Windows	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib
Windows	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Windows	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib
Windows	16	32 GiB	256 GB	GENERAL	reserved.x86-64.16cpu.32gib
Windows	32	64 GiB	256 GB	GENERAL	reserved.x86-64.32cpu.64gib
Windows	36	72 GiB	256 GB	GENERAL	reserved.x86-64.36cpu.72gib

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
Windows	48	96 GiB	512 GB	GENERAL	reserved.x86-64.48cpu.96gib
Windows	64	128 GiB	512 GB	GENERAL	reserved.x86-64.64cpu.128gib
Windows	72	144 GiB	824 GB	GENERAL	reserved.x86-64.72cpu.144gib
Windows	96	192 GiB	824 GB	GENERAL	reserved.x86-64.96cpu.192gib
Windows EC2	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Windows EC2	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib

Para obtener más información sobre el identificador de precios, consulte https://aws.amazon.com/codebuild/pricing/.

Asia Pacific (Sydney)

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
ARM	2	4 GiB	64 GB	`GENERAL`	`reserved.arm.2cpu.4gib`
ARM	4	8 GiB	128 GB	`GENERAL`	`reserved.arm.4cpu.8gib`
ARM	8	16 GiB	128 GB	`GENERAL`	`reserved.arm.8cpu.16gib`
ARM	16	32 GiB	256 GB	`GENERAL`	`reserved.arm.16cpu.32gib`
ARM	32	64 GiB	256 GB	`GENERAL`	`reserved.arm.32cpu.64gib`
ARM	48	96 GiB	512 GB	`GENERAL`	`reserved.arm.48cpu.96gib`
ARM	64	128 GiB	824 GB	`GENERAL`	`reserved.arm.64cpu.128gib`
ARM EC2	2	4 GiB	64 GB	`GENERAL`	`reserved.arm.2cpu.4gib`

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
ARM EC2	4	8 GiB	128 GB	GENERAL	reserved.arm.4cpu.8gib
ARM EC2	8	16 GiB	128 GB	GENERAL	reserved.arm.8cpu.16gib
Linux	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib
Linux	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Linux	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib
Linux	16	32 GiB	256 GB	GENERAL	reserved.x86-64.16cpu.32gib
Linux	36	72 GiB	256 GB	GENERAL	reserved.x86-64.36cpu.72gib
Linux	48	96 GiB	512 GB	GENERAL	reserved.x86-64.48cpu.96gib

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
Linux	72	144 GiB	824 GB	GENERAL	reserved.x86-64.72cpu.144gib
Linux	96	192 GiB	824 GB	GENERAL	reserved.x86-64.96cpu.192gib
Linux	48	96 GiB	824 GB (SSD)	NVME	reserved.x86-64.48cpu.96gib.nvme
Linux	72	144 GiB	824 GB (SSD)	NVME	reserved.x86-64.72cpu.144gib.nvme
Linux EC2	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib
Linux EC2	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Linux EC2	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
Linux GPU	4	16 GiB	235 GB (SSD)	NVME	reserved.gpu.4cpu.16gib.nvme
Linux GPU	8	32 GiB	435 GB (SSD)	NVME	reserved.gpu.8cpu.32gib.nvme
Linux GPU	16	64 GiB	585 GB (SSD)	NVME	reserved.gpu.16cpu.64gib.nvme
Linux GPU	48	192 GiB	3785 GB (SSD)	NVME	reserved.gpu.48cpu.192gib.nvme
macOS	8	24 GiB	128 GB	GENERAL	reserved.arm.m2.8cpu.24gib
macOS	12	32 GiB	256 GB	GENERAL	reserved.arm.m2.12cpu.32gib
Windows	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
Windows	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Windows	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib
Windows	16	32 GiB	256 GB	GENERAL	reserved.x86-64.16cpu.32gib
Windows	32	64 GiB	256 GB	GENERAL	reserved.x86-64.32cpu.64gib
Windows	36	72 GiB	256 GB	GENERAL	reserved.x86-64.36cpu.72gib
Windows	48	96 GiB	512 GB	GENERAL	reserved.x86-64.48cpu.96gib
Windows	64	128 GiB	512 GB	GENERAL	reserved.x86-64.64cpu.128gib

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
Windows	72	144 GiB	824 GB	GENERAL	reserved.x86-64.72cpu.144gib
Windows	96	192 GiB	824 GB	GENERAL	reserved.x86-64.96cpu.192gib
Windows EC2	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Windows EC2	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib

Para obtener más información sobre el identificador de precios, consulte https://aws.amazon.com/codebuild/pricing/.

Europe (Frankfurt)

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
ARM	2	4 GiB	64 GB	GENERAL	reserved.arm.2cpu.4gib

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
ARM	4	8 GiB	128 GB	GENERAL	reserved.arm.4cpu.8gib
ARM	8	16 GiB	128 GB	GENERAL	reserved.arm.8cpu.16gib
ARM	16	32 GiB	256 GB	GENERAL	reserved.arm.16cpu.32gib
ARM	32	64 GiB	256 GB	GENERAL	reserved.arm.32cpu.64gib
ARM	48	96 GiB	512 GB	GENERAL	reserved.arm.48cpu.96gib
ARM	64	128 GiB	824 GB	GENERAL	reserved.arm.64cpu.128gib
ARM EC2	2	4 GiB	64 GB	GENERAL	reserved.arm.2cpu.4gib
ARM EC2	4	8 GiB	128 GB	GENERAL	reserved.arm.4cpu.8gib

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
ARM EC2	8	16 GiB	128 GB	GENERAL	reserved.arm.8cpu.16gib
Linux	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib
Linux	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Linux	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib
Linux	16	32 GiB	256 GB	GENERAL	reserved.x86-64.16cpu.32gib
Linux	36	72 GiB	256 GB	GENERAL	reserved.x86-64.36cpu.72gib
Linux	48	96 GiB	512 GB	GENERAL	reserved.x86-64.48cpu.96gib

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
Linux	72	144 GiB	824 GB	GENERAL	reserved.x86-64.72cpu.144gib
Linux	96	192 GiB	824 GB	GENERAL	reserved.x86-64.96cpu.192gib
Linux	48	96 GiB	824 GB (SSD)	NVME	reserved.x86-64.48cpu.96gib.nvme
Linux	72	144 GiB	824 GB (SSD)	NVME	reserved.x86-64.72cpu.144gib.nvme
Linux EC2	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib
Linux EC2	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Linux EC2	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
Linux GPU	4	16 GiB	235 GB (SSD)	NVME	`reserved.gpu.4cpu.16gib.nvme`
Linux GPU	8	32 GiB	435 GB (SSD)	NVME	`reserved.gpu.8cpu.32gib.nvme`
Linux GPU	16	64 GiB	585 GB (SSD)	NVME	`reserved.gpu.16cpu.64gib.nvme`
Linux GPU	32	128 GiB	85 GB (SSD)	NVME	`reserved.gpu.32cpu.128gib.nvme`
Linux GPU	48	192 GiB	3785 GB (SSD)	NVME	`reserved.gpu.48cpu.192gib.nvme`
macOS	8	24 GiB	128 GB	GENERAL	`reserved.arm.m2.8cpu.24gib`
Windows	2	4 GiB	64 GB	GENERAL	`reserved.x86-64.2cpu.4gib`

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
Windows	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Windows	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib
Windows	16	32 GiB	256 GB	GENERAL	reserved.x86-64.16cpu.32gib
Windows	32	64 GiB	256 GB	GENERAL	reserved.x86-64.32cpu.64gib
Windows	36	72 GiB	256 GB	GENERAL	reserved.x86-64.36cpu.72gib
Windows	48	96 GiB	512 GB	GENERAL	reserved.x86-64.48cpu.96gib
Windows	64	128 GiB	512 GB	GENERAL	reserved.x86-64.64cpu.128gib

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
Windows	72	144 GiB	824 GB	GENERAL	`reserved.x86-64.72cpu.144gib`
Windows	96	192 GiB	824 GB	GENERAL	`reserved.x86-64.96cpu.192gib`
Windows EC2	4	8 GiB	128 GB	GENERAL	`reserved.x86-64.4cpu.8gib`
Windows EC2	8	16 GiB	128 GB	GENERAL	`reserved.x86-64.8cpu.16gib`

Para obtener más información sobre el identificador de precios, consulte https://aws.amazon.com/codebuild/pricing/.

Europe (Ireland)

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
ARM	2	4 GiB	64 GB	GENERAL	`reserved.arm.2cpu.4gib`

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
ARM	4	8 GiB	128 GB	GENERAL	reserved.arm.4cpu.8gib
ARM	8	16 GiB	128 GB	GENERAL	reserved.arm.8cpu.16gib
ARM	16	32 GiB	256 GB	GENERAL	reserved.arm.16cpu.32gib
ARM	32	64 GiB	256 GB	GENERAL	reserved.arm.32cpu.64gib
ARM	48	96 GiB	512 GB	GENERAL	reserved.arm.48cpu.96gib
ARM	64	128 GiB	824 GB	GENERAL	reserved.arm.64cpu.128gib
ARM EC2	2	4 GiB	64 GB	GENERAL	reserved.arm.2cpu.4gib
ARM EC2	4	8 GiB	128 GB	GENERAL	reserved.arm.4cpu.8gib

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
ARM EC2	8	16 GiB	128 GB	`GENERAL`	`reserved.arm.8cpu.16gib`
Linux	2	4 GiB	64 GB	`GENERAL`	`reserved.x86-64.2cpu.4gib`
Linux	4	8 GiB	128 GB	`GENERAL`	`reserved.x86-64.4cpu.8gib`
Linux	8	16 GiB	128 GB	`GENERAL`	`reserved.x86-64.8cpu.16gib`
Linux	16	32 GiB	256 GB	`GENERAL`	`reserved.x86-64.16cpu.32gib`
Linux	36	72 GiB	256 GB	`GENERAL`	`reserved.x86-64.36cpu.72gib`
Linux	48	96 GiB	512 GB	`GENERAL`	`reserved.x86-64.48cpu.96gib`

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
Linux	72	144 GiB	824 GB	GENERAL	reserved.x86-64.72cpu.144gib
Linux	96	192 GiB	824 GB	GENERAL	reserved.x86-64.96cpu.192gib
Linux	48	96 GiB	824 GB (SSD)	NVME	reserved.x86-64.48cpu.96gib.nvme
Linux	72	144 GiB	824 GB (SSD)	NVME	reserved.x86-64.72cpu.144gib.nvme
Linux EC2	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib
Linux EC2	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Linux EC2	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
Linux GPU	4	16 GiB	235 GB (SSD)	NVME	`reserved.gpu.4cpu.16gib.nvme`
Linux GPU	8	32 GiB	435 GB (SSD)	NVME	`reserved.gpu.8cpu.32gib.nvme`
Linux GPU	16	64 GiB	585 GB (SSD)	NVME	`reserved.gpu.16cpu.64gib.nvme`
Linux GPU	32	128 GiB	85 GB (SSD)	NVME	`reserved.gpu.32cpu.128gib.nvme`
Linux GPU	48	192 GiB	3785 GB (SSD)	NVME	`reserved.gpu.48cpu.192gib.nvme`
Windows	2	4 GiB	64 GB	GENERAL	`reserved.x86-64.2cpu.4gib`
Windows	4	8 GiB	128 GB	GENERAL	`reserved.x86-64.4cpu.8gib`

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
Windows	8	16 GiB	128 GB	`GENERAL`	`reserved.x86-64.8cpu.16gib`
Windows	16	32 GiB	256 GB	`GENERAL`	`reserved.x86-64.16cpu.32gib`
Windows	32	64 GiB	256 GB	`GENERAL`	`reserved.x86-64.32cpu.64gib`
Windows	36	72 GiB	256 GB	`GENERAL`	`reserved.x86-64.36cpu.72gib`
Windows	48	96 GiB	512 GB	`GENERAL`	`reserved.x86-64.48cpu.96gib`
Windows	64	128 GiB	512 GB	`GENERAL`	`reserved.x86-64.64cpu.128gib`
Windows	72	144 GiB	824 GB	`GENERAL`	`reserved.x86-64.72cpu.144gib`

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
Windows	96	192 GiB	824 GB	GENERAL	reserved.x86-64.96cpu.192gib
Windows EC2	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib
Windows EC2	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib

Para obtener más información sobre el identificador de precios, consulte https://aws.amazon.com/codebuild/pricing/.

South America (São Paulo)

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
ARM	2	4 GiB	64 GB	GENERAL	reserved.arm.2cpu.4gib
ARM	4	8 GiB	128 GB	GENERAL	reserved.arm.4cpu.8gib
ARM	8	16 GiB	128 GB	GENERAL	reserved.arm.8cpu.16gib

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
ARM	16	32 GiB	256 GB	GENERAL	reserved.arm.16cpu.32gib
ARM	32	64 GiB	256 GB	GENERAL	reserved.arm.32cpu.64gib
ARM	48	96 GiB	512 GB	GENERAL	reserved.arm.48cpu.96gib
ARM EC2	2	4 GiB	64 GB	GENERAL	reserved.arm.2cpu.4gib
ARM EC2	4	8 GiB	128 GB	GENERAL	reserved.arm.4cpu.8gib
ARM EC2	8	16 GiB	128 GB	GENERAL	reserved.arm.8cpu.16gib
Linux	2	4 GiB	64 GB	GENERAL	reserved.x86-64.2cpu.4gib
Linux	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
Linux	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib
Linux	16	32 GiB	256 GB	GENERAL	reserved.x86-64.16cpu.32gib
Linux	36	72 GiB	256 GB	GENERAL	reserved.x86-64.36cpu.72gib
Linux	48	96 GiB	512 GB	GENERAL	reserved.x86-64.48cpu.96gib
Linux	72	144 GiB	824 GB	GENERAL	reserved.x86-64.72cpu.144gib
Linux	96	192 GiB	824 GB	GENERAL	reserved.x86-64.96cpu.192gib
Linux	48	96 GiB	824 GB (SSD)	NVME	reserved.x86-64.48cpu.96gib.nvme

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
Linux	72	144 GiB	824 GB (SSD)	NVME	`reserved.x86-64.72cpu.144gib.nvme`
Linux EC2	2	4 GiB	64 GB	GENERAL	`reserved.x86-64.2cpu.4gib`
Linux EC2	4	8 GiB	128 GB	GENERAL	`reserved.x86-64.4cpu.8gib`
Linux EC2	8	16 GiB	128 GB	GENERAL	`reserved.x86-64.8cpu.16gib`
Windows	2	4 GiB	64 GB	GENERAL	`reserved.x86-64.2cpu.4gib`
Windows	4	8 GiB	128 GB	GENERAL	`reserved.x86-64.4cpu.8gib`
Windows	8	16 GiB	128 GB	GENERAL	`reserved.x86-64.8cpu.16gib`
Windows	16	32 GiB	256 GB	GENERAL	`reserved.x86-64.16cpu.32gib`

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
Windows	32	64 GiB	256 GB	GENERAL	reserved.x86-64.32cpu.64gib
Windows	36	72 GiB	256 GB	GENERAL	reserved.x86-64.36cpu.72gib
Windows	48	96 GiB	512 GB	GENERAL	reserved.x86-64.48cpu.96gib
Windows	64	128 GiB	512 GB	GENERAL	reserved.x86-64.64cpu.128gib
Windows	72	144 GiB	824 GB	GENERAL	reserved.x86-64.72cpu.144gib
Windows	96	192 GiB	824 GB	GENERAL	reserved.x86-64.96cpu.192gib
Windows EC2	4	8 GiB	128 GB	GENERAL	reserved.x86-64.4cpu.8gib

Tipo de entorno	CPU virtuales	Memoria	Espacio en disco	Tipo de máquina	Tipo de instancia de computación
Windows EC2	8	16 GiB	128 GB	GENERAL	reserved.x86-64.8cpu.16gib

Para obtener más información sobre el identificador de precios, consulte https://aws.amazon.com/codebuild/pricing/.

Para elegir un tipo de computación:

- En la CodeBuild consola, en la página de configuración de la flota de cómputos, elige una de las opciones entre las CPU virtuales , la memoria y el disco. Para obtener más información, consulte Creación de una flota de capacidad reservada.
- Para AWS CLI, ejecuta el `update-fleet` comando `create-fleet` o, especificando el valor de `computeType` to`ATTRIBUTE_BASED_COMPUTE`. Para obtener más información, consulte create-fleet o update-fleet.
- En el caso de AWS los SDK, llame al equivalente de la `UpdateFleet` operación `CreateFleet` o correspondiente al lenguaje de programación de destino y especifique el valor de `computeType` to`ATTRIBUTE_BASED_COMPUTE`. Para obtener más información, consulte la AWS Referencia de SDK y herramientas.

> **Note**
>
> En el caso de AWS los SDK AWS CLI y, puedes seguir usando `computeType` entradas como`BUILD_GENERAL1_SMALL`, por ejemplo, para elegir los tipos de procesamiento en lugar de. `ATTRIBUTE_BASED_COMPUTE` Para obtener más información, consulte Acerca de los tipos de entorno bajo demanda.

Familias de instancias admitidas

AWS CodeBuild admite las siguientes instancias para flotas de capacidad reservada:

- Uso general: M5 | M5a | M5ad | M5d | M5dn | M5n | M5zn | M6a | M6g | M6gd | M6i | M6id | M6idn | M6in | M7a | M7g | M7gd | M7i | M8g | T3 | T3a | T4g M7i-flex
- Optimizado para computación: C5 | C5a | C5ad | C5d | C5n | C6a | C6g | C6gd | C6gn | C6i | C6id | C6in | C7a | C7g | C7gd | C7gn | C7i C7i-flex | C8g
- Memoria optimizada: R5 | R5a | R5ad | R5b | R5dn | R5n | R6a | R6g | R6g | R6gd | R6i | R6iDn | R6id | R7a | R7g | R7gd | R7i | R7iz | R8g | | | | | | X1 | X1e | X2d | X2idn | X2iEen | X2iEN | X8g | zd1 U-3tb1 U-6tb1 U-9tb1 U-12tb1 U-18tb1 U-24tb1 U7i-6tb U7i-8tb U7i-12tb U7in-16tb U7in-24tb U7in-32tb
- Optimizadas para el almacenamiento: D3 | D3en | I3 | I3en | I4g | I4i | I7ie | I8g | Im4gn | Is4gen
- Computación acelerada: DL1 | DL2q | F1 | F2 | G4ad | G4dn | G5 | G5g | G6 | G6e | Gr6 | Inf1 | Inf2 | P3 | P3dn | P4d | P5 | P5e | P5en | Trn1 | Trn1n | Trn2 | VT1
- High-performance informática: HPC6a | HPC6id | HPC7a | HPC7g
- Generación anterior: A1

Cómo crear una flota de capacidad reservada con un tipo de instancia específico:

- En la CodeBuild consola, en la página de configuración de la flota de cómputos, diríjase a la sección Configuración de la capacidad. En Modo de selección de computación, seleccione Entrada manual y en Tipo de instancia de computación, elija uno de los tipos de instancia del menú desplegable. Para obtener más información, consulte Creación de una flota de capacidad reservada.
- Para ello AWS CLI, ejecuta el `update-fleet` comando `create-fleet` o, especificando el valor de `computeType` para `CUSTOM_INSTANCE_TYPE` y el `ComputeConfiguration` `instanceType` para el tipo de instancia especificado. Para obtener más información, consulte create-fleet o update-fleet.
- En el caso de AWS los SDK, llame al equivalente de la `UpdateFleet` operación `CreateFleet` u para su lenguaje de programación de destino, especificando el valor de `computeType` to `CUSTOM_INSTANCE_TYPE` y el del ComputeConfiguration `instanceType` tipo de instancia especificado. Para obtener más información, consulte la AWS Referencia de SDK y herramientas.

Acerca de los tipos de entorno bajo demanda

AWS CodeBuild proporciona entornos de construcción con la siguiente memoria, vCPU y espacio en disco disponibles para el modo de procesamiento EC2:

Tipo de computación	Valor de computeType del entorno	Valor de tipo del entorno	Memoria	CPU virtuales	Espacio en disco
ARM Small [1]	`BUILD_GENERAL1_SMALL`	`ARM_CONTAINER` `ARM_EC2`	4 GiB	2	64 GB
ARM Medium [1]	`BUILD_GENERAL1_MEDIUM`	`ARM_CONTAINER` `ARM_EC2`	8 GiB	4	128 GB
ARM Large [1]	`BUILD_GENERAL1_LARGE`	`ARM_CONTAINER` `ARM_EC2`	16 GiB	8	128 GB
ARM XLarge [1]	`BUILD_GENERAL1_XLARGE`	`ARM_CONTAINER`	64 GiB	32	256 GB
ARM 2XLarge [1]	`BUILD_GENERAL1_2XLARGE`	`ARM_CONTAINER`	96 GiB	48	824 GB
Linux Small [1]	`BUILD_GENERAL1_SMALL`	`LINUX_CONTAINER` `LINUX_EC2`	4 GiB	2	64 GB
Linux Medium [1]	`BUILD_GENERAL1_MEDIUM`	`LINUX_CONTAINER` `LINUX_EC2`	8 GiB	4	128 GB
Linux Large [1]	`BUILD_GENERAL1_LARGE`	`LINUX_CONTAINER`	16 GiB	8	128 GB

Tipo de computación	Valor de computeType del entorno	Valor de tipo del entorno	Memoria	CPU virtuales	Espacio en disco
		LINUX_EC2			
Linux XLarge [1]	BUILD_GENERAL1_XLARGE	LINUX_CONTAINER	72 GiB	36	256 GB
Linux 2XLarge	BUILD_GENERAL1_2XLARGE	LINUX_CONTAINER	144 GiB	72	824 GB (SSD)
Linux GPU Sma	BUILD_GENERAL1_SMALL	LINUX_GPU_CONTAINER	16 GiB	4	235 GB (SSD)
Linux GPU Larg	BUILD_GENERAL1_LARGE	LINUX_GPU_CONTAINER	255 GiB	32	50 GB
Windows Mediu [1]	BUILD_GENERAL1_MEDIUM	WINDOWS_SERVER_2019_CONTAINER	8 GiB	4	128 GB
		WINDOWS_SERVER_2022_CONTAINER			
		WINDOWS_EC2			

Tipo de computación	Valor de computeType del entorno	Valor de tipo del entorno	Memoria	CPU virtuales	Espacio en disco

Windows Large [1]	`BUILD_GENERAL1_LARGE`	`WINDOWS_SERVER_2019_CONTAINER`
		`WINDOWS_SERVER_2022_CONTAINER`
		`WINDOWS_EC2`
			16 GiB	8	128 GB

Windows XLarg [1]	`BUILD_GENERAL1_XLARGE`	`WINDOWS_SERVER_2022_CONTAINER`	72 GiB	36	256 GB
Windows 2XLar [1]	`BUILD_GENERAL1_2XLARGE`	`WINDOWS_SERVER_2022_CONTAINER`	144 GiB	72	824 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/amazonlinux-x86_64-standard:5.0` en lugar de una versión más detallada, como `aws/codebuild/amazonlinux-x86_64-standard:5.0-1.0.0`.

AWS CodeBuild proporciona a los entornos de construcción 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 GiB	10 GB
ARM Lambda 2GB	BUILD_LAMBDA_2GB	ARM_LAMBDA_CONTAINER	2 GiB	10 GB
ARM Lambda 4GB	BUILD_LAMBDA_4GB	ARM_LAMBDA_CONTAINER	4 GiB	10 GB
ARM Lambda 8GB	BUILD_LAMBDA_8GB	ARM_LAMBDA_CONTAINER	8 GiB	10 GB
ARM Lambda de 10 GB	BUILD_LAMBDA_10GB	ARM_LAMBDA_CONTAINER	10 GiB	10 GB
Linux Lambda 1GB	BUILD_LAMBDA_1GB	LINUX_LAMBDA_CONTAINER	1 GiB	10 GB
Linux Lambda 2GB	BUILD_LAMBDA_2GB	LINUX_LAMBDA_CONTAINER	2 GiB	10 GB
Linux Lambda 4GB	BUILD_LAMBDA_4GB	LINUX_LAMBDA_CONTAINER	4 GiB	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 GiB	10 GB
Linux Lambda 10G	`BUILD_LAMBDA_10GB`	`LINUX_LAMBDA_CONTAINER`	10 GiB	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 para crear un proyecto de compilación o en la página Editar un proyecto de creación, 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 AWS CLI, ejecuta el `update-project` comando `create-project` o, especificando el `computeType` valor del `environment` objeto. Para obtener más información, consulte Cree un proyecto de compilación (AWS CLI) o Cambia la configuración de un proyecto de compilación (AWS CLI).
- En el AWS caso de los SDK, llame al equivalente de la `UpdateProject` operación `CreateProject` o correspondiente al lenguaje de programación de destino y especifique el equivalente del `computeType` valor del `environment` objeto. Para obtener más información, consulte la AWS Referencia de SDK y herramientas.

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:
  - Este de EE. UU. (Ohio)
  - Este de EE. UU. (Norte de Virginia)
  - Oeste de EE. UU. (Oregón)
  - Asia-Pacífico (Seúl)
  - Asia-Pacífico (Sídney)
  - Asia-Pacífico (Tokio)
  - Canadá (centro)
  - China (Pekín)
  - China (Ningxia)
  - Europa (Fráncfort)
  - Europa (Irlanda)
  - Europa (Londres)
- El tipo de computación `BUILD_GENERAL1_2XLARGE` solo está disponible en las regiones siguientes:
  - Este de EE. UU. (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)

- Asia-Pacífico (Yakarta)
- Asia-Pacífico (Melbourne)
- Asia-Pacífico (Mumbai)
- 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)
- Middle East (Bahrain)
- Medio Oriente (EAU)
- América del Sur (São Paulo)

- El tipo de entorno `ARM_CONTAINER` solo está disponible en las regiones siguientes:
  - Este de EE. UU. (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 (Malasia)

- Asia-Pacífico (Mumbai)
- Asia-Pacífico (Nueva Zelanda)
- Asia-Pacífico (Osaka)
- Asia-Pacífico (Seúl)
- Asia-Pacífico (Singapur)
- Asia-Pacífico (Sídney)
- Asia-Pacífico (Tailandia)
- 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)
- Middle East (Bahrain)
- Medio Oriente (EAU)
- América del Sur (São Paulo)

- El tipo de entorno `WINDOWS_SERVER_2022_CONTAINER` solo está disponible en las regiones siguientes:
  - Este de EE. UU. (Ohio)
  - Este de EE. UU. (Norte de Virginia)
  - Oeste de EE. UU. (Oregón)
  - Asia-Pacífico (Sídney)
  - Asia-Pacífico (Tokio)
  - Europa (Fráncfort)

  - Europa (Irlanda)
  - América del Sur (São Paulo)
- El tipo de entorno `LINUX_EC2` (`BUILD_GENERAL1_SMALL`, `BUILD_GENERAL1_MEDIUM`, `BUILD_GENERAL1_LARGE`) solo está disponible en las regiones siguientes:
  - Este de EE. UU. (Ohio)
  - Este de EE. UU. (Norte de Virginia)
  - Oeste de EE. UU. (Norte de California)
  - Oeste de EE. UU. (Oregón)
  - África (Ciudad del Cabo)
  - Asia-Pacífico (Hong Kong)
  - Asia-Pacífico (Yakarta)
  - Asia-Pacífico (Melbourne)
  - Asia-Pacífico (Hyderabad)
  - Asia-Pacífico (Malasia)
  - Asia-Pacífico (Mumbai)
  - Asia-Pacífico (Nueva Zelanda)
  - Asia-Pacífico (Osaka)
  - Asia-Pacífico (Seúl)
  - Asia-Pacífico (Singapur)
  - Asia-Pacífico (Sídney)
  - Asia-Pacífico (Tailandia)
  - Asia-Pacífico (Tokio)
  - Canadá (centro)
  - China (Pekín)
  - China (Ningxia)
  - Europa (Fráncfort)
  - Europa (Irlanda)
  - Europa (Londres)
  - Europa (Milán)
  - Europa (París)

- Europa (España)
- Europa (Estocolmo)
- Europa (Zúrich)
- Israel (Tel Aviv)
- Middle East (Bahrain)
- Medio Oriente (EAU)
- América del Sur (São Paulo)
- AWS GovCloud (US-West)
- AWS GovCloud (US-East)

- El tipo de entorno `ARM_EC2` (`BUILD_GENERAL1_SMALL`, `BUILD_GENERAL1_MEDIUM`, `BUILD_GENERAL1_LARGE`) solo está disponible en las regiones siguientes:
  - Este de EE. UU. (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 (Malasia)
  - Asia-Pacífico (Mumbai)
  - Asia-Pacífico (Nueva Zelanda)
  - Asia-Pacífico (Osaka)
  - Asia-Pacífico (Seúl)
  - Asia-Pacífico (Tailandia)
  - 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)
- Europa (Zúrich)
- Israel (Tel Aviv)
- Middle East (Bahrain)
- América del Sur (São Paulo)
- AWS GovCloud (US-West)
- AWS GovCloud (US-East)

- El tipo de entorno `WINDOWS_EC2` (`BUILD_GENERAL1_MEDIUM`, `BUILD_GENERAL1_LARGE`) solo está disponible en las regiones siguientes:
  - Este de EE. UU. (Ohio)
  - Este de EE. UU. (Norte de Virginia)
  - Oeste de EE. UU. (Oregón)
  - Asia-Pacífico (Sídney)
  - Asia-Pacífico (Tokio)
  - Europa (Fráncfort)
  - Europa (Irlanda)
  - América del Sur (São Paulo)
- El modo de procesamiento AWS Lambda (`ARM_LAMBDA_CONTAINER`y`LINUX_LAMBDA_CONTAINER`) solo está disponible en estas regiones:
  - Este de EE. UU. (Norte de Virginia)
  - Este de EE. UU. (Ohio)
  - Oeste de EE. UU. (Norte de California)
  - Oeste de EE. UU. (Oregón)
  - Asia-Pacífico (Mumbai)
  - Asia-Pacífico (Singapur)

  - Asia-Pacífico (Sídney)
  - Asia-Pacífico (Tokio)
  - Europa (Fráncfort)
  - Europa (Irlanda)
  - América del Sur (São Paulo)
- El modo de computación `MAC_ARM` solo está disponible en las regiones siguientes:
  - Este de EE. UU. (Norte de Virginia)
  - Este de EE. UU. (Ohio)
  - Oeste de EE. UU. (Oregón)
  - Asia-Pacífico (Sídney)
  - Europa (Fráncfort)

Para el tipo de proceso `BUILD_GENERAL1_2XLARGE`, se admiten las imágenes de Docker de hasta 100 GB sin comprimir.

> **Note**
>
> En el caso de imágenes de entornos de compilación personalizadas, 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 sin VPC. Si quiere usar contenedores de Docker para las compilaciones de VPC, consulte Runtime Privilege and Linux Capabilities en el sitio web de Docker Docs y habilite el modo privilegiado. Además, Windows no admite el modo privilegiado.

Intérpretes de comandos y comandos de los entornos de compilación

Proporcionas un conjunto de comandos AWS CodeBuild para que se ejecuten en un entorno de compilación durante el ciclo de vida de una compilación (por ejemplo, instalando dependencias de compilación y probando y compilando el 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 la Referencia de especificación de compilación para CodeBuild.
- Usa la CodeBuild consola 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 CodeBuild consola 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).
- Usa los AWS CLI AWS SDK para crear un proyecto de compilación o 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 Cambio de la configuración del proyecto de compilación.
- Usa los AWS CLI AWS SDK para iniciar una compilación, especificando un archivo de especificaciones de construcción o una sola cadena que incluya el contenido de un archivo de especificaciones de compilación equivalente. Para obtener más información, consulte la descripción del valor `buildspecOverride` en Ejecución de compilaciones de forma manual.

Puede especificar cualquier comando de Lenguaje de comandos de intérprete de comandos (sh). En la versión 0.1 de buildspec, CodeBuild ejecuta cada comando de Shell 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 se produce un error, es posible que sea más difícil solucionar el error en comparación con ejecutar un único comando en su propia instancia del shell predeterminado.

Los comandos que se ejecutan en una imagen de Windows Server Core utilizan el PowerShell shell.

Variables de entorno en los entornos de compilación

AWS CodeBuild proporciona varias variables de entorno que puedes usar en tus comandos de compilación:

AWS_DEFAULT_REGION

La AWS región en la que se ejecuta la compilación (por ejemplo,`us-east-1`). Esta variable de entorno se utiliza principalmente en la AWS CLI.

AWS_REGION

La AWS región en la que se ejecuta la compilación (por ejemplo,`us-east-1`). Esta variable de entorno la utilizan principalmente los AWS SDK.

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

El CodeBuild ID de la compilación (por ejemplo,`codebuild-demo-project:b1e6661e-e4f2-4156-9ab9-82a19EXAMPLE`).

CODEBUILD_BUILD_IMAGE

El identificador de la imagen de CodeBuild compilación (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 se CodePipeline inició la compilación, este es el nombre de la canalización (por ejemplo,`codepipeline/my-demo-pipeline`). Si un usuario inició la compilación, este es el nombre del usuario (por ejemplo,`MyUserName`). Si el plugin de Jenkins CodeBuild inició la compilación, esta es la cadena`CodeBuild-Jenkins-Plugin`.

CODEBUILD_KMS_KEY_ID

El identificador de la AWS KMS clave que CodeBuild se 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_PROJECT_ARN

Nombre de recurso de Amazon (ARN) del proyecto (por ejemplo, `arn:aws:codebuild:region-ID:account-ID:project/project-name`).

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 Obtención de direcciones URL de proyectos de compilación públicos.

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 origen 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 habilitado el control de versiones, esta variable de entorno no está configurada.

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. En el CodeCommit caso de y GitHub, esta es la URL de clonación 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.
- CodeCommitEn efecto, es el identificador de confirmación o el nombre de la rama asociado a la versión del código fuente que se va a compilar.

- GitHubEn el caso de GitHub Enterprise Server y Bitbucket, es el ID de confirmación, el nombre de la sucursal o el nombre de la etiqueta asociados a la versión del código fuente que se va a crear.

  > **Note**
  >
  > En el caso de una GitHub compilación de GitHub Enterprise Server que se desencadena por un evento de solicitud de extracción de un webhook, lo 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

La ruta del directorio que se CodeBuild usa 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 a partir de la carga útil enviada CodeBuild por GitHub GitHub Enterprise Server o Bitbucket. 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`.

AWS CodeBuild también admite un conjunto de variables de entorno para compilaciones de ejecutores autohospedados. Para obtener más información sobre los CodeBuild ejecutores autohospedados, consulte. Tutorial: Configurar un ejecutor de CodeBuild-hosted GitHub acciones

CODEBUILD_RUNNER_OWNER

El propietario del repositorio que desencadena la compilación del ejecutor autoalojado.

CODEBUILD_RUNNER_REPO

El nombre del repositorio que desencadena la compilación del ejecutor autoalojado.

CODEBUILD_RUNNER_REPO_DOMAIN

El dominio del repositorio que desencadena la compilación del ejecutor autoalojado. Solo compilaciones GitHub empresariales especificadas.

CODEBUILD_WEBHOOK_LABEL

La etiqueta que se utiliza para configurar las anulaciones de compilación y el ejecutor autoalojado durante la compilación.

CODEBUILD_WEBHOOK_RUN_ID

El ID de ejecución del flujo de trabajo asociado a la compilación.

CODEBUILD_WEBHOOK_JOB_ID

El ID del trabajo asociado a la compilación.

CODEBUILD_WEBHOOK_WORKFLOW_NAME

El nombre del flujo de trabajo asociado a la compilación, si existe en la carga útil de solicitud de webhook.

CODEBUILD_RUNNER_WITH_BUILDSPEC

Si se configura una anulación de especificación de compilación en las etiquetas de solicitud del ejecutor autoalojado, se establece en `true`.

También puede proporcionar entornos de compilación con sus propias variables de entorno. Para obtener más información, consulte los temas siguientes:

- Úselo CodeBuild con CodePipeline
- Creación de un proyecto de compilación
- Cambio de la configuración del proyecto de compilación
- Ejecución de compilaciones de forma manual
- Referencia de la especificación de compilación

Para enumerar todas las variables de entorno disponibles en un entorno de compilación, puedes ejecutar el `printenv` comando (para el entorno de Linux-based compilación) o `"Get-ChildItem`

`Env:"` (para los entornos de Windows-based compilación) durante la compilación. A excepción de las enumeradas anteriormente, las variables de entorno que comienzan con `CODEBUILD_` son para uso CodeBuild interno. No deben utilizarse en los comandos de compilación.

> **Important**
>
> No recomendamos encarecidamente el uso de variables de entorno para almacenar valores confidenciales, especialmente los ID de las claves de AWS acceso. Las variables de entorno se pueden mostrar en texto plano mediante herramientas como la CodeBuild consola 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.

CODEBUILD_BUILD_URL

La URL de los resultados de compilación para esta compilación.

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)
```

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
- Cambie la configuración del proyecto de compilación en AWS CodeBuild
- Tokens de acceso múltiple en CodeBuild
- Eliminar proyectos de construcción en AWS CodeBuild
- Obtención de direcciones URL de proyectos de compilación públicos
- Uso compartido de proyectos de compilación
- Etiquetado de proyectos de compilación
- Usa corredores con AWS CodeBuild
- Usa webhooks con AWS CodeBuild
- Vea los detalles de un proyecto de construcción en AWS CodeBuild
- Ver los nombres de los proyectos de construcción en AWS CodeBuild

Creación de un proyecto de compilación en AWS CodeBuild

Puede usar la AWS CodeBuild consola o AWS CLI los AWS SDK para crear un proyecto de compilación.

Temas

- Requisitos previos
- Creación de un proyecto de compilación (consola)
- Cree un proyecto de compilación (AWS CLI)
- Crea un proyecto de compilación (AWS SDK)
- Cree un proyecto de compilación (CloudFormation)

Requisitos previos

Antes de crear un proyecto de compilación, responda las preguntas de Planificación de una compilación.

Creación de un proyecto de compilación (consola)

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

Si se muestra 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 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 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

Credencial

Elija Credencial de origen predeterminada o Credencial de origen personalizada y siga las instrucciones para administrar la credencial de origen predeterminada o personalizar la credencial de origen.

Tipo de conexión

Elige OAuth AWS CodeConnections, contraseña de aplicación o token de acceso personal al que deseas conectarte. CodeBuild

Conexión

Seleccione una conexión de Bitbucket o un secreto de Secrets Manager para conectarse mediante el tipo de conexión especificado.

Repositorio

Elija Repositorio en mi cuenta de Bitbucket o Repositorio público y especifique la URL del repositorio.

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 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 introduzca 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

Credencial

Elija Credencial de origen predeterminada o Credencial de origen personalizada y siga las instrucciones para administrar la credencial de origen predeterminada o personalizar la credencial de origen.

Tipo de conexión

Elige el token de acceso personal, de OAuth o de GitHub aplicación al que deseas conectarte. CodeBuild

Conexión

Selecciona una GitHub conexión o un secreto de Secrets Manager para conectarte a través del tipo de conexión especificado.

Repositorio

Seleccione Repositorio en mi GitHub cuenta, Repositorio público o webhook GitHub con alcance e introduzca la URL del repositorio.

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 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 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.

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

Credencial

Elija Credencial de origen predeterminada o Credencial de origen personalizada y siga las instrucciones para administrar la credencial de origen predeterminada o personalizar la credencial de origen.

Tipo de conexión

Elige un AWS CodeConnectionstoken de acceso personal al que deseas conectarte CodeBuild.

Conexión

Seleccione una conexión GitHub empresarial o un secreto de Secrets Manager para conectarse a través del tipo de conexión especificado.

Repositorio

Elija Repositorio en mi cuenta GitHub GitHub empresarial o webhook de ámbito empresarial e introduzca la URL del repositorio.

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 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 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

Credencial

Elija Credencial de origen predeterminada o Credencial de origen personalizada y siga las instrucciones para administrar la credencial de origen predeterminada o personalizar la credencial de origen.

Tipo de conexión

AWS CodeConnectionsse utiliza para conectarse GitLab a. CodeBuild

Conexión

Seleccione una GitLab conexión a través de la cual conectarse AWS CodeConnections.

Repositorio

Seleccione el repositorio que desea utilizar.

Versión de fuente

Introduzca un ID de solicitud de extracción, una ramificación, un ID de confirmación, un etiqueta o referencia y un ID de confirmación. Para obtener más información, consulte Ejemplo de 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

Credencial

Elija Credencial de origen predeterminada o Credencial de origen personalizada y siga las instrucciones para administrar la credencial de origen predeterminada o personalizar la credencial de origen.

Tipo de conexión

AWS CodeConnectionsse utiliza para conectarse a GitLab Self Managed CodeBuild.

Conexión

Seleccione una conexión GitLab autogestionada a través de la cual conectarse AWS CodeConnections.

Repositorio

Seleccione el repositorio que desea utilizar.

Versión de fuente

Introduzca un ID de solicitud de extracción, una ramificación, un ID de confirmación, un etiqueta o referencia y un ID de confirmación. Para obtener más información, consulte Ejemplo de 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:

- Para usar flotas bajo demanda gestionadas por AWS CodeBuild, elija On-demand. Con flotas bajo demanda, CodeBuild proporciona computación para sus compilaciones. Las máquinas se destruyen cuando finaliza la construcción. On-demand las flotas están totalmente gestionadas e incluyen capacidades de escalado automático para hacer frente a los picos de demanda.
- Para utilizar 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 la construcción. Con flotas de capacidad reservada, sus máquinas están siempre en funcionamiento y seguirán incurriendo en costes mientras estén aprovisionadas.

Para obtener información, consulte Ejecución de compilaciones en flotas de capacidad reservada.

Imagen del entorno

Realice una de las siguientes acciones:

- 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:

- Para usar la computación de EC2, elija EC2. La computación de EC2 ofrece flexibilidad optimizada durante las ejecuciones de acciones.
- Para usar la computación de Lambda, elija Lambda. La computación de 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 Ejecutar se basa en AWS Lambda cálculo.

Rol de servicio

Realice una de las siguientes acciones:

- 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

Auto-retry limit

Especifique el número de reintentos automáticos adicionales tras una compilación con errores. Por ejemplo, si el límite de reintentos automáticos está establecido en 2, CodeBuild llamará a la `RetryBuild` API para volver a intentar automáticamente la compilación hasta 2 veces más.

Timeout (Tiempo de espera)

Especifica un valor, entre 5 minutos y 36 horas, tras el cual CodeBuild se detendrá la compilación si no se ha completado. Si se dejan los campos de horas y minutos en blanco, se usará el valor predeterminado de 60 minutos.

Privilegiado

(Opcional) Seleccione Habilite esta marca si desea compilar imágenes de Docker o desea que sus compilaciones obtengan privilegios elevados solo si pretende usar este proyecto de compilación para compilar 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 consiste en inicializar el daemon de Docker en la `install` fase de la especificación de compilación mediante la ejecución de los siguientes comandos de compilación. No ejecute estos comandos si ha elegido una imagen de entorno de compilación proporcionada con el soporte de Docker. CodeBuild

> **Note**
>
> De forma predeterminada, el daemon de Docker está habilitado para compilaciones sin VPC. Si quiere usar contenedores de Docker para las compilaciones de VPC, consulte Runtime Privilege and Linux Capabilities en el sitio web de Docker Docs y habilite 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.

Credencial de registro

Especifique una credencial de registro cuando el proyecto esté configurado con una imagen de registro no privada.

> **Note**
>
> Esta credencial solo se utilizará si las imágenes se sustituyen por las de registros privados.

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 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 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:

- 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 denominado `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 la 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 Ejecución de compilaciones en lotes.

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 debes poder llamar al `StartBuildStopBuild`, y a `RetryBuild` las 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 computación permitidos para lote

Seleccione los tipos de computación permitidos para el lote. Seleccione todos los que correspondan.

Flotas permitidas para lote

Seleccione las flotas permitidas 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:

- 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. Este es el valor predeterminado.
- 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, 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 de las compilaciones en caché para mejorar el rendimiento.

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 logs

Seleccione registros de CloudWatch .

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.

Cree un proyecto de compilación (AWS CLI)

Para obtener más información sobre el uso del AWS CLI with CodeBuild, consulte laReferencia de la línea de comandos.

Para crear un proyecto de CodeBuild construcción utilizando el AWS CLI, debe crear una estructura de JSON-formatted proyecto, rellenar la estructura y ejecutar el `create-project`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>",
  "autoRetryLimit": <auto-retry-limit>,
  "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>"
      ],
      "fleetsAllowed": [
        "<fleet-name>"
      ]
    },
    "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 AWS cuenta.

description

Opcional. La descripción de este proyecto de compilación.

origen

Obligatorio. Un ProjectSourceobjeto 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, debes usar el `buildspec` atributo para especificar una YAML-formatted cadena para tu especificación de compilación. Para obtener más información, consulte Creación de un proyecto de compilación sin un origen.

source/location

Obligatorio a menos que lo establezcas. *<source-type>* `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>*`/`*<subfoler-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. GitHub Para ello, utilice la CodeBuild consola para crear un proyecto de compilación.
  - Elija Authorize application. (Una vez que se haya conectado a su GitHub cuenta, 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 consola de CodeBuild para crear un proyecto de compilación.
  - En el panel de navegación de Herramientas para desarrolladores, elija Configuración, Conexiones y, después, 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 para`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.

fuente/ git CloneDepth

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 compilación. 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

Contiene información sobre la configuración de autorización para acceder CodeBuild al código fuente que se va a compilar.

source/auth/tipo

Obligatorio. El tipo de autorización que se usará. Los valores válidos son los siguientes:

- `OAUTH`
- `CODECONNECTIONS`
- `SECRETS_MANAGER`

source/auth/recurso

Opcional. El valor del recurso que se aplica al tipo de autorización especificado. Puede ser el ARN de Secrets Manager o el ARN. AWS CodeConnections

fuente/ informe BuildStatus

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.

fuente/ compilación StatusConfig

Contiene información que define cómo 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`.

source/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
```

source/buildStatusConfig/URL de destino

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 del 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/ git SubmodulesConfig

Opcional. Información acerca de la configuración de submódulos de Git. Se utiliza CodeCommit únicamente GitHub con GitHub Enterprise Server y Bitbucket.

source/gitSubmodulesConfig/FetchSubmódulos

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/sourceIdentifier

Identificador definido por el usuario para la fuente del proyecto. Opcional para la fuente principal. Obligatorio para las fuentes secundarias.

secondarySources

Opcional. Conjunto de ProjectSourceobjetos 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 . En un objeto de fuente secundario, `sourceIdentifier` es obligatorio.

secundario SourceVersions

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 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 versión fuente con AWS CodeBuild.

artifacts

Obligatorio. Un ProjectArtifactsobjeto 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 los siguientes:

- `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.

La ruta del depósito de salida en el que se va a colocar el archivo ZIP o la carpeta. Si no especifica un valor`path`, CodeBuild utiliza `namespaceType` (si se ha especificado) y `name` para

determinar la ruta y el nombre del archivo o carpeta 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 del archivo ZIP o carpeta 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/ anulación ArtifactName

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. Matriz de ProjectArtifactsobjetos que contiene 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. ProjectCacheObjeto que contiene información sobre la configuración de caché de este proyecto de compilación. Para obtener más información, consulte Almacenamiento de las compilaciones en caché.

entorno

Obligatorio. Un ProjectEnvironmentobjeto 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. Típicamente, este identificador se expresa como*`image-name`*:*`tag`*. 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 ruta y la clave de objeto que contiene el certificado. PEM-encoded La clave de objeto puede ser solo el archivo.pem o un archivo.zip que contenga el certificado. PEM-encoded 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 EnvironmentVariableobjetos que contiene las variables de entorno que desea 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. Pues`name`, 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. Establece `type` en `PARAMETER_STORE`. Utilizando un parámetro denominado `/CodeBuild/dockerLoginPassword` como ejemplo, establezca `name` con `LOGIN_PASSWORD`. Establece `value` en `/CodeBuild/dockerLoginPassword`. Establece `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 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 usa Secrets Manager, para `value`, defina el nombre del parámetro tal como está almacenado en Secrets Manager. Establece `type` en `SECRETS_MANAGER`. Utilizando un secreto denominado `/CodeBuild/dockerLoginPassword` como ejemplo, establezca `name` con `LOGIN_PASSWORD`. Establece `value` en `/CodeBuild/dockerLoginPassword`. Establece `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 RegistryCredentialobjeto que especifica las credenciales que proporcionan acceso a un registro de Docker privado.

environment/registryCredential/credencial

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/Proveedor de credenciales

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.

environment/imagePullCredentialsType

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 del entorno de compilación proporcionada por CodeBuild compatible con Docker.

> **Note**
>
> De forma predeterminada, el daemon de Docker está habilitado para compilaciones sin VPC. Si quiere usar contenedores de Docker para las compilaciones de VPC, consulte Runtime Privilege and Linux Capabilities en el sitio web de Docker Docs y habilite 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`*).

auto RetryLimit

Opcional. El número de reintentos automáticos adicionales tras una compilación con errores. Por ejemplo, si el límite de reintentos automáticos está establecido en 2, CodeBuild llamará a la `RetryBuild` API para volver a intentar automáticamente la compilación hasta 2 veces más.

tiempo de espera InMinutes

Opcional. El número de minutos, entre 5 y 2160 (36 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`.

en cola TimeoutInMinutes

Opcional. El número de minutos, entre 5 y 480 (8 horas), tras los cuales 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 VpcConfigobjeto 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:

vpcId

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>
```

seguridad GroupIds

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 Ejemplo de creación de insignias con CodeBuild.

logsConfig

Un LogsConfigobjeto que contiene información sobre la ubicación de los registros de esta compilación.

LogsConfig/ cloud WatchLogs

CloudWatchLogsConfigObjeto que contiene información sobre cómo enviar registros a Logs. CloudWatch

logsConfig/s3Logs

S3LogsConfigObjeto que contiene información sobre cómo enviar registros a Amazon S3.

archivo SystemLocations

Opcional. Conjunto de ProjectFileSystemsLocationobjetos que contienen información sobre la configuración de Amazon EFS.

construir BatchConfig

Opcional. El `buildBatchConfig` objeto es una ProjectBuildBatchConfigestructura que contiene la información de configuración de construcción por lotes del proyecto.

buildBatchConfig/ServiceRole

ARN del rol de servicio para el proyecto de compilación por lotes.

build/CombineArtifacts BatchConfig

Valor booleano que especifica si se deben combinar los artefactos de compilación de la compilación por lotes en una única ubicación de artefactos.

buildBatchConfig/restrictions/máximo BuildsAllowed

Número máximo de compilaciones permitidas.

buildBatchConfig/restrictions/computar TypesAllowed

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/restrictions/Flotas permitidas

Matriz de cadenas que especifican las flotas que se permiten para la compilación por lotes. Consulte Ejecución de compilaciones en flotas de capacidad reservada para obtener más información.

compilar/tiempo de espera BatchConfig InMinutes

Cantidad máxima de tiempo, en minutos, en que se debe completar la compilación por lotes.

compilarBatchConfig/lote ReportMode

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.

concurrente BuildLimit

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 Cambia la configuración de un proyecto de compilación (AWS CLI).

Para empezar a ejecutar una compilación, consulte Ejecute 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, consulteIniciar la ejecución de compilaciones automáticamente (AWS CLI).

Crea un proyecto de compilación (AWS SDK)

Para obtener información sobre su uso AWS CodeBuild con los AWS SDK, consulte la. AWS Referencia de SDK y herramientas

Cree un proyecto de compilación (CloudFormation)

Para obtener información sobre su uso AWS CodeBuild con CloudFormation, consulte la CloudFormation plantilla CodeBuild en la Guía del AWS CloudFormation usuario.

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 usar 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 Consola de administración de AWS y abra la CodeBuild consola 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, selecciona Básico si quieres que solo se EventBridge incluya en la notificación la información proporcionada a Amazon. Selecciona Completa si deseas incluir la

información proporcionada a Amazon EventBridge y la información que pueda proporcionar el administrador de notificaciones CodeBuild o el administrador de notificaciones.

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 Amazon Q Developer en aplicaciones de chat (Slack) o Tema de SNS. En Elegir destino, elija el nombre del cliente (para un cliente de Slack configurado en Amazon Q Developer en aplicaciones de chat) 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 Amazon Q Developer en aplicaciones de chat durante la creación de una regla de notificación. Si elige Amazon Q Developer en aplicaciones de chat (Slack), aparecerá un botón que lo llevará a configurar un cliente en Amazon Q Developer en aplicaciones de chat. Elija esa opción para abrir la consola de Amazon Q Developer en aplicaciones de chat. Para obtener más información, consulte Configuración de la integración entre las notificaciones y Amazon Q Developer en aplicaciones de chat.
   > - 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 puede configurar la integración entre notificaciones y Amazon Q Developer en aplicaciones de chat para enviar notificaciones a salas de chat de Amazon Chime. Para obtener más información, consulte Configuración de la integración entre notificaciones y Amazon Q Developer en aplicaciones de chat.

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*.

2. 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 siguiente ejemplo, se muestra una regla de notificación con el nombre **MyNotificationRule** de un proyecto de construcción nombrado *MyBuildProject* en AWS una cuenta con el ID*123456789012*. Las notificaciones se envían con todos los detalles a un tema de Amazon SNS denominado *codestar-notifications-MyNotificationTopic* Cuando las compilaciones se realizan correctamente:

```
{
    "Name": "MyNotificationRule",
    "EventTypeIds": [
        "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:codestar-
notifications-MyNotificationTopic"
        }
    ],
    "Status": "ENABLED",
    "DetailType": "FULL"
}
```

Guarde el archivo.

3. 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
```

4. Si se ejecuta correctamente, el comando devuelve el ARN de la regla de notificación, similar a lo siguiente:

```
{
    "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/
dc82df7a-EXAMPLE"
}
```

Cambie la configuración del proyecto de compilación en AWS CodeBuild

Puedes usar la AWS CodeBuild consola o los AWS CLI AWS SDK 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 Permisos de informes de prueba.

Temas

- Cambiar la configuración de un proyecto de compilación (consola)
- Cambia la configuración de un proyecto de compilación (AWS CLI)
- Cambiar la configuración de un proyecto de construcción (AWS SDK)

Cambiar la configuración de un proyecto de compilación (consola)

Para cambiar la configuración de un proyecto de compilación, realice el procedimiento siguiente:

1. Abre 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 creación 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 público 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 Obtención de direcciones URL de proyectos de compilación públicos.

> **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 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 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

Credencial

Elija Credencial de origen predeterminada o Credencial de origen personalizada y siga las instrucciones para administrar la credencial de origen predeterminada o personalizar la credencial de origen.

Tipo de conexión

Elige OAuth AWS CodeConnections, contraseña de aplicación o token de acceso personal al que deseas conectarte. CodeBuild

Conexión

Seleccione una conexión de Bitbucket o un secreto de Secrets Manager para conectarse mediante el tipo de conexión especificado.

Repositorio

Elija Repositorio en mi cuenta de Bitbucket o Repositorio público y especifique la URL del repositorio.

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 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 introduzca 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

Credencial

Elija Credencial de origen predeterminada o Credencial de origen personalizada y siga las instrucciones para administrar la credencial de origen predeterminada o personalizar la credencial de origen.

Tipo de conexión

Elige el token de acceso personal, de OAuth o de GitHub aplicación al que deseas conectarte. CodeBuild

Conexión

Selecciona una GitHub conexión o un secreto de Secrets Manager para conectarte a través del tipo de conexión especificado.

Repositorio

Seleccione Repositorio en mi GitHub cuenta, Repositorio público o webhook GitHub con alcance e introduzca la URL del repositorio.

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 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 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.

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

Credencial

Elija Credencial de origen predeterminada o Credencial de origen personalizada y siga las instrucciones para administrar la credencial de origen predeterminada o personalizar la credencial de origen.

Tipo de conexión

Elige un AWS CodeConnectionstoken de acceso personal al que deseas conectarte CodeBuild.

Conexión

Seleccione una conexión GitHub empresarial o un secreto de Secrets Manager para conectarse a través del tipo de conexión especificado.

Repositorio

Seleccione Repositorio en mi cuenta GitHub GitHub empresarial o webhook de ámbito empresarial e introduzca la URL del repositorio.

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 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 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

Credencial

Elija Credencial de origen predeterminada o Credencial de origen personalizada y siga las instrucciones para administrar la credencial de origen predeterminada o personalizar la credencial de origen.

Tipo de conexión

AWS CodeConnectionsse utiliza para conectarse GitLab a. CodeBuild

Conexión

Seleccione una GitLab conexión a través de la cual conectarse AWS CodeConnections.

Repositorio

Seleccione el repositorio que desea utilizar.

Versión de fuente

Introduzca un ID de solicitud de extracción, una ramificación, un ID de confirmación, un etiqueta o referencia y un ID de confirmación. Para obtener más información, consulte Ejemplo de 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

Credencial

Elija Credencial de origen predeterminada o Credencial de origen personalizada y siga las instrucciones para administrar la credencial de origen predeterminada o personalizar la credencial de origen.

Tipo de conexión

AWS CodeConnectionsse utiliza para conectarse a GitLab Self Managed CodeBuild.

Conexión

Seleccione una conexión GitLab autogestionada a través de la cual conectarse AWS CodeConnections.

Repositorio

Seleccione el repositorio que desea utilizar.

Versión de fuente

Introduzca un ID de solicitud de extracción, una ramificación, un ID de confirmación, un etiqueta o referencia y un ID de confirmación. Para obtener más información, consulte Ejemplo de 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 gestionadas por AWS CodeBuild, elija On-demand. Con flotas bajo demanda, CodeBuild proporciona computación para sus compilaciones. Las máquinas se destruyen cuando finaliza la construcción. On-demand las flotas están totalmente gestionadas e incluyen capacidades de escalado automático para hacer frente a los picos de demanda.
- Para utilizar 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 la construcción. Con flotas de capacidad reservada, sus máquinas están siempre en funcionamiento y seguirán incurriendo en costes mientras estén aprovisionadas.

Para obtener información, consulte Ejecución de compilaciones en flotas de capacidad reservada.

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 reemplaza las imágenes de Docker personalizadas `ENTRYPOINT`.

Rol de servicio

Realice una de las siguientes acciones:

- 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 36 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 Habilite esta marca si desea compilar imágenes de Docker o desea que sus compilaciones obtengan privilegios elevados solo si pretende usar este proyecto de compilación para compilar 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 consiste en inicializar el daemon de Docker en la `install` fase de la especificación de compilación mediante la ejecución de los siguientes comandos de

compilación. No ejecute estos comandos si ha elegido una imagen de entorno de compilación proporcionada con el soporte de Docker. CodeBuild

> **Note**
>
> De forma predeterminada, el daemon de Docker está habilitado para compilaciones sin VPC. Si quiere usar contenedores de Docker para las compilaciones de VPC, consulte Runtime Privilege and Linux Capabilities en el sitio web de Docker Docs y habilite 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.

Credencial de registro

Especifique una credencial de registro cuando el proyecto esté configurado con una imagen de registro no privada.

> **Note**
>
> Esta credencial solo se utilizará si las imágenes se sustituyen por las de registros privados.

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 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 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:

- 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 denominado `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 la 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 Ejecución de compilaciones en lotes.

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 al `StartBuildStopBuild`, y a `RetryBuild` las 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 computación permitidos para lote

Seleccione los tipos de computación permitidos para el lote. Seleccione todos los que correspondan.

Flotas permitidas para lote

Seleccione las flotas permitidas 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:

- 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:

- 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. 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, intduzca 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, 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 de las compilaciones en caché para mejorar el rendimiento.

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 logs

Seleccione registros de CloudWatch .

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.

Cambia la configuración de un proyecto de compilación (AWS CLI)

Para obtener información sobre el uso del AWS CLI with AWS CodeBuild, consulte laReferencia de la línea de comandos.

Para actualizar un CodeBuild proyecto con el AWS CLI, debe crear un archivo JSON con las propiedades actualizadas y pasar ese archivo al `update-project`comando. 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/amazonlinux-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 *`project-info.json`* archivo contiene una serie de proyectos, por lo que no se puede usar directamente para actualizar un proyecto. Sin embargo, puede copiar las propiedades que desee modificar del *`project-info.json`* archivo y pegarlas en el archivo de actualización como línea base para 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 Cree 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 construcción (AWS SDK)

Para obtener información sobre su uso AWS CodeBuild con los AWS SDK, consulte la. AWS Referencia de SDK y herramientas

Tokens de acceso múltiple en CodeBuild

CodeBuild admite el suministro de tokens de acceso a proveedores externos a partir de sus datos secretos en AWS CodeConnections las conexiones AWS Secrets Manager o a través de ellas. Puedes configurar tu secreto o conexión como la credencial predeterminada para las interacciones con un proveedor externo específico GitHub, como GitHub Enterprise o Bitbucket.

Puede establecer las credenciales de origen en tres niveles distintos:

1. Credenciales de cuenta para todos los proyectos: son las credenciales predeterminadas para todos los proyectos de una cuenta de AWS . Se usarán en un proyecto cuando no se especifique ninguna credencial de origen o proyecto.
2. Credenciales a nivel de fuente para un repositorio específico: es cuando se define un secreto o una AWS CodeConnections conexión de Secrets Manager en una fuente de proyecto. Estas credenciales solo se utilizarán para las operaciones en el repositorio de origen especificado. Esto le permite configurar varios tokens de acceso con distintos ámbitos de permiso en el mismo proyecto y no utilizar las credenciales de cuenta predeterminadas.
3. Credenciales alternativas de proyecto: puede establecer una credencial alternativa de proyecto si usa `NO_SOURCE` como tipo de origen principal y define un secreto o una conexión en este. Úselo cuando tenga varias fuentes en un proyecto pero quiera usar las mismas credenciales para ellas, o cuando no quiera usar las credenciales de nivel de cuenta predeterminadas para su proyecto.

Temas

- Paso 1: Crear un secreto de Secrets Manager o un AWS CodeConnections conexión
- Paso 2: Otorgar acceso al rol de IAM CodeBuild del proyecto a los secretos de Secrets Manager
- Paso 3: Configurar Secrets Manager o AWS CodeConnections tokens
- Opciones de configuración adicionales

Paso 1: Crear un secreto de Secrets Manager o un AWS CodeConnections conexión

Siga las instrucciones siguientes para crear un secreto de Secrets Manager o una AWS CodeConnections conexión:

- Creación y almacenamiento de un token en un secreto de Secrets Manager.
- Cree una conexión a GitHub
- Cree una conexión a GitHub Enterprise Server
- Creación de una conexión a Bitbucket

Paso 2: Otorgar acceso al rol de IAM CodeBuild del proyecto a los secretos de Secrets Manager

> **Note**
>
> Antes de continuar, debes tener acceso al token creado en Secrets Manager o AWS CodeConnections.

Para conceder acceso a la función de IAM del CodeBuild proyecto a Secrets Manager o AWS CodeConnections, debes añadir la siguiente política de IAM.

Para conceder el acceso a la función de CodeBuild IAM del proyecto

1. Cree un rol de IAM para su CodeBuild proyecto siguiendo las instrucciones CodeBuild Permitir interactuar con otros AWS servicios para su CodeBuild proyecto.
2. Realice una de las siguientes acciones:

- Añada la siguiente política de IAM a su función de CodeBuild proyecto para conceder acceso a su secreto.

  JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:iam::*:role/Service*"
            ]
        }
    ]
}
```

  (Opcional) Si utilizas claves gestionadas por el AWS KMS cliente para cifrar un secreto de Secrets Manager, puedes añadir la siguiente declaración de política para conceder el acceso.

  JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": "arn:aws:iam::*:role/Service*",
            "Condition": {
                "StringEquals": {
                    "kms:EncryptionContext:SecretARN":
 "arn:aws:iam::*:role/Service*"
```

```
                }
            }
        }
    ]
}
```

- Añada la siguiente política de IAM a su función de CodeBuild proyecto para conceder acceso a su conexión.

  JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "codeconnections:GetConnectionToken",
                "codeconnections:GetConnection"
            ],
            "Resource": [
                "arn:aws:iam::*:role/Service*"
            ]
        }
    ]
}
```

Paso 3: Configurar Secrets Manager o AWS CodeConnections tokens

Puedes configurar tus credenciales de origen en tres niveles diferentes con Secrets Manager o con AWS CodeConnections tokens.

Configurar Secrets Manager o AWS CodeConnections tokens como credenciales a nivel de cuenta

Puede configurar un secreto o una AWS CodeConnections conexión de Secrets Manager como una credencial a nivel de cuenta y utilizarla en un proyecto.

Consola de administración de AWS

Para configurar una conexión como credencial a nivel de cuenta en el Consola de administración de AWS

1. En el caso del proveedor de código fuente, selecciona Bitbucket o GitHubGitHub Enterprise.
2. En Credencial, realice una de las siguientes acciones:
   - Elija Credencial de origen predeterminada para aplicar la credencial de origen predeterminada de su cuenta a todos los proyectos.
     a. Si no está conectado a su proveedor de origen, elija Administrar la credencial de origen predeterminada.
     b. En Tipo de credencial, elija un tipo de credencial.
     c. Si lo deseas AWS CodeConnections, elige usar una conexión existente o crear una nueva conexión.

        Si ha elegido un tipo de credencial distinto, elija en Servicio el servicio que quiere usar para almacenar el token y haga lo siguiente:
        - Si ha elegido usar Secrets Manager, puede elegir entre usar una conexión de secreto existente o crear un secreto nuevo y elegir Guardar. Para obtener más información sobre cómo crear un secreto nuevo, consulte Creación y almacenamiento de un token en un secreto de Secrets Manager.
        - Si eliges usarla CodeBuild, introduce tu token o tu nombre de usuario y contraseña de la aplicación y selecciona Guardar. Para un token de API de Bitbucket, introduce la dirección de correo electrónico y el token de API de tu cuenta de Atlassian en su lugar.
   - Elija Credencial de origen personalizada a fin de usar una credencial de este tipo para anular la configuración predeterminada de su cuenta.
     a. En Tipo de credencial, elija un tipo de credencial.
     b. En Conexión, elija usar una conexión existente o cree una nueva.

AWS CLI

Para configurar una conexión como una credencial a nivel de cuenta en el AWS CLI

- Abra un terminal (Linux, macOS o Unix) o un símbolo del sistema (Windows). Utilice el AWS CLI para ejecutar el import-source-credentials comando.

  Utilice el comando siguiente para configurar un secreto de Secrets Manager:

  ```
  aws codebuild import-source-credentials \
      --token "<secret-arn>" \
      --server-type <source-provider> \
      --auth-type SECRETS_MANAGER \
      --region <aws-region>
  ```

  Use el siguiente comando para configurar una AWS CodeConnections conexión:

  ```
  aws codebuild import-source-credentials \
      --token "<connection-arn>" \
      --server-type <source-provider> \
      --auth-type CODECONNECTIONS \
      --region <aws-region>
  ```

  Este comando permite importar un token como credenciales de origen predeterminadas de cuenta. Al importar una credencial mediante la ImportSourceCredentialsAPI, CodeBuild usará el token para todas las interacciones con el proveedor de origen, como los webhooks, los informes de estado de compilación y las operaciones de clonación de git, a menos que se haya configurado un conjunto de credenciales más específico en el proyecto.

Ahora puede usar el token en el proyecto de compilación y ejecutarlo. Para obtener más información, consulte Creación de un proyecto de compilación en AWS CodeBuild y Ejecute AWS CodeBuild compilaciones manualmente.

Configuración de varios tokens como credenciales de origen

Para usar los secretos o AWS CodeConnections las conexiones de Secrets Manager como credenciales de nivel fuente, haz referencia directamente al token en el CodeBuild proyecto e inicia una compilación.

Consola de administración de AWS

Para configurar varios tokens como credenciales de nivel fuente en el Consola de administración de AWS

1. En Proveedor de origen, elija GitHub.
2. En Credencial, realice una de las siguientes acciones:
   - Elija Credencial de origen predeterminada para aplicar la credencial de origen predeterminada de su cuenta a todos los proyectos.
     a. Si no está conectado a GitHub, elija Administrar la credencial de origen predeterminada.
     b. En Tipo de credencial, selecciona GitHub Aplicación.
     c. En Conexión, elija usar una conexión existente o cree una nueva.
   - Elija Credencial de origen personalizada a fin de usar una credencial de este tipo para anular la configuración predeterminada de su cuenta.
     a. En Tipo de credencial, elija GitHub Aplicación.
     b. En Conexión, elija usar una conexión existente o cree una nueva.
3. Elija Agregar el origen y repita el proceso de elegir el proveedor de origen y las credenciales.

AWS CLI

Para configurar varios tokens como credenciales de nivel de origen en el AWS CLI

- Abra un terminal (Linux, macOS o Unix) o un símbolo del sistema (Windows). Utilice el AWS CLI para ejecutar el create-project comando.

  Utilice el siguiente comando :

```
aws codebuild create-project --region <aws-region> \
    --name <project-name> \
    --artifacts type=NO_ARTIFACTS \
    --environment "type=LINUX_CONTAINER,
                   computeType=BUILD_GENERAL1_SMALL,
                   image=aws/codebuild/amazonlinux-x86_64-standard:5.0" \
    --service-role <service-role-name> \
    --source "type=GITHUB,
              location=<github-repository-1>,
```

```
                auth={type=SECRETS_MANAGER,resource=<secret-or-connection-arn-1>}"
 \
    --secondary-sources "type=GITHUB,
                location=<github-repository-2>,
                auth={type=SECRETS_MANAGER,resource=<secret-or-connection-arn-2>},
                sourceIdentifier=secondary"

aws codebuild start-build --region <aws-region> --project-name <project-name>
```

Establecimiento de una alternativa para las credenciales de origen de proyecto

Para configurar una alternativa a las credenciales de origen de proyecto, use `NO_SOURCE` como origen principal del proyecto y haga referencia al token.

```
aws codebuild create-project \
    --name <project-name> \
    --service-role <service-role-name> \
    --artifacts type=NO_ARTIFACTS \
    --environment "type=LINUX_CONTAINER,
                   computeType=BUILD_GENERAL1_SMALL,
                   image=aws/codebuild/amazonlinux-x86_64-standard:5.0" \
    --service-role <service-role-name> \
    --source "type=NO_SOURCE,
              auth={type=SECRETS_MANAGER,resource=<secret-or-connection-arn>},
              buildspec=<buildspec>"
    --secondary-sources "type=GITHUB,
                         location=<github-repository>,
                         sourceIdentifier=secondary"

aws codebuild start-build --region <aws-region> --project-name <project_name>
```

Cuando se usa `NO_SOURCE`, suele proporcionarse una especificación de compilación en el modelo de origen, ya que no está configurado directamente para usar un origen externo para obtener la especificación de compilación. Por lo general, un origen `NO_SOURCE` se encarga de clonar todos los repositorios pertinentes desde la especificación de compilación. Para asegurarse de que las credenciales configuradas estén disponibles para esas operaciones, puede habilitar la opción `git-credential-helper` en la especificación de compilación.

```
env:
  git-credential-helper: yes
```

Durante la compilación, CodeBuild leerá el `AuthServer` campo del token configurado y usará las credenciales del token para todas las solicitudes de git a ese proveedor de código externo en particular.

Opciones de configuración adicionales

Puede configurar las credenciales de nivel de cuenta de Secrets Manager mediante CloudFormation plantillas. Puede usar la siguiente CloudFormation plantilla para configurar una credencial a nivel de cuenta:

```
Parameters:
  GitHubToken:
    Type: String
    NoEcho: true
    Default: placeholder
Resources:
  CodeBuildAuthTokenSecret:
    Type: AWS::SecretsManager::Secret
    Properties:
      Description: CodeBuild auth token
      Name: codebuild-auth-token
      SecretString:
        !Join
          - ''
          - - '{"ServerType":"GITHUB","AuthType":"PERSONAL_ACCESS_TOKEN","Token":"'
            - !Ref GitHubToken
            - '"}'
      Tags:
        - Key: codebuild:source:provider
          Value: github
        - Key: codebuild:source:type
          Value: personal_access_token
  CodeBuildSecretsManagerAccountCredential:
    Type: AWS::CodeBuild::SourceCredential
    Properties:
      ServerType: GITHUB
      AuthType: SECRETS_MANAGER
      Token: !Ref CodeBuildAuthTokenSecret
```

> **Note**
>
> Si también vas a crear un proyecto en la misma pila, usa el CloudFormation atributo DependsOnpara asegurarte de que `AccountCredential` se haya creado antes del proyecto.

También puede configurar varias credenciales de nivel de fuente de Secrets Manager mediante CloudFormation plantillas. Puede usar la siguiente CloudFormation plantilla para usar varios tokens y acceder a varias fuentes:

```
Parameters:
  GitHubTokenOne:
    Type: String
    NoEcho: true
    Default: placeholder
  GitHubTokenTwo:
    Type: String
    NoEcho: true
    Default: placeholder

Resources:
  CodeBuildSecretsManagerProject:
    Type: AWS::CodeBuild::Project
    Properties:
      Name: codebuild-multitoken-example
      ServiceRole: <service-role>
      Environment:
        Type: LINUX_CONTAINER
        ComputeType: BUILD_GENERAL1_SMALL
        Image: aws/codebuild/amazonlinux-x86_64-standard:5.0
      Source:
        Type: GITHUB
        Location: <github-repository-one>
        Auth:
          Type: SECRETS_MANAGER
          Resource: !Ref CodeBuildAuthTokenSecretOne
      SecondarySources:
        - Type: GITHUB
          Location: <github-repository-two>
          Auth:
            Type: SECRETS_MANAGER
```

```
            Resource: !Ref CodeBuildAuthTokenSecretTwo
          SourceIdentifier: secondary
      Artifacts:
        Type: NO_ARTIFACTS
      LogsConfig:
        CloudWatchLogs:
          Status: ENABLED
  CodeBuildProjectIAMRoleSecretAccess:
    Type: AWS::IAM::RolePolicy
    Properties:
      RoleName: <role-name>
      PolicyName: CodeBuildProjectIAMRoleSecretAccessPolicy
      PolicyDocument:
        Version: '2012-10-17       '
        Statement:
          - Effect: Allow
            Action:
              - secretsmanager:GetSecretValue
            Resource:
              - !Ref CodeBuildAuthTokenSecretOne
              - !Ref CodeBuildAuthTokenSecretTwo
  CodeBuildAuthTokenSecretOne:
    Type: AWS::SecretsManager::Secret
    Properties:
      Description: CodeBuild auth token one
      Name: codebuild-auth-token-one
      SecretString:
        !Join
          - ''
          - - '{"ServerType":"GITHUB","AuthType":"PERSONAL_ACCESS_TOKEN","Token":"'
            - !Ref GitHubTokenOne
            - '"}'
      Tags:
        - Key: codebuild:source:provider
          Value: github
        - Key: codebuild:source:type
          Value: personal_access_token
  CodeBuildAuthTokenSecretTwo:
    Type: AWS::SecretsManager::Secret
    Properties:
      Description: CodeBuild auth token two
      Name: codebuild-auth-token-two
      SecretString:
        !Join
```

```
        - ''
        - - '{"ServerType":"GITHUB","AuthType":"PERSONAL_ACCESS_TOKEN","Token":"'
          - !Ref GitHubTokenTwo
          - '"}'
      Tags:
        - Key: codebuild:source:provider
          Value: github
        - Key: codebuild:source:type
          Value: personal_access_token
```

Eliminar proyectos de construcción en AWS CodeBuild

Puedes usar la CodeBuild consola o AWS CLI AWS los SDK para eliminar un proyecto de compilación. 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 (AWS SDK)

Eliminar un proyecto de compilación (consola)

1. Abre 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:
   - 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.

> **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 requerida. 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 el uso del AWS CLI with AWS CodeBuild, consulte laReferencia de la línea de comandos.

Eliminar un proyecto de compilación (AWS SDK)

Para obtener más información sobre su uso AWS CodeBuild con los AWS SDK, consulta la. AWS Referencia de SDK y herramientas

Obtención de direcciones URL de proyectos de compilación públicos

AWS CodeBuild te permite poner a disposición del público en general los resultados de construcción, los registros y los artefactos de tus proyectos de construcción. 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 público 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 de su proyecto que es exclusiva de su proyecto.

Para obtener la URL pública de su proyecto de compilación, utilice el siguiente procedimiento.

Para obtener la URL de un proyecto de compilación público

1. Abre la AWS CodeBuild consola 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 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.

Uso compartido de proyectos de compilación

Compartir proyectos permite a los propietarios de proyectos compartir sus AWS CodeBuild proyectos con otras AWS cuentas o usuarios. 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.

Temas

- Uso compartido de un proyecto
- Servicios relacionados
- Accede a CodeBuild los proyectos compartidos contigo
- Cómo dejar de compartir un proyecto compartido
- Identificación de un proyecto compartido
- Permisos de los proyectos compartidos

Uso compartido de un proyecto

El consumidor puede usar tanto la AWS CLI consola como la AWS CodeBuild consola para ver el proyecto y las compilaciones que has 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 .

Puedes usar la AWS CodeBuild consola, la AWS RAM consola o la AWS CLI para compartir un proyecto que te pertenezca.

Requisitos previos para compartir proyectos

Antes de empezar a compartir un proyecto, asegúrate de que es propiedad de tu AWS cuenta. No puede compartir un proyecto que se ha compartido con usted.

Para compartir un proyecto de tu propiedad (CodeBuild consola)

1. Abre la AWS CodeBuild consola en https://console.aws.amazon.com/codesuite/codebuild/home.
2. En el panel de navegación, elija Proyectos de compilación.

   > **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 de tu propiedad (AWS RAM consola)

Consulte Creación de un uso compartido de recursos en la Guía del usuario de AWS RAM .

Para compartir un proyecto de tu propiedad (AWS RAM comando)

Utilice el comando create-resource-share.

Para compartir un proyecto que posee (comando CodeBuild)

Utilice el comando put-resource-policy:

1. Cree un archivo llamado `policy.json` y copie lo siguiente en él.

   JSON

```
{
  "Version":"2012-10-17",
  "Statement":[{
    "Effect":"Allow",
    "Action":[
      "codebuild:BatchGetProjects",
      "codebuild:BatchGetBuilds",
      "codebuild:ListBuildsForProject"],
    "Resource":"arn:aws:iam::*:role/Service*"
  }]
}
```

2. Actualice `policy.json` con el ARN del proyecto y los identificadores para compartirlo. El siguiente ejemplo concede acceso de solo lectura al usuario raíz de la AWS cuenta identificada por 123456789012.

   JSON

```
{
  "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. 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 *<resource-share-arn>* valor para usarlo en el siguiente paso.

5. Ejecute el comando AWS RAM https://docs.aws.amazon.com/cli/latest/reference/ram/promote-resource-share-created-from-policy.html promote-resource-share-created-from-policy.

```
aws ram promote-resource-share-created-from-policy --resource-share-arn <resource-
share-arn>
```

Servicios relacionados

La compartición de proyectos se integra con AWS Resource Access Manager (AWS RAM), un servicio que te permite compartir tus recursos con cualquier cuenta o a través de ella. AWS

AWS 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 AWS cuentas individuales, unidades organizativas o toda una organización en AWS Organizations. AWS Organizations

Para obtener más información, consulte la Guía del usuario de AWS RAM.

Accede a CodeBuild los proyectos compartidos contigo

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.

Cómo 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 dejas de compartir un proyecto, cualquier AWS cuenta o usuario con el que lo hayas compartido anteriormente no podrá acceder al proyecto ni a sus compilaciones.

Para dejar de compartir un proyecto compartido que posee, debe quitarlo del recurso compartido. Puedes usar la AWS CodeBuild consola, la AWS RAM consola o AWS CLI para hacer esto.

Para dejar de compartir un proyecto compartido de tu propiedad (AWS RAM consola)

Consulte Actualización de un recurso compartido en la Guía del usuario de AWS RAM .

Para dejar de compartir un proyecto compartido de tu propiedad (AWS CLI)

Utilice el comando disassociate-resource-share.

Para dejar de compartir un proyecto de tu propiedad (comando) 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 utilizarla AWS CLI para identificar proyectos compartidos.

Para identificar los proyectos compartidos con sus AWS cuenta o usuario (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 de compilación

Una etiqueta es una etiqueta de atributo personalizada que usted o AWS asigna a un AWS recurso. Cada AWS etiqueta consta de dos partes:

- Una clave de etiqueta (por ejemplo, `CostCenter`, `Environment`, `Project` o `Secret`). Las claves de etiquetas distinguen entre mayúsculas y minúsculas.
- Un campo opcional que se denomina valor de etiqueta (por ejemplo, `111122223333`, `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 etiquetas, los valores de las etiquetas distinguen mayúsculas de 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 Tags.

Las etiquetas ayudan a identificar y organizar AWS los recursos. Muchos AWS servicios 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 CodeBuild proyecto que 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 usar la CodeBuild consola AWS CLI, las CodeBuild API o AWS los SDK para agregar, administrar y eliminar 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 AWS CodeBuild recursos.

> **Important**
>
> Cuando se usa la característica de capacidad reservada, otros proyectos de la misma cuenta pueden acceder a los datos almacenados en caché en las instancias de flota (incluidos los archivos de origen, las capas de Docker y los directorios almacenados en caché que se indican en la especificación de compilación). Esto es así por diseño y permite que los proyectos de la misma cuenta compartan instancias de flotas.

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

Añadir etiquetas a un proyecto puede ayudarte a identificar y organizar tus AWS recursos y a gestionar 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 Tags. Cuando tenga las etiquetas, puede crear políticas de IAM para administrar el acceso al proyecto en función de estas etiquetas. Puede utilizar la CodeBuild consola o la AWS CLI para añadir etiquetas a un proyecto.

> **Important**
>
> Cuando se usa la característica de capacidad reservada, otros proyectos de la misma cuenta pueden acceder a los datos almacenados en caché en las instancias de flota (incluidos los archivos de origen, las capas de Docker y los directorios almacenados en caché que se indican en la especificación de compilación). Esto es así por diseño y permite que los proyectos de la misma cuenta compartan instancias de flotas.

Para obtener más información sobre cómo agregar etiquetas a un proyecto al crearlo, consulte Agregar una etiqueta a un proyecto (consola).

> **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 AWS CodeBuild recursos.

Temas

- Agregar una etiqueta a un proyecto (consola)
- Agregar una etiqueta a un proyecto (AWS CLI)

Agregar una etiqueta a un proyecto (consola)

Puede utilizar la CodeBuild consola para añadir una o más etiquetas a un CodeBuild proyecto.

1. Abre la CodeBuild consola 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 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 Clave, escriba un nombre para la etiqueta. Puede agregar un valor opcional para la etiqueta en 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 Cree 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 la 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 AWS recursos y a 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 AWS CodeBuild recursos.

Consultar etiquetas de un proyecto (consola)

Puede utilizar la CodeBuild consola para ver las etiquetas asociadas a un CodeBuild proyecto.

1. Abra la CodeBuild consola 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 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 se ejecuta correctamente, este comando devuelve JSON-formatted información sobre el 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 agregar 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 Tags.

> **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 AWS CodeBuild recursos.

Editar una etiqueta de un proyecto (consola)

Puedes usar la CodeBuild consola para editar las etiquetas asociadas a un CodeBuild proyecto.

1. Abra la CodeBuild consola 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 Configuración. Elija Etiquetas de proyecto de compilación.
4. Elija Edit (Edición de).
5. Realice 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 Cambia la configuración de un proyecto de compilación (AWS CLI). Actualice la `tags` sección de los JSON-formatted datos que utiliza para actualizar el proyecto.

Quitar una etiqueta de un proyecto

Puede quitar una o varias etiquetas asociadas con un proyecto. Al eliminar una etiqueta, no se elimina la etiqueta de otros AWS recursos que estén asociados a 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 AWS CodeBuild recursos.

Quitar una etiqueta de un proyecto (consola)

Puedes usar la CodeBuild consola para eliminar la asociación entre una etiqueta y un CodeBuild proyecto.

1. Abre la CodeBuild consola 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 Configuración. Elija Etiquetas de proyecto de compilación.
4. Elija Edit (Edición de).
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 Cambia la configuración de un proyecto de compilación (AWS CLI). Actualice la `tags` sección de JSON-formatted los datos con una lista actualizada de etiquetas que no contenga las que desee eliminar. Si desea eliminar todas las etiquetas, actualice la sección `tags` para:

```
"tags: []"
```

> **Note**
>
> Si elimina un proyecto de CodeBuild construcción, todas las asociaciones de etiquetas se eliminan del proyecto de construcción eliminado. No es necesario quitar las etiquetas antes de eliminar un proyecto de compilación.

Usa corredores con AWS CodeBuild

AWS CodeBuild admite la integración con los corredores GitHub Actions, los GitLab corredores autogestionados y el corredor Buildkite.

Temas

- Self-hosted GitHub Actions se ejecuta en AWS CodeBuild
- Self-managed GitLab se ejecuta en AWS CodeBuild
- Self-managed BuildKite Runner en AWS CodeBuild

Self-hosted GitHub Actions se ejecuta 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 alojados en máquinas. CodeBuild

Los pasos generales para configurar un CodeBuild proyecto para que ejecute 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 CodeBuild-hosted GitHub acciones.

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 ARM-based instancias.

Temas

- Acerca del ejecutor de CodeBuild-hosted GitHub acciones
- Tutorial: Configurar un ejecutor de CodeBuild-hosted GitHub acciones
- Solución de problemas del webhook
- El ejecutor Actions admite anulaciones de CodeBuild-hosted GitHub etiquetas
- Calcule imágenes compatibles con el ejecutor CodeBuild-hosted GitHub Actions

Acerca del ejecutor de CodeBuild-hosted GitHub acciones

Las siguientes son algunas de las preguntas más frecuentes sobre el corredor de CodeBuild-hosted GitHub acciones.

¿Cuándo debo incluir las anulaciones de instancias e imágenes en la etiqueta?

Puedes incluir las modificaciones de imagen e instancia en la etiqueta para especificar un entorno de creación 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, es útil cuando necesita usar una matriz para sus trabajos de flujo de trabajo.

```
name: Hello World
on: [push]
jobs:
  Hello-World-Job:
    runs-on:
      - codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}
        image:${{ matrix.os }}
        instance-size:${{ matrix.size }}
    strategy:
      matrix:
        include:
          - os: arm-3.0
            size: small
          - os: linux-5.0
            size: 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 usar CloudFormation ¿para esta función?

Sí, puedes incluir un grupo de filtros en tu 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 Filtrar eventos de GitHub webhooks (CloudFormation).

Si necesitas ayuda para configurar las credenciales del proyecto en tu CloudFormation plantilla, consulta AWS::CodeBuild:: SourceCredential en la Guía del AWS CloudFormation usuario para obtener más información.

¿Cómo puedo enmascarar los secretos al usar esta característica?

De forma predeterminada, los secretos que se imprimen en el registro no están enmascarados. Si desea enmascarar los secretos, puede utilizar la sintaxis siguiente: `::add-mask::`*`value`*. A continuación se muestra un ejemplo de cómo puede usar esta sintaxis en el YAML:

```
name: Secret Job
on: [push]
jobs:
  Secret-Job:
    runs-on: codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}
    env:
      SECRET_NAME: "secret-name"
    steps:
      - run: echo "::add-mask::$SECRET_NAME"
```

Para obtener más información, consulte Enmascarar un valor en un inicio de sesión GitHub.

¿Puedo recibir eventos de webhook de GitHub Actions desde varios repositorios dentro de un mismo proyecto?

CodeBuild admite webhooks organizativos y globales, que reciben eventos de una organización o empresa específica. Para obtener más información, consulte GitHub webhooks globales y organizacionales.

¿Qué regiones admiten el uso de un CodeBuild-hosted GitHub Actions Runner?

CodeBuild-hosted GitHub Los corredores de Actions están disponibles en todas CodeBuild las regiones. Para obtener más información sobre Regiones de AWS dónde CodeBuild está disponible, consulte AWS Servicios por región.

¿Qué plataformas admiten el uso de un ejecutor de CodeBuild-hosted GitHub acciones?

CodeBuild-hosted GitHub Los ejecutores de acciones son compatibles tanto con Amazon EC2 como con Compute. AWS Lambda 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.

Tutorial: Configurar un ejecutor de CodeBuild-hosted GitHub acciones

Este tutorial le muestra cómo configurar sus CodeBuild proyectos para ejecutar trabajos de GitHub Actions. Para obtener más información sobre el uso de GitHub Actions con, CodeBuild consulteTutorial: Configurar un ejecutor de CodeBuild-hosted GitHub acciones.

Para completar este tutorial, primero debe:

- Conéctate con un token de acceso personal, un secreto de Secrets Manager, una aplicación OAuth o GitHub una aplicación. Si quieres conectarte con una aplicación de 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 instrucciones adicionales, consulte GitHub y acceso a GitHub Enterprise Server en CodeBuild.
- 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 un token de acceso personal, un secreto de Secrets Manager, una aplicación OAuth o GitHub una aplicación. Para obtener instrucciones, consulte GitHub y acceso a GitHub Enterprise Server en CodeBuild.
  - Puedes importar tus GitHub credenciales a través de la ImportSourceCredentials API. Esto solo se puede hacer con un token de acceso personal. Si se conecta mediante una aplicación OAuth, debe usar la consola. Para obtener instrucciones, consulte Conéctese GitHub con un token de acceso (CLI).

> **Note**
>
> Esto solo debe hacerse si no te has conectado GitHub a tu 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. También puedes elegir GitHub Enterprise como tu proveedor de código fuente. Para obtener

más información sobre cómo crear un webhook en GitHub Enterprise, consulteGitHub webhooks manuales.

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).
3. En Tipo de proyecto, seleccione Proyecto de ejecutor.

   En Ejecutor:

   a. Para el proveedor Runner, elija GitHub.

   b. En Ubicación del ejecutor, seleccione Repositorio.

   c. Para la URL del repositorio en Repositorio, elija https://github.com/user-name/repository-name.

   > **Note**
   >
   > De forma predeterminada, el proyecto únicamente recibirá eventos `WORKFLOW_JOB_QUEUED` para un solo repositorio. Si quiere recibir eventos para todos los repositorios de una organización o empresa, consulte GitHub webhooks globales y organizacionales.

4. • En Environment (Entorno):
   - Elija una Imagen del entorno y una Computación 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:
     - Tenga en cuenta que su especificación de compilación se ignorará a menos que se agregue `buildspec-override:true` como etiqueta. En su lugar, la CodeBuild anulará para usar comandos que configurarán el ejecutor autohospedado.
5. Continúe con los valores predeterminados y, a continuación, elija Crear el proyecto de compilación.

6. 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, consulte Uso de etiquetas con ejecutores autohospedados y El ejecutor Actions admite anulaciones de CodeBuild-hosted GitHub etiquetas.

Actualiza tu flujo de trabajo de GitHub Acciones (YAML)

Ve a la `runs-on`configuración de tu flujo de trabajo de GitHub Actions (YAML) `GitHub`y actualízala para configurar tu entorno de compilación. Para ello, puede hacer una de las acciones siguientes:

- Puede especificar el nombre del proyecto y el ID de ejecución, en cuyo caso la compilación usará la configuración de proyecto existente para la computación, la imagen, la versión de la imagen y el tamaño de la instancia. El nombre del proyecto es necesario para vincular los ajustes AWS relacionados con 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 asigna la compilación a ejecuciones de flujo de trabajo específicas y detiene la compilación cuando se cancela la ejecución del flujo de trabajo. Para obtener más información, consulte el `contexto de github`.

  ```
  runs-on: codebuild-<project-name>-${{ github.run_id }}-${{ github.run_attempt }}
  ```

  > **Note**
  >
  > Asegúrate de que *<project-name>* coincide con el nombre del proyecto que creaste 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 puede anular el tipo de computación e imagen en la etiqueta. Consulte Calcule imágenes compatibles con el ejecutor CodeBuild-hosted GitHub Actions para ver una lista de imágenes seleccionadas. Para usar imágenes personalizadas, consulte El ejecutor Actions admite anulaciones de CodeBuild-hosted GitHub etiquetas. El tipo de computación y la imagen de la etiqueta anularán la configuración del entorno en el proyecto. Para anular la configuración de su entorno para una compilación de procesamiento CodeBuild EC2 o Lambda, utilice la siguiente sintaxis:

  ```
  runs-on:
    - codebuild-<project-name>-${{ github.run_id }}-${{ github.run_attempt }}
      image:<environment-type>-<image-identifier>
      instance-size:<instance-size>
  ```

  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 }}
          image:arm-3.0
          instance-size:small
      steps:
        - run: echo "Hello World!"
  ```

- Puede anular la flota utilizada para su compilación en la etiqueta. Al hacerlo, se anularán los valores de flota configurados en el proyecto para usar la flota especificada. Para obtener más información, consulte Ejecución de compilaciones en flotas de capacidad reservada. Si quiere anular la configuración de la flota para una compilación de computación de Amazon EC2, utilice la siguiente sintaxis:

  ```
  runs-on:
    - codebuild-<project-name>-${{ github.run_id }}-${{ github.run_attempt }}
  ```

```
    fleet:<fleet-name>
```

Para anular tanto la flota como la imagen utilizadas para la compilación, utilice la sintaxis siguiente:

```
runs-on:
  - codebuild-<project-name>-${{ github.run_id }}-${{ github.run_attempt }}
    fleet:<fleet-name>
    image:<environment-type>-<image-identifier>
```

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 }}
        fleet:myFleet
        image:arm-3.0
    steps:
      - run: echo "Hello World!"
```

- Para ejecutar tus trabajos de GitHub Actions en una imagen personalizada, puedes configurar una imagen personalizada en tu CodeBuild proyecto y evitar tener que incluir una etiqueta de reemplazo de imagen. CodeBuild utilizará la imagen configurada en el proyecto si no se proporciona una etiqueta de anulación de imagen.
- Si lo desea, puede proporcionar etiquetas distintas de las CodeBuild compatibles. Estas etiquetas se ignorarán para el fin de anular los atributos de la compilación, pero no provocarán un error de la solicitud de webhook. Por ejemplo, agregar `testLabel` como etiqueta no impedirá que la compilación se ejecute.

> **Note**
>
> Si una dependencia proporcionada por GitHub-hosted los ejecutores no está disponible en el CodeBuild entorno, puede instalarla mediante GitHub Actions en la ejecución del flujo de trabajo. Por ejemplo, puede usar la acción `setup-python` para instalar Python en su entorno de compilación.

Ejecuta los comandos buildspec en las fases INSTALL, PRE_BUILD y POST_BUILD

De forma predeterminada, CodeBuild omite cualquier comando de buildspec al ejecutar una compilación de Actions autohospedada. GitHub Para ejecutar comandos buildspec durante la compilación, se puede agregar `buildspec-override:true` como sufijo a la etiqueta:

```
runs-on:
  - codebuild-<project-name>-${{ github.run_id }}-${{ github.run_attempt }}
    buildspec-override:true
```

Al usar este comando, CodeBuild se creará una carpeta llamada carpeta de origen `actions-runner` principal del contenedor. Cuando el ejecutor de GitHub Actions se inicie durante la `BUILD` fase, se ejecutará en el `actions-runner` directorio.

Existen varias limitaciones a la hora de utilizar una modificación de las especificaciones de compilación en una compilación de Actions autohospedada: GitHub

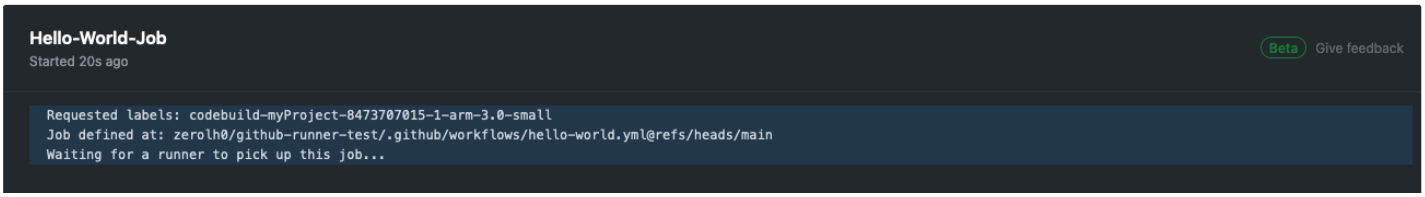
- CodeBuild no ejecutará los comandos buildspec durante la fase, ya que el ejecutor autohospedado se ejecuta en la `BUILD` fase. `BUILD`
- CodeBuild no descargará ninguna fuente primaria o secundaria durante la fase. `DOWNLOAD_SOURCE` Si ha configurado un archivo buildspec, solo se descargará ese archivo del origen principal del proyecto.
- Si un comando de compilación falla en la `INSTALL` fase `PRE_BUILD` o, no CodeBuild se iniciará el ejecutor autohospedado y será necesario cancelar manualmente el trabajo del flujo de trabajo de GitHub Actions.
- CodeBuild obtiene el token del ejecutor durante la `DOWNLOAD_SOURCE` fase, que tiene un plazo de caducidad de una hora. Si tus `PRE_BUILD` `INSTALL` fases superan una hora, el token de corredor puede caducar antes de que comience el corredor GitHub autohospedado.

Paso 3: revisión de los resultados

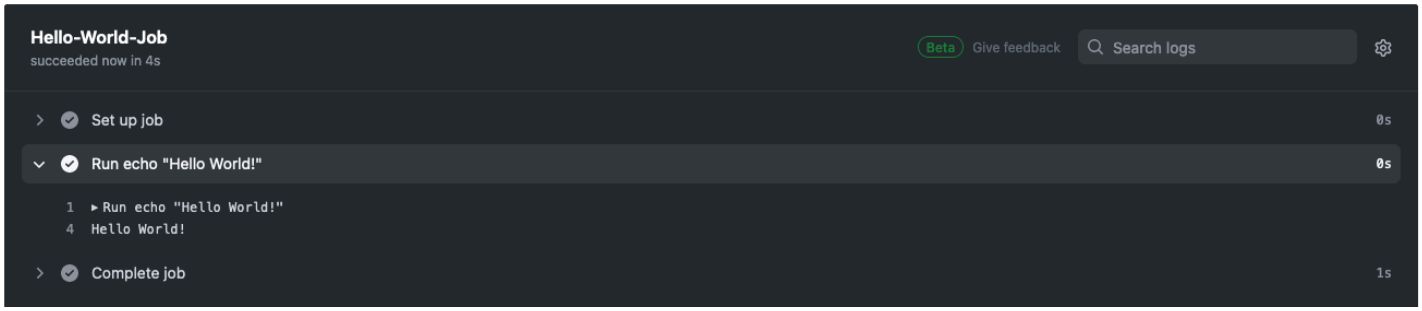
Siempre que se ejecute un flujo de trabajo de GitHub Actions, CodeBuild recibirá los eventos del trabajo del flujo de trabajo a través del webhook. Para cada trabajo del flujo de trabajo, CodeBuild inicia una compilación para ejecutar un ejecutor efímero GitHub de Actions. Este es responsable de ejecutar un único trabajo del flujo de trabajo. Una vez que se complete el trabajo, el ejecutor y el proceso de compilación asociado finalizarán de forma inmediata.

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 de este.



GitHub Opciones de configuración del ejecutor de acciones

Puede especificar las siguientes variables de entorno en la configuración del proyecto para modificar la configuración de instalación de los ejecutores autoalojados.

`CODEBUILD_CONFIG_GITHUB_ACTIONS_ORG_REGISTRATION_NAME`

CodeBuild registrará a los corredores autohospedados con el nombre de la organización especificado como valor de esta variable de entorno. Para obtener más información sobre el registro de ejecutores en la organización y los permisos necesarios, consulte Create configuration for a just-in-time runner for an organization.

`CODEBUILD_CONFIG_GITHUB_ACTIONS_ENTERPRISE_REGISTRATION_NAME`

CodeBuild registrará los ejecutores autohospedados con el nombre de empresa especificado como valor de esta variable de entorno. Para obtener más información sobre el registro de ejecutores en el nivel de empresa y los permisos necesarios, consulte Creación de una configuración para un ejecutor justo a tiempo para una empresa.

> **Note**
>
> De forma predeterminada, los ejecutores empresariales no están disponibles en los repositorios de la organización. Para que los ejecutores autoalojados recojan trabajos de flujos de trabajo, es posible que deba configurar los ajustes de acceso al grupo de ejecutores. Para obtener más información, consulte Making enterprise runners available to repositories.

`CODEBUILD_CONFIG_GITHUB_ACTIONS_RUNNER_GROUP_ID`

CodeBuild registrará los ejecutores autohospedados con el identificador entero del grupo de ejecutores almacenado como valor de esta variable de entorno. De forma predeterminada este valor es 1. Para obtener más información sobre los grupos de ejecutores autoalojados, consulte Administración del acceso a los ejecutores autohospedados mediante grupos.

`CODEBUILD_CONFIG_GITHUB_ACTIONS_ORG_REGISTRATION_NAME`

Para configurar el registro de los ejecutores a nivel de organización mediante el archivo YAML del flujo de trabajo de GitHub Actions, puedes usar la siguiente sintaxis:

```
name: Hello World
on: [push]
jobs:
  Hello-World-Job:
    runs-on:
      - codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}
        organization-registration-name:myOrganization
    steps:
      - run: echo "Hello World!"
```

`CODEBUILD_CONFIG_GITHUB_ACTIONS_ENTERPRISE_REGISTRATION_NAME`

Para configurar el registro de ejecutores de nivel empresarial mediante el archivo YAML del flujo de trabajo de GitHub Actions, puedes usar la siguiente sintaxis:

```
name: Hello World
on: [push]
jobs:
  Hello-World-Job:
    runs-on:
```

```
      - codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}
        enterprise-registration-name:myEnterprise
    steps:
      - run: echo "Hello World!"
```

CODEBUILD_CONFIG_GITHUB_ACTIONS_RUNNER_GROUP_ID

Para configurar el registro de los corredores con un ID de grupo de corredores específico mediante el archivo YAML del flujo de trabajo de GitHub Actions, puedes usar la siguiente sintaxis:

```
name: Hello World
on: [push]
jobs:
  Hello-World-Job:
    runs-on:
      - codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}
        registration-group-id:3
    steps:
      - run: echo "Hello World!"
```

Filtrar los eventos del webhook de GitHub Actions (CloudFormation)

La siguiente YAML-formatted parte de una CloudFormation plantilla crea un grupo de filtros que activa una compilación cuando se evalúa como verdadera. El siguiente 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 `\[CI-CodeBuild\]` normal.

```
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: CODEBUILD_DEFAULT_WEBHOOK_SOURCE_LOCATION
    Triggers:
      Webhook: true
      ScopeConfiguration:
        Name: organization-name
        Scope: GITHUB_ORGANIZATION
      FilterGroups:
        - - Type: EVENT
            Pattern: WORKFLOW_JOB_QUEUED
          - Type: WORKFLOW_NAME
            Pattern: \[CI-CodeBuild\]
```

Filtrar GitHub acciones y eventos de webhook (AWS CDK)

La siguiente AWS CDK plantilla crea un grupo de filtros que activa una compilación cuando se evalúa como verdadera. El siguiente grupo de filtros especifica una solicitud de trabajo del flujo de trabajo de GitHub Actions.

```
import { aws_codebuild as codebuild } from 'aws-cdk-lib';
import {EventAction, FilterGroup} from "aws-cdk-lib/aws-codebuild";

const source = codebuild.Source.gitHub({
      owner: 'owner',
      repo: 'repo',
      webhook: true,
      webhookFilters: [FilterGroup.inEventOf(EventAction.WORKFLOW_JOB_QUEUED)],
    })
```

Filtrar GitHub acciones y eventos de webhook (Terraform)

La siguiente plantilla de Terraform crea un grupo de filtros que desencadena una compilación cuando se evalúa como true. El siguiente grupo de filtros especifica una solicitud de trabajo del flujo de trabajo de GitHub Actions.

```
resource "aws_codebuild_webhook" "example" {
  project_name = aws_codebuild_project.example.name
  build_type   = "BUILD"
  filter_group {
    filter {
      type    = "EVENT"
      pattern = "WORKFLOW_JOB_QUEUED"
    }
```

```
  }
}
```

Filtrar GitHub acciones y eventos de webhook (AWS CLI)

Los siguientes AWS CLI comandos crean un proyecto de ejecución de GitHub Actions autohospedado con un grupo de filtros de solicitudes de trabajo del flujo de trabajo de GitHub Actions que activa una compilación cuando se evalúa como verdadera.

```
aws codebuild create-project \
--name <project name> \
--source "{\"type\":\"GITHUB\",\"location\":\"<repository location>\",\"buildspec\":
\"\"}" \
--artifacts {"\"type\":\"NO_ARTIFACTS\""} \
--environment "{\"type\": \"LINUX_CONTAINER\",\"image\": \"aws/codebuild/amazonlinux-
x86_64-standard:5.0\",\"computeType\": \"BUILD_GENERAL1_MEDIUM\"}" \
--service-role "<service role ARN>"
```

```
aws codebuild create-webhook \
--project-name <project name> \
--filter-groups "[[{\"type\":\"EVENT\",\"pattern\":\"WORKFLOW_JOB_QUEUED\"}]]"
```

Solución de problemas del webhook

Problema: el webhook que configuraste Tutorial: Configurar un ejecutor de CodeBuild-hosted GitHub acciones no funciona o tu flujo de trabajo no funciona. GitHub

Posibles causas:

- Puede que el evento Trabajos de flujo de trabajo del webhook no pueda desencadenar una compilación. Revise los registros de Respuesta para ver la respuesta o el mensaje de error.
- Sus trabajos se asignan a un agente de ejecutor incorrecto debido a su configuración de etiquetas. Este problema puede producirse cuando uno de los trabajos de una sola ejecución de flujo de trabajo tiene menos etiquetas que otro trabajo. Por ejemplo, si tiene dos trabajos con las siguientes etiquetas en la misma ejecución de flujo de trabajo:
  - Trabajo 1: `codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}`
  - Trabajo 2: `codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}`, `instance-size:medium`

Al enrutar un trabajo de GitHub Actions autohospedado, GitHub lo enrutará a cualquier ejecutor con todas las etiquetas especificadas para el trabajo. Este comportamiento significa que el ejecutor creado para Trabajo 1 o Trabajo 2 puede recoger Trabajo 1, pero Trabajo 2 solo lo puede recoger el ejecutor creado para Trabajo 2, ya que tiene una etiqueta adicional. Si el ejecutor creado para Trabajo 1 recoge Trabajo 1, Trabajo 2 se bloqueará, ya que el ejecutor de Trabajo 1 no tiene la etiqueta `instance-size:medium`.

Soluciones recomendadas:

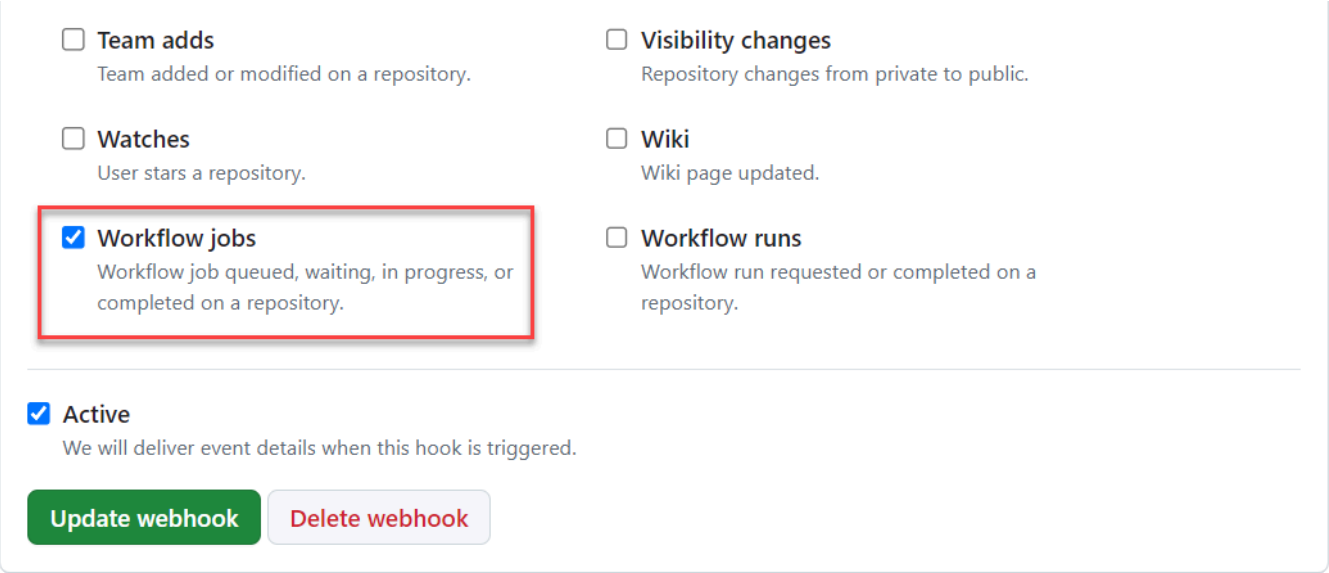
Al crear varios trabajos dentro de la misma ejecución de flujo de trabajo, utilice el mismo número de anulaciones de etiquetas para cada trabajo o asigne a cada trabajo una etiqueta personalizada, como `job1` o `job2`.

Asignar a cada trabajo una etiqueta personalizada única es la opción más fiable. Garantiza que el conjunto de etiquetas de ningún trabajo sea un subconjunto del de otro trabajo. Cuando cada trabajo lleva una etiqueta que ningún otro trabajo tiene, hace GitHub coincidir cada trabajo solo con el ejecutor creado para él, independientemente del número de anulaciones que utilice cada trabajo. En el siguiente flujo de trabajo, cada trabajo tiene una etiqueta (`job1`y`job2`) única. Aunque el Job 2 tiene una `instance-size:medium` anulación adicional, el Job 1 ya no se puede hacer coincidir con el corredor creado para el Job 2, por lo que ninguno de los dos se bloquea:

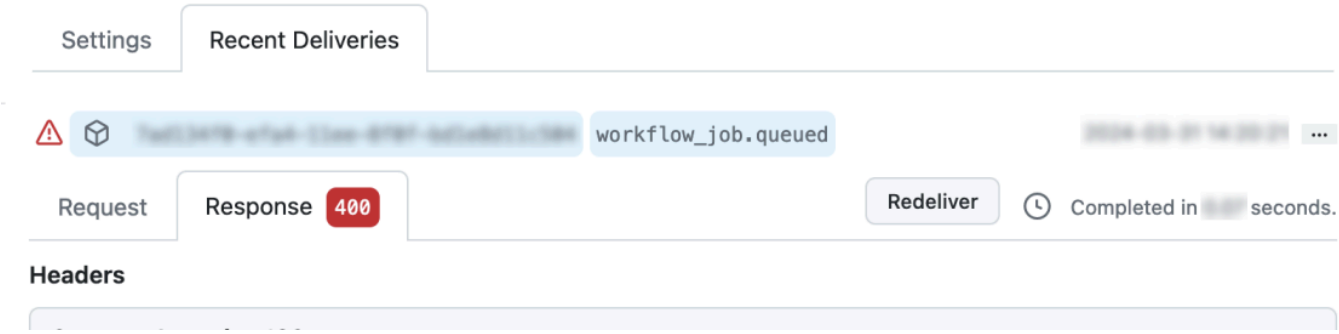
```
name: Hello World
on: [push]
jobs:
  job1:
    runs-on:
      - codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}
      - job1
    steps:
      - run: echo "Hello from Job 1!"
  job2:
    runs-on:
      - codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}
      - instance-size:medium
      - job2
    steps:
      - run: echo "Hello from Job 2!"
```

Si el error persiste, sigue estas instrucciones para solucionar el problema.

1. Abre la GitHub consola en `https://github.com/user-name/repository-name/settings/hooks` para ver la configuración del webhook de tu repositorio. En esta página, verá un webhook que se ha creado para su repositorio.
2. Selecciona Editar y confirma que el webhook está habilitado para entregar eventos de Workflow Jobs.



3. Vaya a la pestaña Recent Deliveries, busque el evento `workflow_job.queued` correspondiente y expándalo.
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.



6. Como alternativa, puedes depurar los errores de los webhooks mediante las API GitHub. Puede ver las entregas recientes de un webhook mediante la API List deliveries for a repository webhook:

```
gh api \
```

```
  -H "Accept: application/vnd.github+json" \
  -H "X-GitHub-Api-Version: 2022-11-28" \
  /repos/owner/repo/hooks/hook-id/deliveries
```

Después de encontrar la entrega de webhook que deseas depurar y anotar su ID, puedes usar la API de webhook Get a delivery for a repository. CodeBuildPuedes encontrar la respuesta a la carga útil de entrega del webhook en la sección: `response`

```
gh api \
  -H "Accept: application/vnd.github+json" \
  -H "X-GitHub-Api-Version: 2022-11-28" \
  /repos/owner/repo/hooks/hook-id/deliveries/delivery-id
```

Problema: Cuando GitHub las reglas de protección del despliegue están habilitadas, las acciones se implementan CodeBuild antes de que se apruebe la implementación.

Causas posibles: CodeBuild busca la implementación y el entorno asociados al trabajo de GitHub Actions, si existen, para comprobar si el despliegue está aprobado. Si CodeBuild no se obtiene la implementación o el entorno, es posible que la CodeBuild compilación se active prematuramente.

Soluciones recomendadas: compruebe que las credenciales asociadas a sus CodeBuild proyectos tengan permisos de lectura para las implementaciones y las acciones internas. GitHub

El ejecutor Actions admite anulaciones de CodeBuild-hosted GitHub etiquetas

En tu flujo de trabajo de GitHub Actions (YAML), puedes incluir una variedad de sustituciones de etiquetas que modifiquen la compilación del ejecutor autohospedado. Se CodeBuild ignorarán todas las compilaciones que no reconozcas, pero no rechazarán tu solicitud de webhook. El siguiente flujo de trabajo (YAML) define dos tareas: `job1` usa anulaciones para la imagen, el tamaño de la instancia, la flota y la especificación de compilación, y usa solo una anulación de imagen. `job2` Asigne a cada trabajo una etiqueta única (`job1`y`job2`) para que dirija cada trabajo al GitHub ejecutor creado para él:

```
name: Hello World
on: [push]
jobs:
  job1:
    runs-on:
      - codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}
```

```
      - image:arm-3.0
      - instance-size:small
      - fleet:myFleet
      - buildspec-override:true
      - job1
    steps:
      - run: echo "Hello from Job 1!"
  job2:
    runs-on:
      - codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}
      - image:linux-5.0
      - job2
    steps:
      - run: echo "Hello from Job 2!"
```

También puede utilizar una estrategia matricial para ejecutar un único trabajo en varias configuraciones. Las expansiones de un trabajo de matriz única siempre tienen el mismo número de etiquetas, por lo que el problema de coincidencia de ejecutores no las afecta. Sin embargo, si una ejecución de flujo de trabajo contiene más de una tarea matricial, asigne a cada tarea matricial su propia etiqueta personalizada única (por ejemplo, `matrix-job-1` y`matrix-job-2`) para que GitHub no haga coincidir una tarea con una ejecución creada para una tarea matricial diferente:

```
name: Hello World
on: [push]
jobs:
  Hello-World-Job:
    runs-on:
      - codebuild-myProject-${{ github.run_id }}-${{ github.run_attempt }}
      - image:${{ matrix.os }}
      - instance-size:${{ matrix.size }}
      - fleet:myFleet
      - buildspec-override:true
      - matrix-job-1
    strategy:
      matrix:
        include:
          - os: arm-3.0
            size: small
          - os: linux-5.0
            size: large
    steps:
      - run: echo "Hello World!"
```

> **Note**
>
> Si su trabajo de flujo de trabajo está pendiente GitHub, consulte Solución de problemas del webhook y Uso de etiquetas personalizadas para enrutar los trabajos.

`codebuild-<project-name>-${{github.run_id}}-${{github.run_attempt}}` (obligatorio)

- Ejemplo: `codebuild-fake-project-${{ github.run_id }}-${{ github.run_attempt }}`
- Necesario para todos los YAML del flujo de trabajo de GitHub Actions. *<project name>*debe ser igual al nombre del proyecto para el que está configurado el webhook del ejecutor autohospedado.

`image:<environment-type>-<image-identifier>`

- Ejemplo: `image:arm-3.0`
- Anula la imagen y el tipo de entorno utilizados al iniciar la compilación del ejecutor autoalojado con una imagen seleccionada. Para obtener más información sobre los valores admitidos, consulte Calcule imágenes compatibles con el ejecutor CodeBuild-hosted GitHub Actions.
  - Para anular la imagen y el tipo de entorno utilizados con una imagen personalizada, utilice `image:custom-<environment-type>-<custom-image-identifier>`
  - Ejemplo: `image:custom-arm-public.ecr.aws/codebuild/amazonlinux-aarch64-standard:3.0`

> **Note**
>
> Si la imagen personalizada se encuentra en un registro privado, consulte Configuración de una credencial de registro privada para ejecutores autoalojados.

`instance-size:<instance-size>`

- Ejemplo: `instance-size:medium`
- Anula el tipo de instancia utilizado al iniciar la compilación del ejecutor autoalojado. Para obtener más información sobre los valores admitidos, consulte Calcule imágenes compatibles con el ejecutor CodeBuild-hosted GitHub Actions.

`fleet:<fleet-name>`

- Ejemplo: `fleet:myFleet`
- Anula la configuración de flota establecida en el proyecto para usar la flota especificada. Para obtener más información, consulte Ejecución de compilaciones en flotas de capacidad reservada.

`buildspec-override:<boolean>`

- Ejemplo: `buildspec-override:true`
- Permite que la compilación ejecute comandos buildspec en las fases `INSTALL`, `PRE_BUILD` y `POST_BUILD` si está establecido en `true`.

`<unique-label>`(se recomienda cuando la ejecución de un flujo de trabajo tiene varios trabajos)

- Ejemplo: `job1`
- Una etiqueta personalizada que usted defina y que ningún otro trabajo de la misma ejecución de flujo de trabajo utilice. Al asignar a cada trabajo una etiqueta única, se garantiza GitHub que cada trabajo se dirija solo al corredor que se creó para él. Sin ella, cuando los trabajos de la misma ejecución de flujo de trabajo tienen un número diferente de anulaciones de etiquetas, un corredor creado para un trabajo con más etiquetas puede recoger un trabajo con menos etiquetas, lo que deja el segundo trabajo sin un corredor. Para obtener más información, consulte Solución de problemas del webhook.

Anulación de una sola etiqueta

CodeBuild permite proporcionar varias anulaciones en una sola etiqueta mediante lo siguiente:

> **Note**
>
> Como todas las anulaciones se combinan en una sola etiqueta, el ejecutor de cada trabajo se registra con exactamente una etiqueta. Esto significa que el problema de coincidencia entre varios ejecutores de tareas descrito en Solución de problemas del webhook no puede producirse con este formato, ni siquiera sin una etiqueta única en cada trabajo. Utilice el formato de etiqueta única si prefiere este comportamiento y sus trabajos solo necesitan las anulaciones que admite. No lo use cuando necesite una anulación que no sea compatible. Por ejemplo, las imágenes personalizadas (incluidas las imágenes de un registro privado) solo se pueden especificar mediante el formato de `image:custom-<environment-`

type-*<custom-image-identifier>* etiqueta. Para esas anulaciones, utilice el formato de etiqueta independiente descrito anteriormente en este tema.

- Para anular la configuración de su entorno para una compilación de Amazon EC2/Lambda Compute, utilice la siguiente sintaxis:

```
runs-on: codebuild-<project-name>-${{ github.run_id }}-
${{ github.run_attempt }}-<environment-type>-<image-identifier>-<instance-size>
```

- Para anular la configuración de la flota en una compilación de computación de Amazon EC2, utilice la sintaxis siguiente:

```
runs-on: codebuild-<project-name>-${{ github.run_id }}-${{ github.run_attempt }}-
fleet-<fleet-name>
```

- Para anular tanto la flota como la imagen utilizadas para la compilación, utilice la sintaxis siguiente:

```
runs-on: codebuild-<project-name>-${{ github.run_id }}-
${{ github.run_attempt }}-image-<image-version>-fleet-<fleet-name>
```

- Para ejecutar comandos buildspec durante la compilación, se puede agregar `-with-buildspec` como sufijo a la etiqueta:

```
runs-on: codebuild-<project-name>-${{ github.run_id }}-
${{ github.run_attempt }}-<image>-<image-version>-<instance-size>-with-buildspec
```

- Si lo desea, puede proporcionar una anulación del tamaño de la instancia sin anular la imagen. En el caso de las compilaciones de Amazon EC2, puede excluir tanto el tipo de entorno como el identificador de imagen. En el caso de las compilaciones de Lambda, puede excluir el identificador de imagen.

Calcule imágenes compatibles con el ejecutor CodeBuild-hosted GitHub Actions

En la etiqueta que configuróTutorial: Configurar un ejecutor de CodeBuild-hosted GitHub acciones, 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. Para obtener más información acerca de

Tipo de entorno	Identificador de imagen	Tamaño de instancia	Plataforma	Imagen resuelta	Definición

`linux`	`4.0`	`small`
		`medium`
		`large`
		`xlarge`
			Amazon Linux 2	`aws/codebuild/amazonlinux-x86_64-standard:4.0`	al/standard/4.0

`linux`	`5.0`	`2xlarge`
		`gpu_small`
		`gpu_large`
			Amazon Linux 2023	`aws/codebuild/amazonlinux-x86_64-standard:5.0`	al/standard/5.0

`linux-ec2`	`latest`	`small`
		`medium`
		`large`
			Amazon Linux 2023	`aws/codebuild/ami/amazonlinux-x86_64-base:latest`	Ninguno

`arm`	`2.0`	`small`
		`medium`
		`large`
		`xlarge`
			Amazon Linux 2	`aws/codebuild/amazonlinux-aarch64-standard:2.0`	al/aarch64/0standard/2.

`arm`	`3.0`	`2xlarge`	Amazon Linux 2023	`aws/codebuild/amazonlinux-aarch64-standard:3.0`	al/aarch64/0standard/3.

Tipo de entorno	Identificador de imagen	Tamaño de instancia	Plataforma	Imagen resuelta	Definición

`arm-ec2`	`latest`	`small`
		`medium`
		`large`
			Amazon Linux 2023	`aws/codebuild/ami/amazonlinux-arm-base:latest`	Ninguno

`ubuntu`	`5.0`	`small`
		`medium`
		`large`
		`xlarge`
		`2xlarge`
		`gpu_small`
		`gpu_large`
			Ubuntu 20.04	`aws/codebuild/standard:5.0`	ubuntu/standard/5.0

`ubuntu`	`6.0`		Ubuntu 22.04	`aws/codebuild/standard:6.0`	ubuntu/standard/6.0
`ubuntu`	`7.0`		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 Serve Core 2022	`aws/codebuild/windows-base:2022-1.0`	N/A
`windows`	`2.0`		Windows Server Core 2019	`aws/codebuild/windows-base:2019-2.0`	N/A

Tipo de entorno	Identificador de imagen	Tamaño de instancia	Plataforma	Imagen resuelta	Definición
`windows`	`3.0`		Windows Server Core 2019	`aws/codebuild/windows-base:2019-3.0`	N/A
`windows-ec2`	`2022`	`medium` `large`	Windows Server Core 2022	`aws/codebuild/ami/windows-base:2022`	Ninguno

Además, puede anular la configuración del entorno de Lambda mediante los siguientes valores. Para obtener más información sobre la computación CodeBuild Lambda, consulte. Ejecutar se basa en AWS Lambda cálculo CodeBuild admite las siguientes imágenes de procesamiento Lambda:

Tipo de entorno	Identificador de imagen	Tamaño de instancia			

`linux-lambda`
`arm-lambda`
	`dotnet6`
	`go1.21`
	`corretto11`
	`corretto17`
	`corretto21`
	`nodejs18`
	`nodejs20`
		`1GB`
		`2GB`
		`4GB`
		`8GB`
		`10GB`			

Tipo de entorno	Identificador de imagen	Tamaño de instancia			

	`python3.11`
	
	`python3.12`
	
	`ruby3.2`				

Para obtener más información, consulte Modos y tipos de computación del entorno de compilación y Imágenes de Docker proporcionadas por CodeBuild.

Self-managed GitLab se ejecuta en AWS CodeBuild

GitLab proporciona dos modos de ejecución para ejecutar las GitLab tareas en CI/CD proceso. Uno de los modos GitLab-hosted son los ejecutores, gestionados por ellos GitLab y totalmente integrados con ellos GitLab. El otro modo son los ejecutores autogestionados, que te permiten crear tu propio entorno personalizado para ejecutar las tareas en GitLab CI/CD proceso.

Los pasos generales para configurar un CodeBuild proyecto para ejecutar tareas GitLab CI/CD en proceso son los siguientes:

1. Si aún no lo has hecho, conéctate con una aplicación de OAuth a la que conectar tu proyecto. GitLab
2. Ve a la CodeBuild consola y crea un CodeBuild proyecto con un webhook y configura tus filtros de webhook.
3. Actualiza el YAML de tu GitLab CI/CD canalización GitLab para configurar tu entorno de compilación.

Para obtener un procedimiento más detallado, consulte Tutorial: Configurar un corredor CodeBuild-hosted GitLab.

Esta función permite que sus GitLab CI/CD trabajos en proceso se integren de forma nativa AWS, lo que proporciona seguridad y comodidad a través de funciones como IAM y Amazon VPC. AWS CloudTrail Puede acceder a los tipos de instancias más recientes, incluidas las ARM-based instancias.

Temas

- Acerca del CodeBuild-hosted GitLab corredor
- Tutorial: Configurar un corredor CodeBuild-hosted GitLab
- El corredor admite anulaciones de etiquetas CodeBuild-hosted GitLab
- Calcule las imágenes compatibles con el ejecutor CodeBuild-hosted GitLab

Acerca del CodeBuild-hosted GitLab corredor

Las siguientes son algunas preguntas frecuentes sobre el CodeBuild-hosted GitLab corredor.

¿Qué tipos de fuentes admiten los CodeBuild-hosted GitLab corredores?

CodeBuild-hosted GitLab los ejecutores son compatibles con el tipo `GITLAB_SELF_MANAGED` de fuente `GITLAB` y.

¿Cuándo debo incluir las anulaciones de instancias e imágenes 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 de GitLab CI/CD canalización. Esto se puede hacer sin necesidad de crear varios CodeBuild proyectos o webhooks.

¿Puedo usar CloudFormation ¿para esta función?

Sí, puedes incluir un grupo de filtros en tu CloudFormation plantilla que especifique un filtro de eventos de trabajo GitLab de flujo de trabajo en el webhook de tu proyecto.

```
Triggers:
  Webhook: true
  FilterGroups:
    - - Type: EVENT
        Pattern: WORKFLOW_JOB_QUEUED
```

Para obtener más información, consulte Filtrar eventos de GitLab webhook (CloudFormation).

Si necesitas ayuda para configurar las credenciales del proyecto en tu CloudFormation plantilla, consulta AWS::CodeBuild:: SourceCredential en la Guía del AWS CloudFormation usuario para obtener más información.

¿Cómo puedo enmascarar los secretos al usar esta característica?

De forma predeterminada, los secretos que se imprimen en el registro no están enmascarados. Si desea ocultar sus secretos, puede hacerlo actualizando la configuración de las variables de CI/CD entorno:

Para enmascarar secretos en GitLab

1. En tu GitLab configuración, elige CI/CD.
2. En Variables, elija Edit para el secreto que quiere enmascarar.
3. En Visibility, seleccione Mask variable y, a continuación, elija Update variable para guardar los cambios.

¿Puedo recibir eventos de GitLab webhook de varios proyectos de un mismo grupo?

CodeBuild admite webhooks grupales, que reciben eventos de un grupo específico GitLab . Para obtener más información, consulte GitLab webhooks grupales.

¿Puedo realizar un trabajo en el ejecutor de docker para el ejecutor autoadministrado? Por ejemplo, quiero ejecutar un trabajo de canalización en una imagen específica para mantener el mismo entorno de compilación en un contenedor separado y aislado.

Puedes ejecutar el ejecutor GitLab autogestionado CodeBuild con una imagen específica creando el proyecto con una imagen personalizada o anulando la imagen en tu archivo. `.gitlab-ci.yml`

¿Con qué ejecutor se ejecuta el ejecutor autogestionado? CodeBuild

El ejecutor autogestionado CodeBuild se ejecuta con el ejecutor del shell, donde la compilación se ejecuta localmente junto con el ejecutor GitLab que se ejecuta dentro del contenedor docker.

¿Puedo proporcionar comandos buildspec junto con el ejecutor autoadministrado?

Sí, se pueden agregar comandos buildspec junto con el ejecutor autoadministrado. Puedes incluir el archivo buildspec.yml en tu GitLab repositorio y usar la **`buildspec-override:true`** etiqueta de la sección Etiquetas del trabajo. Para obtener más información, consulte Nombre de archivo y ubicación de almacenamiento de buildspec.

¿Qué regiones admiten el uso de un corredor? CodeBuild-hosted GitLab

CodeBuild-hosted GitLab Los corredores son compatibles en todas CodeBuild las regiones. Para obtener más información sobre Regiones de AWS dónde CodeBuild está disponible, consulte AWS Servicios por región.

¿Qué plataformas admiten el uso de un CodeBuild-hosted GitLab corredor?

CodeBuild-hosted GitLab los ejecutores son compatibles tanto con Amazon EC2 como con Compute. AWS Lambda 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.

Tutorial: Configurar un corredor CodeBuild-hosted GitLab

Este tutorial le muestra cómo configurar sus CodeBuild proyectos para ejecutar trabajos GitLab CI/CD en proceso. Para obtener más información sobre el uso GitLab o la GitLab autogestión con CodeBuild ella, consulteSelf-managed GitLab se ejecuta en AWS CodeBuild.

Para completar este tutorial, primero debe:

- Conéctate con una aplicación OAuth mediante. AWS CodeConnections Ten en cuenta que cuando te conectes con una app de OAuth, debes usar la CodeBuild consola para hacerlo. Para obtener instrucciones adicionales, consulte GitLab acceder en CodeBuild.
- Conéctate CodeBuild a tu GitLab cuenta. Para ello, puedes añadirlo GitLab como proveedor de fuentes en la consola. Para obtener instrucciones, consulte GitLab acceder en CodeBuild.

> **Note**
>
> Esto solo es necesario si no te has conectado GitLab a tu cuenta.
> Con esta función, CodeBuild necesita permisos adicionales, como `create_runner` y `manage_runner` desde la aplicación GitLab OAuth. Si existen AWS CodeConnections para una GitLab cuenta en particular, no solicitará automáticamente las actualizaciones de permisos. Para ello, puedes ir a la AWS CodeConnections consola y crear una conexión ficticia con la misma GitLab cuenta para activar la reautorización y obtener los permisos adicionales. Tras completar este paso, todas las conexiones existentes podrán utilizar la función Runner. Una vez que se finalice, puede eliminar la conexión ficticia.

Paso 1: Crea un CodeBuild proyecto con un webhook

En este paso, crearás un CodeBuild proyecto con un webhook y lo revisarás en la GitLab 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 Tipo de proyecto, seleccione Proyecto de ejecutor.

   - En Ejecutor:
     - Para el proveedor Runner, elija GitLab.
     - En Credencial, elija una de las opciones siguientes:
       - Elija Credencial de origen predeterminada. La conexión predeterminada aplica una GitLab conexión predeterminada en todos los proyectos.
       - Elija Credencial de origen personalizada. La conexión personalizada aplica una GitLab conexión personalizada que anula la configuración predeterminada de tu cuenta.

     > **Note**
     >
     > Si aún no has creado una conexión con tu proveedor, tendrás que crear una nueva GitLab . Para obtener instrucciones, consulte Conectarse CodeBuild a GitLab.

     - En Ubicación del ejecutor, seleccione Repositorio.
     - En Repositorio, elija el nombre de su proyecto GitLab especificando la ruta del proyecto con el espacio de nombres.
   - En Environment (Entorno):
     - Elija una Imagen del entorno y una Computación compatibles. Ten en cuenta que tienes la opción de anular la configuración de la imagen y la instancia mediante una etiqueta en el código YAML de tu GitLab CI/CD canalización. Para obtener más información, consulte Paso 2: creación de un archivo .gitlab-ci.yml en el repositorio.
   - En Buildspec:
     - Tenga en cuenta que su especificación de compilación se ignorará a menos que se agregue `buildspec-override:true` como etiqueta. En su lugar, la CodeBuild anulará para usar comandos que configurarán el ejecutor autogestionado.

-

3. Continúe con los valores predeterminados y, a continuación, elija Crear el proyecto de compilación.
4. Abre la GitLab consola en `https://gitlab.com/`*`user-name`*`/`*`repository-name`*`/-/hooks` para comprobar que se ha creado un webhook y que está habilitado para entregar eventos de Workflow Jobs.

Paso 2: creación de un archivo .gitlab-ci.yml en el repositorio

En este paso, crearás un `.gitlab-ci.yml` archivo GitLabpara configurar tu entorno de compilación y utilizar ejecutores GitLab autogestionados en él. CodeBuild Para obtener más información, consulte Use self-managed runners.

Actualiza tu GitLab CI/CD canalización (YAML)

Desplácese a `https://gitlab.com/`*`user-name`*`/`*`project-name`*`/-/tree/`*`branch-name` y cree un archivo `.gitlab-ci.yml` en su repositorio. Para configurar el entorno de compilación, realice una de las siguientes acciones:

- Puedes especificar el nombre del CodeBuild proyecto, 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 GitLab trabajo 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.

  ```
  tags:
      - codebuild-<codebuild-project-name>-$CI_PROJECT_ID-$CI_PIPELINE_IID-$CI_JOB_NAME
  ```

  `$CI_PROJECT_ID-$CI_PIPELINE_IID-$CI_JOB_NAME`es necesario para asignar la compilación a tareas de canalización específicas y detener la compilación cuando se cancela la ejecución de la canalización.

> **Note**
>
> Asegúrate de que tu nombre *<project-name>* coincide con el del proyecto en el que lo creaste CodeBuild. Si no coincide, no CodeBuild procesará el webhook y es posible que la GitLab CI/CD canalización se bloquee.

El siguiente es un ejemplo de un GitLab CI/CD proceso YAML:

```
workflow:
  name: HelloWorld
stages:          # List of stages for jobs, and their order of execution
  - build

build-job:       # This job runs in the build stage, which runs first.
  stage: build
  script:
    - echo "Hello World!"
  tags:
    - codebuild-myProject-$CI_PROJECT_ID-$CI_PIPELINE_IID-$CI_JOB_NAME
```

- También puede anular el tipo de computación e imagen en la etiqueta. Consulte Calcule las imágenes compatibles con el ejecutor CodeBuild-hosted GitLab para ver una lista de imágenes seleccionadas. Para usar imágenes personalizadas, consulte El corredor admite anulaciones de etiquetas CodeBuild-hosted GitLab. El tipo de computación y la imagen de la etiqueta anularán la configuración del entorno en el proyecto. Si quiere anular la configuración del entorno para una compilación de computación de Amazon EC2, utilice la sintaxis siguiente:

```
tags:
    - codebuild-<codebuild-project-name>-$CI_PROJECT_ID-$CI_PIPELINE_IID-$CI_JOB_NAME
    - image:<environment-type>-<image-identifier>
    - instance-size:<instance-size>
```

El siguiente es un ejemplo de un YAML de GitLab CI/CD canalización:

```
stages:
  - build

build-job:
```

```
  stage: build
  script:
    - echo "Hello World!"
  tags:
    - codebuild-myProject-$CI_PROJECT_ID-$CI_PIPELINE_IID-$CI_JOB_NAME
    - image:arm-3.0
    - instance-size:small
```

- Puede anular la flota utilizada para su compilación en la etiqueta. Al hacerlo, se anularán los valores de flota configurados en el proyecto para usar la flota especificada. Para obtener más información, consulte Ejecución de compilaciones en flotas de capacidad reservada. Si quiere anular la configuración de la flota para una compilación de computación de Amazon EC2, utilice la siguiente sintaxis:

```
tags:
    - codebuild-<codebuild-project-name>-$CI_PROJECT_ID-$CI_PIPELINE_IID-$CI_JOB_NAME
    - fleet:<fleet-name>
```

  Para anular tanto la flota como la imagen utilizadas para la compilación, utilice la sintaxis siguiente:

```
tags:
    - codebuild-<codebuild-project-name>-$CI_PROJECT_ID-$CI_PIPELINE_IID-$CI_JOB_NAME
    - fleet:<fleet-name>
    - image:<environment-type>-<image-identifier>
```

  El siguiente es un ejemplo de un YAML de GitLab CI/CD canalización:

```
stages:
  - build

build-job:
  stage: build
  script:
    - echo "Hello World!"
  tags:
    - codebuild-myProject-$CI_PROJECT_ID-$CI_PIPELINE_IID-$CI_JOB_NAME
    - fleet:myFleet
    - image:arm-3.0
```

- Para ejecutar tus trabajos de GitLab CI/CD canalización en una imagen personalizada, puedes configurar una imagen personalizada en tu CodeBuild proyecto y evitar tener que incluir una

etiqueta de anulación de imagen. CodeBuild utilizará la imagen configurada en el proyecto si no se proporciona una etiqueta de reemplazo de imagen.

Cuando confirmes los cambios `.gitlab-ci.yml`, se activará una GitLab canalización y se `build-job` enviará una notificación de webhook para iniciar la compilación. CodeBuild

Ejecuta los comandos buildspec en las fases INSTALL, PRE_BUILD y POST_BUILD

De forma predeterminada, CodeBuild omite cualquier comando de buildspec al ejecutar una compilación autogestionada. GitLab Para ejecutar comandos buildspec durante la compilación, se puede agregar `buildspec-override:true` como sufijo a `tags`:

```
tags:
    - codebuild-<codebuild-project-name>-$CI_PROJECT_ID-$CI_PIPELINE_IID-$CI_JOB_NAME
    - buildspec-override:true
```

Al usar este comando, CodeBuild se creará una carpeta llamada carpeta de origen `gitlab-runner` principal del contenedor. Cuando el GitLab ejecutor se inicie durante la `BUILD` fase, se ejecutará en el `gitlab-runner` directorio.

Existen varias limitaciones a la hora de utilizar una modificación de las especificaciones de compilación en una compilación autogestionada: GitLab

- CodeBuild no ejecutará los comandos buildspec durante la fase, ya que el ejecutor autogestionado se ejecuta en la `BUILD` fase. `BUILD`
- CodeBuild no descargará ninguna fuente primaria o secundaria durante la fase. `DOWNLOAD_SOURCE` Si ha configurado un archivo buildspec, solo se descargará ese archivo del origen principal del proyecto.
- Si un comando de compilación falla en la `INSTALL` fase `PRE_BUILD` o, no CodeBuild se iniciará el ejecutor autogestionado y será necesario cancelar la tarea de GitLab CI/CD canalización manualmente.
- CodeBuild obtiene el token del corredor durante la `DOWNLOAD_SOURCE` fase, que tiene un tiempo de caducidad de una hora. Si tus `PRE_BUILD` `INSTALL` fases superan una hora, el token de corredor puede caducar antes de que comience el corredor GitLab autogestionado.

Paso 3: revisión de los resultados

Siempre que se ejecute una GitLab CI/CD canalización, CodeBuild recibirá los eventos de las tareas de CI/CD canalización a través del webhook. Para cada trabajo en CI/CD proceso, CodeBuild inicia una compilación para ejecutar un ejecutor efímero GitLab . El ejecutor es responsable de ejecutar un único trabajo de CI/CD canalización. Una vez que se complete el trabajo, el ejecutor y el proceso de compilación asociado finalizarán de forma inmediata.

Para ver los registros CI/CD de tus trabajos en proceso, navega hasta tu repositorio GitLab, selecciona Build, Jobs y, a continuación, elige el trabajo específico para el que quieres revisar los registros.

Puedes revisar las etiquetas solicitadas en el registro mientras el trabajo está esperando a que lo recoja una persona autogestionada CodeBuild.

Filtra los eventos de los GitLab webhooks (CloudFormation)

La siguiente YAML-formatted parte de una CloudFormation plantilla crea un grupo de filtros que activa una compilación cuando se evalúa como verdadera. El siguiente grupo de filtros especifica una solicitud de trabajo de GitLab CI/CD canalización con un nombre de CI/CD canalización que coincide con la expresión `\[CI-CodeBuild\]` regular.

```
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: CODEBUILD_DEFAULT_WEBHOOK_SOURCE_LOCATION
    Triggers:
      Webhook: true
      ScopeConfiguration:
        Name: group-name
        Scope: GITLAB_GROUP
      FilterGroups:
```

```
        - - Type: EVENT
            Pattern: WORKFLOW_JOB_QUEUED
          - Type: WORKFLOW_NAME
            Pattern: \[CI-CodeBuild\]
```

El corredor admite anulaciones de etiquetas CodeBuild-hosted GitLab

En tu versión de GitLab CI/CD YAML en proceso, puedes incluir una variedad de sustituciones de etiquetas que modifiquen tu compilación de ejecutores autogestionados. Se CodeBuild ignorarán todas las compilaciones que no reconozcas, pero no rechazarán tu solicitud de webhook. Por ejemplo, el siguiente YAML incluye anulaciones para la imagen, el tamaño de la instancia, la flota y la especificación de compilación:

```
workflow:
  name: HelloWorld
stages:
  - build

build-job:
  stage: build
  script:
    - echo "Hello World!"
  tags:
    - codebuild-myProject-$CI_PROJECT_ID-$CI_PIPELINE_IID-$CI_JOB_NAME
    - image:arm-3.0
    - instance-size:small
    - fleet:myFleet
    - buildspec-override:true
```

`codebuild-`*`<project-name>`*`-$CI_PROJECT_ID-$CI_PIPELINE_IID-$CI_JOB_NAME` (obligatorio)

- Ejemplo: `codebuild-myProject-$CI_PROJECT_ID-$CI_PIPELINE_IID-$CI_JOB_NAME`
- Necesario para todos los YAML de GitLab CI/CD canalización. *`<project name>`*debe ser igual al nombre del proyecto para el que está configurado el webhook del ejecutor autogestionado.

`image:`*`<environment-type>`*`-`*`<image-identifier>`*

- Ejemplo: `image:arm-3.0`

- Anula la imagen y el tipo de entorno utilizados al iniciar la compilación del ejecutor autoadministrado. Para obtener más información sobre los valores admitidos, consulte Calcule las imágenes compatibles con el ejecutor CodeBuild-hosted GitLab.
  - Para anular la imagen y el tipo de entorno utilizados con una imagen personalizada, utilice `image:custom-<environment-type>-<custom-image-identifier>`
  - Ejemplo: `image:custom-arm-public.ecr.aws/codebuild/amazonlinux-aarch64-standard:3.0`

> **Note**
>
> Si la imagen personalizada se encuentra en un registro privado, consulte Configuración de una credencial de registro privada para ejecutores autoalojados.

`instance-size:<instance-size>`

- Ejemplo: `instance-size:small`
- Anula el tipo de instancia utilizado al iniciar la compilación del ejecutor autoadministrado. Para obtener más información sobre los valores admitidos, consulte Calcule las imágenes compatibles con el ejecutor CodeBuild-hosted GitLab.

`fleet:<fleet-name>`

- Ejemplo: `fleet:myFleet`
- Anula la configuración de flota establecida en el proyecto para usar la flota especificada. Para obtener más información, consulte Ejecución de compilaciones en flotas de capacidad reservada.

`buildspec-override:<boolean>`

- Ejemplo: `buildspec-override:true`
- Permite que la compilación ejecute comandos buildspec en las fases `INSTALL`, `PRE_BUILD` y `POST_BUILD` si está establecido en `true`.

Calcule las imágenes compatibles con el ejecutor CodeBuild-hosted GitLab

En la etiqueta que configuróTutorial: Configurar un corredor CodeBuild-hosted GitLab, 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. Para obtener más información acerca de

Tipo de entorno	Identificador de imagen	Tamaño de instancia	Plataforma	Image	Definición

`linux`	`4.0`	`small`
		`medium`
		`large`
		`xlarge`
			Amazon Linux 2	`aws/codebuild/amazonlinux-x86_64-standard:4.0`	al/standard/4.0

`linux`	`5.0`	`2xlarge`
		`gpu_small`
		`gpu_large`
			Amazon Linux 2023	`aws/codebuild/amazonlinux-x86_64-standard:5.0`	al/standard/5.0

`linux-ec2`	`latest`	`small`
		`medium`
		`large`
			Amazon Linux 2023	`aws/codebuild/ami/amazonlinux-x86_64-base:latest`	Ninguno

`arm`	`2.0`	`small`
		`medium`
		`large`
		`xlarge`
			Amazon Linux 2	`aws/codebuild/amazonlinux-aarch64-standard:2.0`	al/aarch64/0standard/2.

Tipo de entorno	Identificador de imagen	Tamaño de instancia	Plataforma	Image	Definición
arm	3.0	2xlarge	Amazon Linux 2023	aws/codebuild/amazonlinux-aarch64-standard:3.0	al/aarch64/0standard/3.
arm-ec2	latest	small medium large	Amazon Linux 2023	aws/codebuild/ami/amazonlinux-arm-base:latest	Ninguno
ubuntu	5.0	small medium large xlarge 2xlarge gpu_small gpu_large	Ubuntu 20.04	aws/codebuild/standard:5.0	ubuntu/standard/5.0
ubuntu	6.0		Ubuntu 22.04	aws/codebuild/standard:6.0	ubuntu/standard/6.0
ubuntu	7.0		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

Tipo de entorno	Identificador de imagen	Tamaño de instancia	Plataforma	Image	Definición
			Windows Server Core 2022	`aws/codebuild/windows-base:2022-1.0`	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
`windows-ec2`	`2022`	`medium` `large`	Windows Server Core 2022	`aws/codebuild/ami/windows-base:2022`	Ninguno

Además, puede anular la configuración del entorno de Lambda mediante los siguientes valores. Para obtener más información sobre la computación CodeBuild Lambda, consulte. Ejecutar se basa en AWS Lambda cálculo CodeBuild admite las siguientes imágenes de procesamiento Lambda:

Tipo de entorno	Versión de runtime	Tamaño de instancia			

`linux-lambda`	`dotnet6`
	`go1.21`
	`corretto11`
		`1GB`
		`2GB`
		`4GB`
		`8GB`			

`arm-lambda`					

Tipo de entorno	Versión de runtime	Tamaño de instancia			

	`corretto17`
	
	`corretto21`
	
	`nodejs18`
	
	`nodejs20`
	
	`python3.11`
	
	`python3.12`
	
	`ruby3.2`
		`10GB`			

Para obtener más información, consulte Modos y tipos de computación del entorno de compilación y Imágenes de Docker proporcionadas por CodeBuild.

Self-managed BuildKite Runner en AWS CodeBuild

Puedes configurar tu proyecto para configurar ejecutores de Buildkite autohospedados en CodeBuild contenedores para procesar tus trabajos de Buildkite. Puedes hacerlo configurando un webhook con tu CodeBuild proyecto y actualizando los pasos de YAML de Buildkite Pipeline para usar ejecutores autohospedados alojados en máquinas. CodeBuild

Los pasos generales para configurar un CodeBuild proyecto para ejecutar tareas de Buildkite son los siguientes:

- Ve a la CodeBuild consola y crea un CodeBuild proyecto con la configuración del tipo ejecutor de proyectos de Buildkite
- Agregue un webhook de `job.scheduled` a su organización de Buildkite.
- Actualice los pasos de YAML de canalización de Buildkite en Buildkite para configurar el entorno de compilación.

Para obtener un procedimiento más detallado, consulte Tutorial: Configurar un ejecutor de CodeBuild-hosted Buildkite. Esta función permite que sus trabajos de Buildkite se integren de forma nativa AWS, lo que proporciona seguridad y comodidad a través de funciones como IAM AWS Secrets Manager, AWS CloudTrail y Amazon VPC. Puede acceder a los tipos de instancias más recientes, incluidas las instancias. ARM-based

Acerca del ejecutor CodeBuild-hosted Buildkite

Las siguientes son algunas de las preguntas más frecuentes sobre el CodeBuild-hosted Buildkite Runner.

¿Cuándo debo incluir las anulaciones de instancias e imágenes en la etiqueta?

Puede incluir las anulaciones de instancias e imágenes en la etiqueta a fin de especificar distintos entornos de compilación para cada uno de sus trabajos de Buildkite. Esto se puede hacer sin necesidad de crear varios CodeBuild proyectos o webhooks. Por ejemplo, es útil cuando necesita usar una matriz para trabajos de Buildkite.

```
agents:
  queue: "myQueue"
steps:
  - command: "echo \"Hello World\""
    agents:
      project: "codebuild-myProject"
      image: "{{matrix.os}}"
      instance-size: "{{matrix.size}}"
    matrix:
      setup:
        os:
          - "arm-3.0"
          - "al2-5.0"
        size:
          - "small"
          - "large"
```

¿Puedo CodeBuild crear webhooks en Buildkite automáticamente?

Actualmente, Buildkite requiere que todos los webhooks se creen manualmente usando su consola. Puede seguir el tutorial en Tutorial: Configurar un ejecutor de CodeBuild-hosted Buildkite para crear un webhook de Buildkite manualmente en la consola de Buildkite.

¿Puedo usar CloudFormation para crear webhooks de Buildkite?

CloudFormation actualmente no es compatible con los webhooks de Buildkite Runner, ya que Buildkite requiere que los webhooks se creen manualmente desde su consola.

¿Qué regiones admiten el uso de un corredor de Buildkite? CodeBuild-hosted

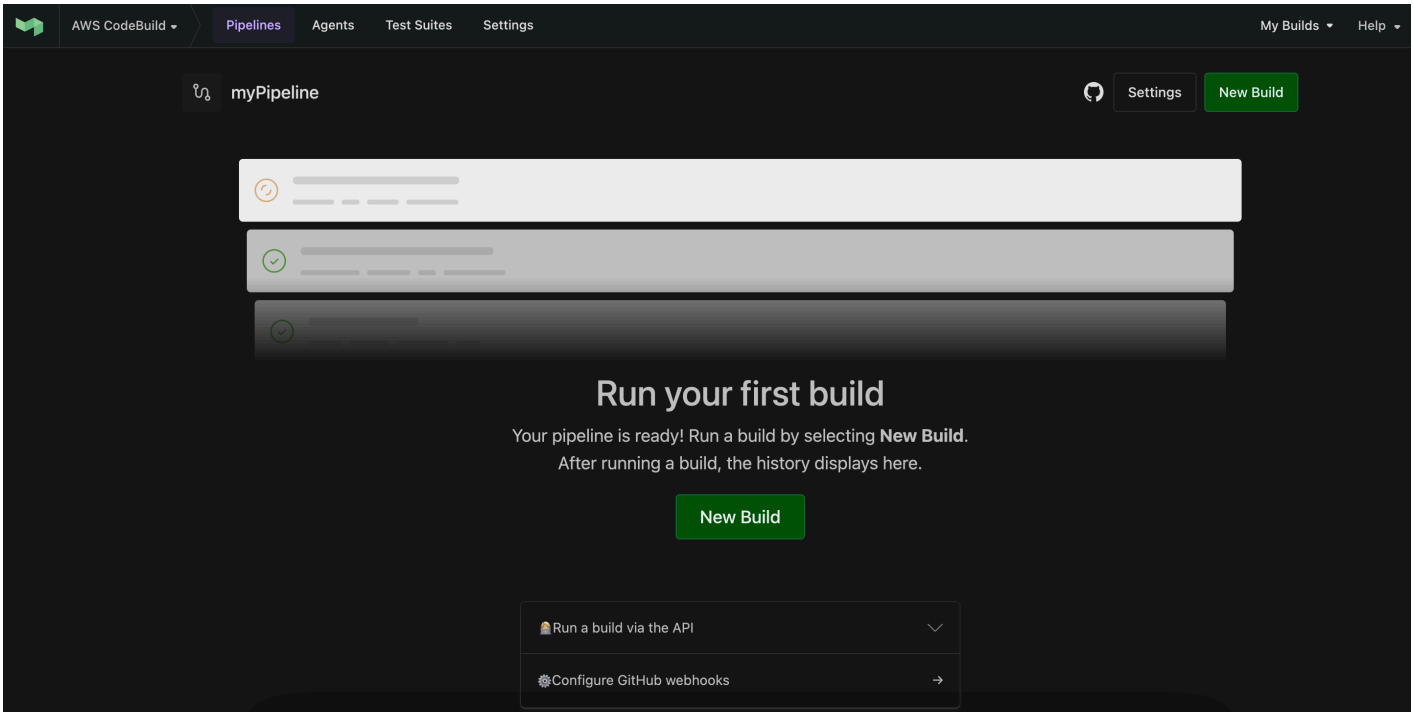
CodeBuild-hosted Los corredores de Buildkite son compatibles en todas las regiones. CodeBuild Para obtener más información sobre AWS las regiones en las que CodeBuild está disponible, consulta AWS Servicios por región.

Tutorial: Configurar un ejecutor de CodeBuild-hosted Buildkite

Este tutorial le muestra cómo configurar sus CodeBuild proyectos para ejecutar trabajos de Buildkite. Para obtener más información sobre el uso de Buildkite con, consulte. CodeBuild Self-managed BuildKite Runner en AWS CodeBuild

Para completar este tutorial, primero debe:

- Tener acceso a una organización de Buildkite. Para obtener más información sobre cómo configurar una cuenta y una organización de Buildkite, puede seguir este Tutorial de introducción.
- Cree una canalización, un clúster y una cola de Buildkite configurados para usar ejecutores autoalojados. Para obtener más información sobre la configuración de estos recursos, puede consultar el Tutorial de configuración de canalización de Buildkite.

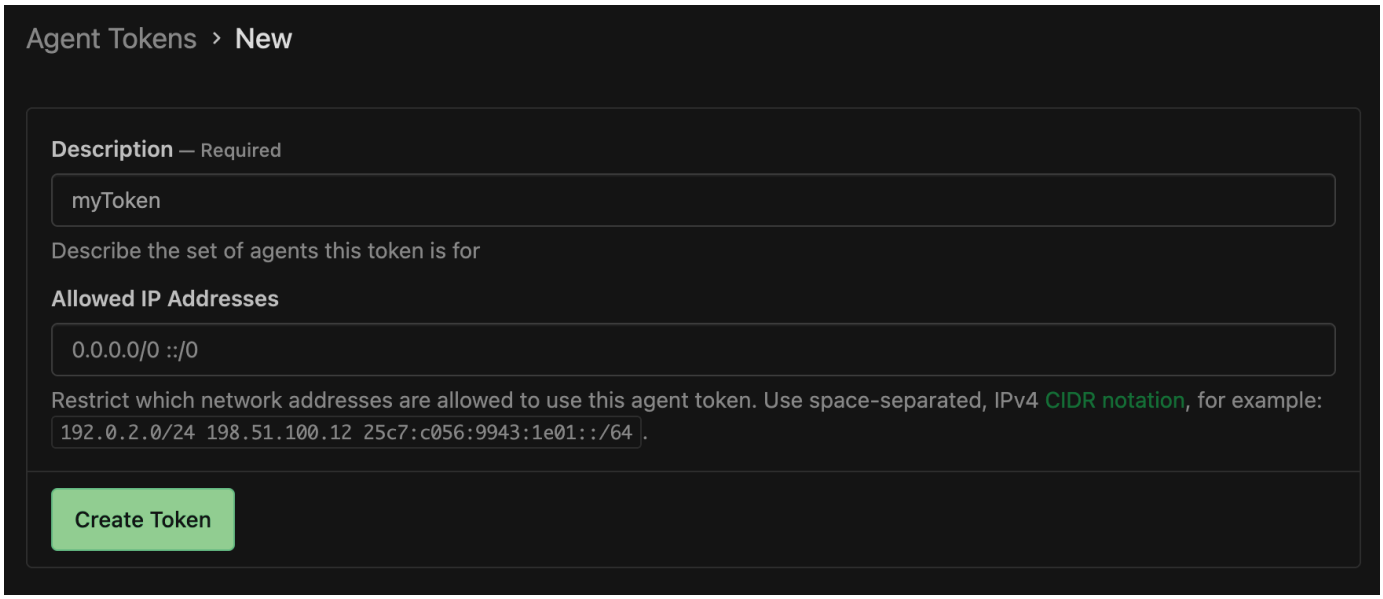


Paso 1: generación de un token de agente de Buildkite

En este paso, generarás un token de agente en Buildkite que se utilizará para autenticar a los corredores autohospedados. CodeBuild Para obtener más información sobre este recurso, consulte Buildkite Agent Tokens.

Generación de un token de agente de Buildkite

1. En el clúster de Buildkite, elija Tokens de agente y, a continuación, Nuevo token.
2. Agregue una descripción al token y haga clic en Crear token.
3. Guarda el valor del token del agente, ya que se utilizará más adelante durante la configuración del proyecto. CodeBuild



Paso 2: Crea un CodeBuild proyecto con un webhook

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 autoalojado. Para obtener más información, consulte Creación de un proyecto de compilación (consola) y Ejecutar una compilación (consola).
   - En Project Configurator, seleccione Proyecto de ejecutor. En Ejecutor:
     - En Proveedor de ejecutor, elija Buildkite.

- En Token de agente de Buildkite, seleccione Crear un nuevo token de agente mediante la página de creación de secreto. Se te pedirá que crees un nuevo secreto AWS Secrets Manager con un valor secreto igual al token de agente de Buildkite que generaste anteriormente.
- (Opcional) Si quieres usar credenciales CodeBuild administradas para tu trabajo, selecciona el proveedor del repositorio de origen de tu trabajo en las opciones de credenciales de origen de Buildkite y verifica que las credenciales estén configuradas para tu cuenta. Además, verifique que la canalización de Buildkite utilice Extracción mediante HTTPS.

> **Note**
>
> Buildkite requiere credenciales de origen en el entorno de compilación para poder utilizar la fuente para su trabajo. Consulte en Autenticación de Buildkite en un repositorio privado las opciones de credenciales de código fuente disponibles.

- (Opcional) En Entorno:
  - Elija una Imagen del entorno y una Computación compatibles.

    Tenga en cuenta que tiene la opción de anular la configuración de imagen e instancia mediante una etiqueta en los pasos de YAML de Buildkite. Para obtener más información, consulte Paso 4: actualización de los pasos de la canalización de Buildkite.
- (Opcional) En Buildspec:
  - La buildspec se ignorará de forma predeterminada a menos que se agregue `buildspec-override: "true"` como etiqueta. En su lugar, lo CodeBuild anulará para usar comandos que configurarán el ejecutor autohospedado.

> **Note**
>
> CodeBuild no admite archivos de especificaciones de compilación para compilaciones de ejecutores autohospedados por Buildkite. Para las especificaciones de compilación integradas, tendrás que habilitar git-credential-helper en tu especificación de compilación si has configurado las credenciales de fuente gestionadas CodeBuild

3. Continúe con los valores predeterminados y, a continuación, elija Crear el proyecto de compilación.

4. Guarde los valores de URL de carga útil y Secreto de la ventana emergente Crear webhook. Siga las instrucciones de la ventana emergente para crear un nuevo webhook de organización de Buildkite o continúe con la siguiente sección.

Paso 3: Crea un webhook en Buildkite CodeBuild

En este paso, utilizarás la URL de carga útil y los valores secretos del CodeBuild webhook para crear un nuevo webhook en Buildkite. Este webhook se usará para activar las compilaciones internas CodeBuild cuando se inicie un trabajo válido de Buildkite.

Cómo crear un nuevo webhook en Buildkite

1. Abre la página de configuración de tu organización de Buildkite.
2. En Integraciones, seleccione Servicios de notificación.
3. Seleccione Agregar junto a la casilla Webhook. En la página Agregar notificación de webhook, use la siguiente configuración:
   a. En URL de webhook, agregue el valor de URL de carga útil guardado.
   b. En Token, verifica que esté seleccionada la opción Enviar el token como X-Buildkite-Token. Agregue el valor de Secreto de webhook al campo Token.
   c. En, compruebe que esté seleccionada la opción Enviar el token como X-Buildkite-Token. Agregue el valor de Secreto de webhook al campo Token.
   d. En Eventos, seleccione el evento del webhook `job.scheduled`.
   e. (Opcional) En Canalizaciones, puede elegir opcionalmente desencadenar solo las compilaciones de una canalización específica.
4. Seleccione Agregar notificación de webhook.

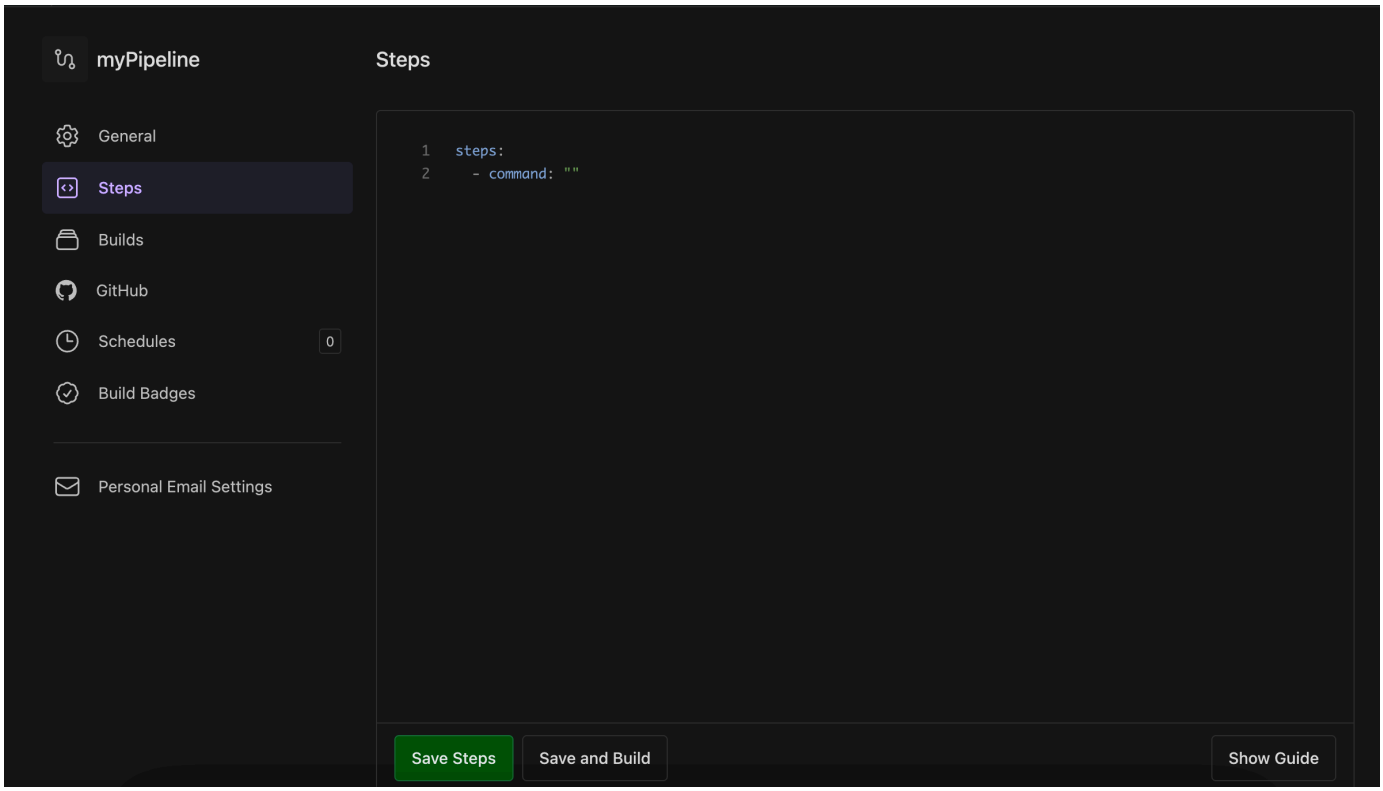
Paso 4: actualización de los pasos de la canalización de Buildkite

En este paso, actualiza los pasos de tu canalización de Buildkite para añadir las etiquetas necesarias y las anulaciones opcionales. Para ver la lista completa de las anulaciones de etiquetas admitidas, consulte El ejecutor Buildkite admite anulaciones de etiquetas CodeBuild-hosted.

Actualización de los pasos de la canalización

1. Abre la página de pasos de la canalización de Buildkite. Para ello, selecciona la canalización de Buildkite, selecciona Configuración y, a continuación, selecciona Pasos.

Si aún no lo has hecho, selecciona Convertir a pasos de YAML.



2. Como mínimo, tendrás que especificar una etiqueta de agente de Buildkite que haga referencia al nombre de tu canalización. CodeBuild El nombre del proyecto es necesario para vincular los ajustes AWS relacionados con tu trabajo de Buildkite a un proyecto específico. CodeBuild Al incluir el nombre del proyecto en el YAML, CodeBuild se pueden invocar trabajos con la configuración de proyecto correcta.

```
agents:
  project: "codebuild-<project name>"
```

A continuación, se muestra un ejemplo de pasos de canalización de Buildkite solo con la etiqueta del proyecto:

```
agents:
  project: "codebuild-myProject"
steps:
  - command: "echo \"Hello World\""
```

También puede anular el tipo de computación e imagen en la etiqueta. Consulte Calcule imágenes compatibles con el ejecutor CodeBuild-hosted Buildkite para obtener una lista

de las imágenes disponibles. El tipo de computación y la imagen de la etiqueta anularán la configuración del entorno en el proyecto. Para anular la configuración de su entorno para una compilación de procesamiento CodeBuild EC2 o Lambda, utilice la siguiente sintaxis:

```
agents:
  project: "codebuild-<project name>"
  image: "<environment-type>-<image-identifier>"
  instance-size: "<instance-size>"
```

A continuación, se muestra un ejemplo de pasos de canalización de Buildkite con anulaciones de tamaño de imagen e instancia:

```
agents:
  project: "codebuild-myProject"
  image: "arm-3.0"
  instance-size: "small"
steps:
  - command: "echo \"Hello World\""
```

Puede anular la flota utilizada para su compilación en la etiqueta. Al hacerlo, se anularán los valores de flota configurados en el proyecto para usar la flota especificada. Para obtener más información, consulte Ejecución de compilaciones en flotas de capacidad reservada.

Si quiere anular la configuración de la flota para una compilación de computación de Amazon EC2, utilice la siguiente sintaxis:

```
agents:
  project: "codebuild-<project name>"
  fleet: "<fleet-name>"
```

Para anular tanto la flota como la imagen utilizadas para la compilación, utilice la sintaxis siguiente:

```
agents:
  project: "codebuild-<project name>"
  fleet: "<fleet-name>"
  image: "<environment-type>-<image-identifier>"
```

A continuación, se muestra un ejemplo de pasos de canalización de Buildkite con anulaciones de flota e imagen:

```
agents:
  project: "codebuild-myProject"
  fleet: "myFleet"
  image: "arm-3.0"
steps:
  - command: "echo \"Hello World\""
```

3. Puede optar por ejecutar comandos buildspec integrados durante la compilación del ejecutor autoalojado en Buildkite (consulte Ejecución de comandos buildspec durante las fases INSTALL, PRE_BUILD y POST_BUILD para obtener más información). Para especificar que la CodeBuild compilación debe ejecutar los comandos buildspec durante la compilación del ejecutor autohospedado de Buildkite, utilice la siguiente sintaxis:

```
agents:
  project: "codebuild-<project name>"
  buildspec-override: "true"
```

A continuación, se muestra un ejemplo de canalización de Buildkite con anulaciones de buildspec:

```
agents:
  project: "codebuild-myProject"
  buildspec-override: "true"
steps:
  - command: "echo \"Hello World\""
```

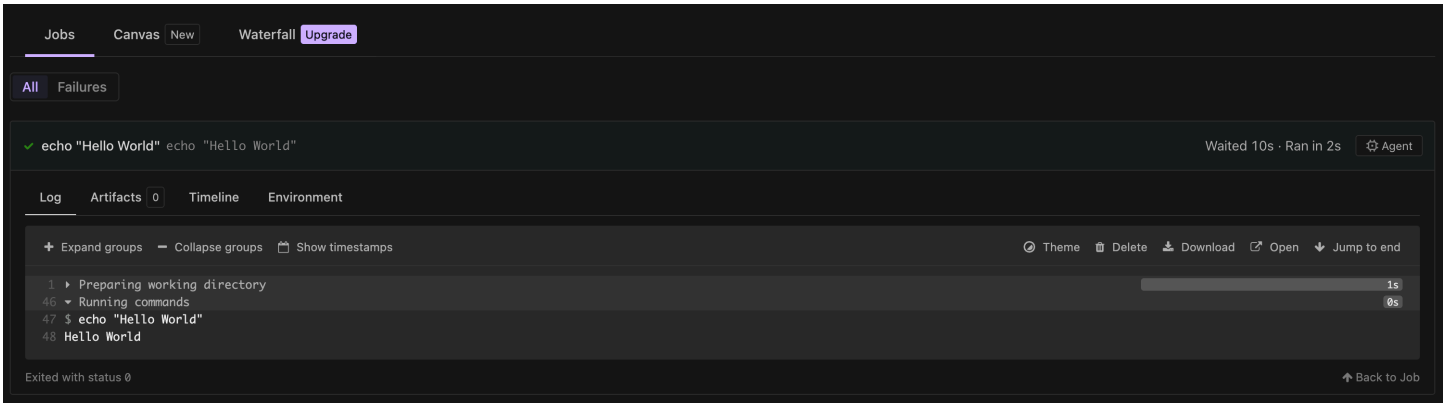
4. De forma opcional, puedes proporcionar etiquetas distintas de las compatibles. CodeBuild Estas etiquetas se ignorarán para el fin de anular los atributos de la compilación, pero no provocarán un error de la solicitud de webhook. Por ejemplo, agregar `myLabel: "testLabel"` como etiqueta no impedirá que la compilación se ejecute.

Paso 5: Revisa los resultados

Cada vez que se inicie un trabajo de Buildkite en tu cartera, CodeBuild recibirás un evento de webhook a través del `job.scheduled` webhook de Buildkite. Para cada trabajo de tu compilación de Buildkite, se CodeBuild iniciará una compilación para ejecutar un corredor efímero de Buildkite.

El ejecutor es responsable de ejecutar un único trabajo de Buildkite. Una vez que se complete el trabajo, el ejecutor y el proceso de compilación asociado finalizarán de forma inmediata.

Para ver los registros de trabajos de flujo de trabajo, navegue hasta la canalización de Buildkite y seleccione la compilación más reciente (para desencadenar una nueva compilación, seleccione Nueva compilación). Una vez que la CodeBuild compilación asociada a cada uno de tus trabajos comience y recoja el trabajo, deberías ver los registros del trabajo en la consola de Buildkite



Autenticación de Buildkite en un repositorio privado

Si tiene un repositorio privado configurado en la canalización de Buildkite, Buildkite requiere permisos adicionales en el entorno de compilación para extraer el repositorio, ya que Buildkite no vende credenciales a ejecutores autoalojados para que las extraigan de repositorios privados. Para autenticar el agente de ejecutor autoalojado en Buildkite en el repositorio de código fuente privado externo, puede usar una de las siguientes opciones.

Para autenticarte con CodeBuild

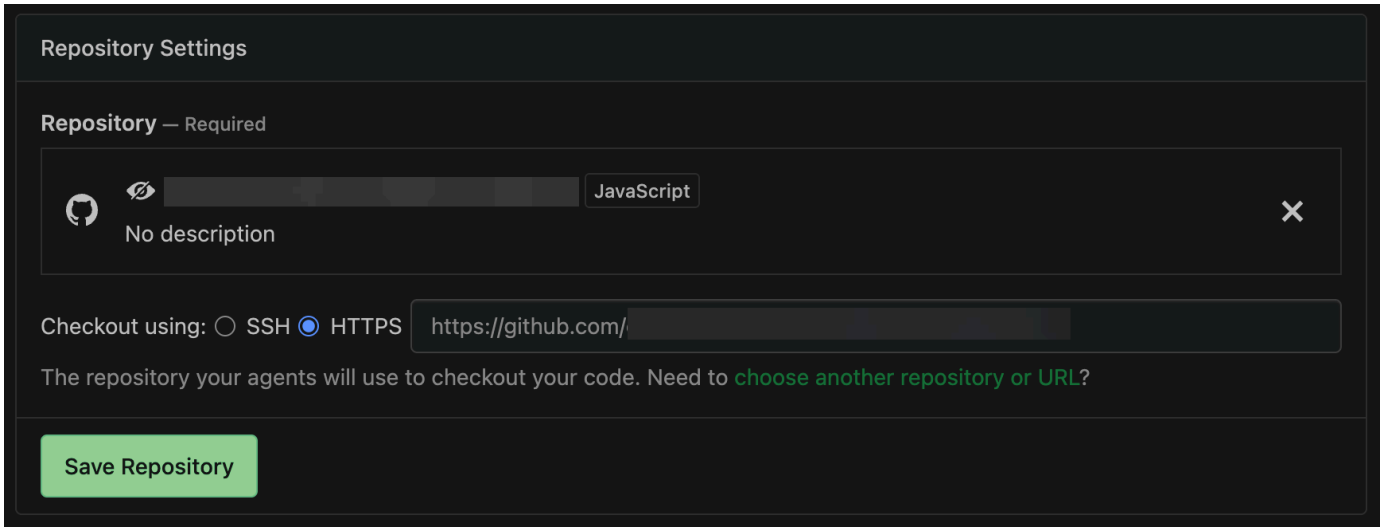
CodeBuild ofrece la gestión de credenciales gestionadas para los tipos de fuentes compatibles. Para usar las credenciales de CodeBuild origen para acceder al repositorio de fuentes de su trabajo, puede seguir los siguientes pasos:

1. En la CodeBuild consola, vaya a Editar proyecto o cree un CodeBuild proyecto nuevo siguiendo los pasos que se indicanPaso 2: Crea un CodeBuild proyecto con un webhook.
2. En las opciones de credenciales de origen de Buildkite, selecciona el proveedor de repositorios de origen de tu trabajo.

    1. Si quieres usar CodeBuild credenciales a nivel de cuenta, comprueba que estén configuradas correctamente. Además, si el proyecto tiene una buildspec integrada configurada, compruebe que git-credential-helper esté habilitado.

2. Si desea usar credenciales a nivel de proyecto, seleccione Usar CodeBuild credenciales de anulación solo para este proyecto y configure las credenciales para su proyecto.

3. En la configuración de la canalización de Buildkite, vaya a Configuración del repositorio. Establezca la configuración de extracción del repositorio de código fuente en Extracción mediante HTTPS



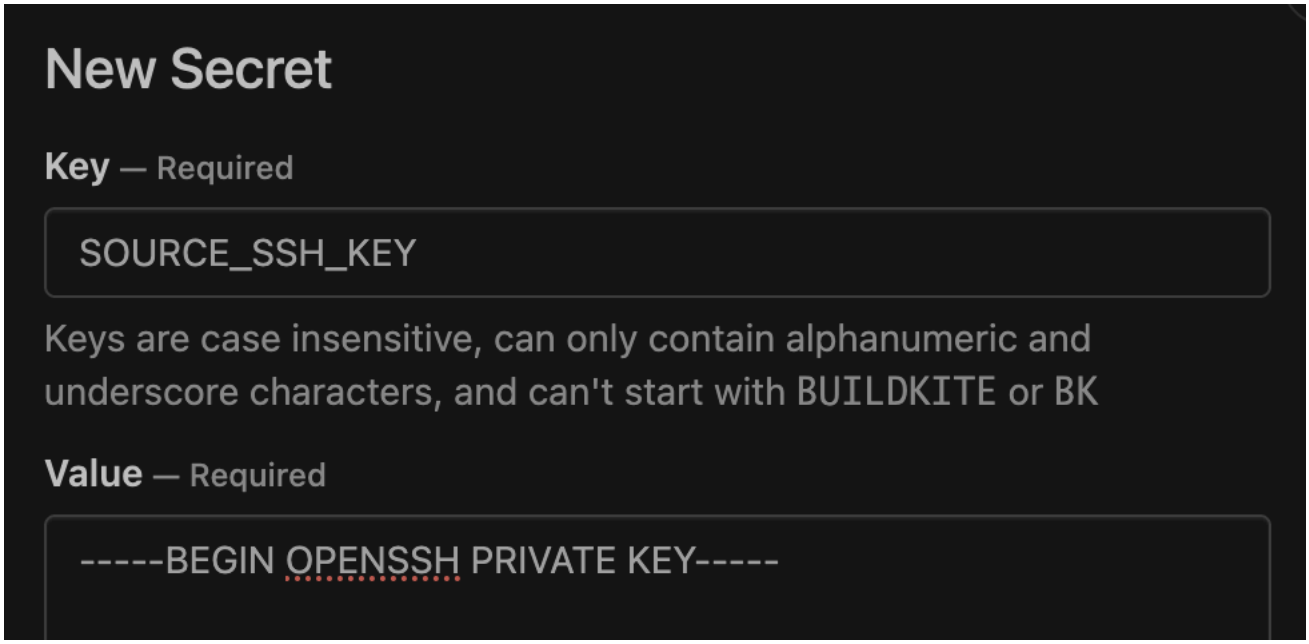
Cómo autenticarse con secretos de Buildkite

Buildkite mantiene un complemento ssh-checkout que se puede utilizar para autenticar el ejecutor autoalojado en un repositorio de código fuente externo mediante una clave ssh. El valor de clave se guarda como Secreto de Buildkite y el agente de ejecutor autoalojado en Buildkite lo recupera automáticamente al intentar acceder a un repositorio privado. Para configurar el complemento ssh-checkout para la canalización de Buildkite, puede seguir estos pasos:

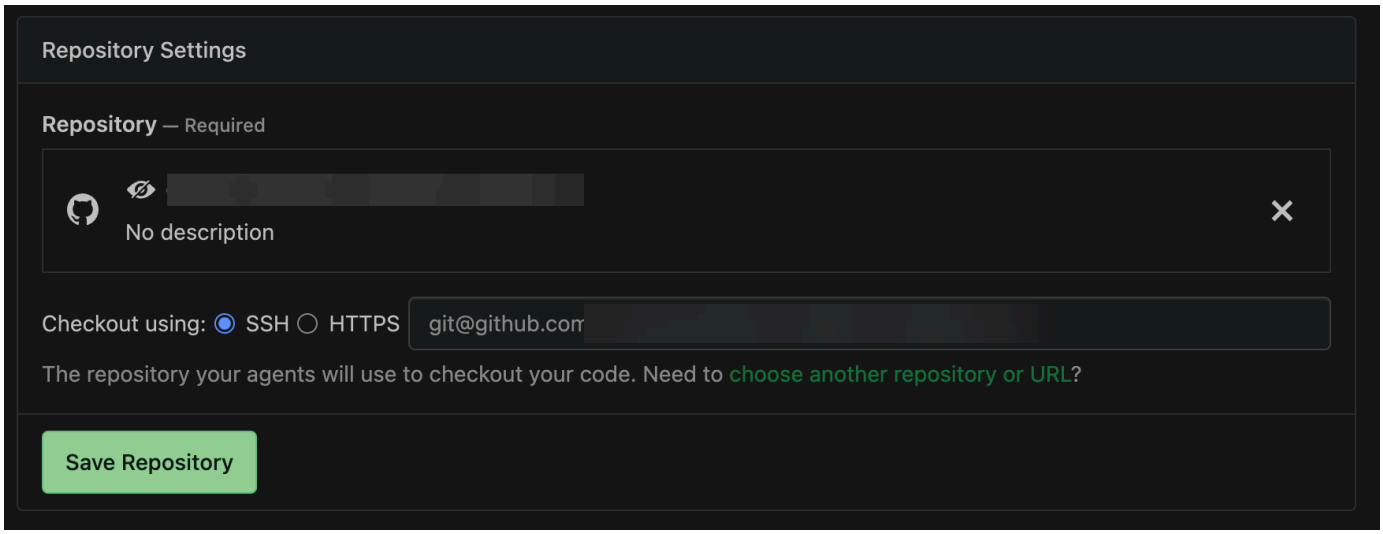
1. Genere una clave ssh privada y pública con su dirección de correo electrónico, p. ej. `ssh-keygen -t rsa -b 4096 -C "myEmail@address.com"`

2. Agregue la clave pública al repositorio de código fuente privado. Por ejemplo, puedes seguir esta guía para añadir una clave a una GitHub cuenta.

3. Agregue una nueva clave SSH secreta al clúster de Buildkite. En el clúster de Buildkite, seleccione Secretos → Nuevo secreto. Agregue un nombre para el secreto en el campo Clave y agregue la clave SSH privada al campo Valor:



4. En la canalización de Buildkite, navegue hasta la configuración del repositorio y configure el proceso de extracción para que use SSH.



5. Actualice los pasos de YAML de la canalización para usar el complemento `git-ssh-checkout`. Por ejemplo, el siguiente archivo YAML de canalización utiliza la acción de extracción con la clave del secreto de Buildkite anterior:

```
agents:
  project: "codebuild-myProject"
steps:
  - command: "npm run build"
    plugins:
      - git-ssh-checkout#v0.4.1:
```

```
            ssh-secret-key-name: 'SOURCE_SSH_KEY'
```

6. Al ejecutar un trabajo de ejecución autohospedado en Buildkite CodeBuild, Buildkite ahora usará automáticamente el valor secreto que hayas configurado al acceder a tu repositorio privado

Opciones de configuración del ejecutor

Puede especificar las siguientes variables de entorno en la configuración del proyecto para modificar la configuración de instalación de los ejecutores autoalojados:

- `CODEBUILD_CONFIG_BUILDKITE_AGENT_TOKEN`: CodeBuild obtendrá el valor secreto configurado como el valor de esta variable de entorno para registrar el agente de ejecución autohospedado AWS Secrets Manager de Buildkite. Esta variable de entorno debe ser de tipo `SECRETS_MANAGER` y el valor debe ser el nombre del secreto en Secrets Manager. Se requiere una variable de entorno de token de agente de Buildkite para todos los proyectos de ejecutor de Buildkite.
- `CODEBUILD_CONFIG_BUILDKITE_CREDENTIAL_DISABLE`: De forma predeterminada, CodeBuild cargará las credenciales de origen a nivel de cuenta o proyecto en el entorno de compilación, ya que el agente de Buildkite utiliza estas credenciales para acceder al repositorio fuente de la tarea. Para deshabilitar este comportamiento, puede agregar esta variable de entorno al proyecto con el valor establecido en `true`, lo que evitará que las credenciales de código fuente se carguen en el entorno de compilación.

Ejecución de comandos buildspec durante las fases INSTALL, PRE_BUILD y POST_BUILD

De forma predeterminada, CodeBuild ignora cualquier comando de buildspec al ejecutar una compilación autohospedada de Buildkite Runner. Para ejecutar comandos buildspec durante la compilación,

```
buildspec-override: "true"
```

se puede agregar como sufijo a la etiqueta:

```
agents:
  project: "codebuild-<project name>"
  buildspec-override: "true"
```

Al usar este comando, CodeBuild creará una carpeta llamada `buildkite-runner` carpeta de origen principal del contenedor. Cuando el ejecutor de Buildkite se inicia durante la fase `BUILD`, el ejecutor se ejecutará en el directorio `buildkite-runner`.

Existen varias limitaciones al usar una anulación de buildspec en una compilación de Buildkite autoalojada:

- El agente Buildkite requiere que las credenciales de origen existan en el entorno de compilación para acceder al repositorio de fuentes de la tarea. Si utilizas las credenciales CodeBuild de origen para la autenticación, debes habilitarlas `git-credential-helper` en tu buildspec. Por ejemplo, puede usar la siguiente builspec para habilitar `git-credential-helper` en las compilaciones de Buildkite:

  ```
  version: 0.2
  env:
    git-credential-helper: yes
  phases:
    pre_build:
      commands:
         - echo "Hello World"
  ```

- CodeBuild no ejecutará los comandos buildspec durante la `BUILD` fase, ya que el ejecutor autohospedado se ejecuta en la fase. `BUILD`
- CodeBuild no admite los archivos buildspec para las compilaciones de Buildkite Runner. Solo se admiten las especificaciones de compilación en línea para los ejecutores autohospedados de Buildkite
- Si un comando de compilación falla en la `INSTALL` fase `PRE_BUILD` o, no CodeBuild se iniciará el ejecutor autohospedado y será necesario cancelar la tarea de Buildkite manualmente.

Configuración de un ejecutor de Buildkite mediante programación

Para configurar un proyecto de ejecución de Buildkite mediante programación, debes configurar los siguientes recursos:

Cómo crear un ejecutor de Buildkite mediante programación

1. Cree un token de agente de Buildkite y guárdelo en texto sin formato en AWS Secrets Manager.
2. Configura un CodeBuild proyecto con la configuración que prefieras. Deberá configurar los siguientes atributos adicionales:

1. Un valor de entorno con nombre `CODEBUILD_CONFIG_BUILDKITE_AGENT_TOKEN`, tipo `SECRETS_MANAGER` y un valor igual al token de agente de Buildkite asociado al clúster de Buildkite.
2. Tipo de código fuente igual a `NO_SOURCE`
3. Permisos para acceder al secreto creado en el paso 1 en la función de servicio de tu proyecto

Por ejemplo, puede usar el siguiente comando para crear un proyecto de ejecutor de Buildkite válido a través de la CLI:

```
aws codebuild create-project \
--name buildkite-runner-project \
--source "{\"type\": \"NO_SOURCE\",\"buildspec\":\"\"}" \
--environment "{\"image\":\"aws/codebuild/amazonlinux-x86_64-standard:5.0\",
\"type\":\"LINUX_CONTAINER\",\"computeType\":\"BUILD_GENERAL1_MEDIUM\",
\"environmentVariables\":[{\"name\":\"CODEBUILD_CONFIG_BUILDKITE_AGENT_TOKEN\",
\"type\":\"SECRETS_MANAGER\",\"value\":\"<buildkite-secret-name>\"}]}" \
--artifacts "{\"type\": \"NO_ARTIFACTS\"}" \
--service-role <service-role>
```

3. Cree un webhook de ejecutor de Buildkite en el proyecto creado en el paso 2. Deberá usar las siguientes opciones de configuración al crear el webhook:

   1. build-type debe ser igual `RUNNER_BUILDKITE_BUILD`
   2. Un filtro con un tipo `EVENT` y un patrón igual a `WORKFLOW_JOB_QUEUED`

   Por ejemplo, puede usar el siguiente comando para crear un proyecto de ejecutor de webhook válido a través de la CLI:

```
aws codebuild create-webhook \
--project-name buildkite-runner-project \
--filter-groups "[[{\"type\":\"EVENT\",\"pattern\":\"WORKFLOW_JOB_QUEUED\"}]]" \
--build-type RUNNER_BUILDKITE_BUILD
```

4. Guarde los valores de URL de carga útil y Secreto devueltos por la llamada `create-webhook` y use las credenciales para crear un webhook en la consola de Buildkite. Puedes consultar el paso 3: Crear un CodeBuild webhook en Buildkite Tutorial: Configurar un ejecutor de CodeBuild-hosted Buildkite para obtener una guía sobre cómo configurar este recurso.

Solución de problemas del webhook en caso de compilaciones fallidas o un trabajo pendientes

Problema:

El webhook que ha configurado en Tutorial: Configurar un ejecutor de CodeBuild-hosted Buildkite no funciona o el trabajo del flujo de trabajo está pendiente en Buildkite.

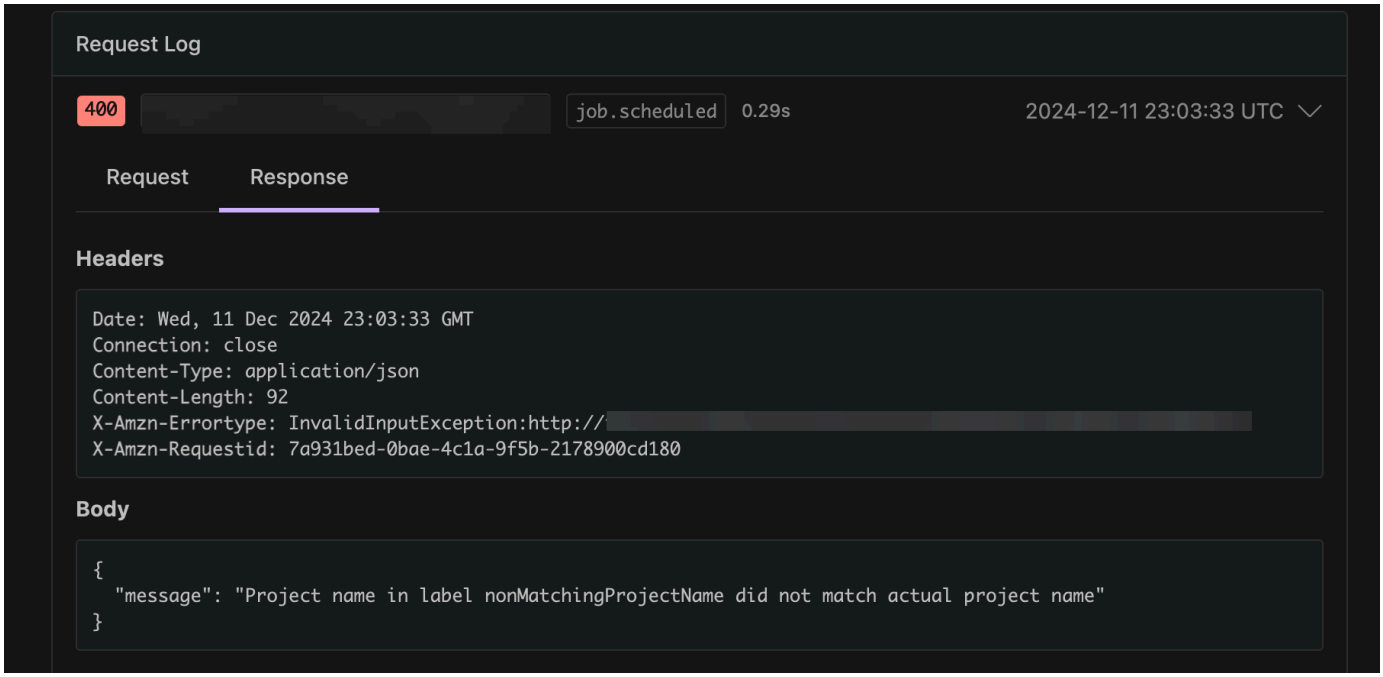
Posibles causas:

- Puede que el evento job.scheduled del webhook no pueda desencadenar una compilación. Revise los registros de Respuesta para ver la respuesta o el mensaje de error.
- Tu CodeBuild compilación falla antes de iniciar el agente de ejecución autohospedado de Buildkite para que se encargue de tu trabajo.

Soluciones recomendadas:

Cómo depurar eventos de webhook de Buildkite fallidos:

1. En la configuración de tu organización de Buildkite, ve a los Servicios de Notificación, selecciona tu CodeBuild webhook y, a continuación, busca el registro de solicitudes.
2. Busque el evento de webhook `job.scheduled` asociado al trabajo de Buildkite bloqueado. Puede usar el campo de ID de trabajo en la carga útil del webhook para correlacionar el evento del webhook con el trabajo de Buildkite.
3. Seleccione la pestaña Respuesta y compruebe el cuerpo de la respuesta. Comprueba que el código de estado de la respuesta sea `200` y que el cuerpo de la respuesta no contenga ningún mensaje inesperado.



Solución de problemas de permisos del webhook

Problema:

La tarea de Buildkite no puede acceder al repositorio de código fuente del trabajo debido a problemas con los permisos.

Causas posibles:

- CodeBuild no tiene permisos suficientes para consultar el repositorio de origen del trabajo.
- La configuración del repositorio de la canalización está configurada para que las credenciales CodeBuild administradas se comprueben mediante SSH.

Soluciones recomendadas:

- Compruebe que CodeBuild tiene configurados los permisos suficientes para consultar el repositorio de origen del trabajo. Además, compruebe que el rol de servicio de su CodeBuild proyecto tenga permisos suficientes para acceder a la opción de permiso de origen configurada.
- Comprueba que tu canalización de Buildkite esté configurada para utilizar el proceso de pago mediante HTTPS si utilizas credenciales de repositorio de fuentes CodeBuild gestionadas.

El ejecutor Buildkite admite anulaciones de etiquetas CodeBuild-hosted

En las etiquetas de agente de los pasos de canalización de Buildkite, puede proporcionar una variedad de anulaciones de etiquetas que modifiquen la compilación del ejecutor autoalojado. Se CodeBuild ignorarán todas las compilaciones que no reconozcas, pero no rechazarán tu solicitud de webhook. Por ejemplo, el siguiente YAML de flujo de trabajo incluye anulaciones para imagen, tamaño de instancia, flota y buildspec:

```
agents:
  queue: "myQueue"
steps:
  - command: "echo \"Hello World\""
    agents:
      project: "codebuild-myProject"
      image: "{{matrix.os}}"
      instance-size: "{{matrix.size}}"
      buildspec-override: "true"
    matrix:
      setup:
        os:
          - "arm-3.0"
          - "al2-5.0"
        size:
          - "small"
          - "large"
```

`project:codebuild-<project-name>` (obligatorio)

- Ejemplo: `project: "codebuild-myProject"`
- Necesario para todas las configuraciones de pasos de canalización de Buildkite. *<project name>*debe ser igual al nombre del proyecto para el que está configurado el webhook del ejecutor autohospedado.

`queue: "<queue-name>"`

- Ejemplo: `queue: "<queue-name>"`
- Se usa para enrutar los trabajos de Buildkite a una cola específica. Consulte Buildkite Agent Queue Tag para obtener más información.

`image: "<environment-type>-<image-identifier>"`

- Ejemplo: `image: "arm-3.0"`
- Anula la imagen y el tipo de entorno utilizados al iniciar la compilación del ejecutor autoalojado con una imagen seleccionada. Para obtener más información sobre los valores admitidos, consulte Calcule imágenes compatibles con el ejecutor CodeBuild-hosted Buildkite.
  1. Para anular la imagen y el tipo de entorno utilizados con una imagen personalizada, utilice `image: "custom-<environment-type>-<custom-image-identifier>"`
  2. Ejemplo:

```
image:
      "custom-arm-public.ecr.aws/codebuild/amazonlinux-aarch64-standard:3.0"
```

> **Note**
>
> Si la imagen personalizada reside en un registro privado, debes configurar las credenciales de registro adecuadas en tu CodeBuild proyecto.

`instance-size: "<instance-size>"`

- Ejemplo: `instance-size: "medium"`
- Anula el tipo de instancia utilizado al iniciar la compilación del ejecutor autoalojado. Para obtener más información sobre los valores admitidos, consulte Calcule imágenes compatibles con el ejecutor CodeBuild-hosted Buildkite.

`fleet: "<fleet-name>"`

- Ejemplo: `fleet: "myFleet"`
- Anula la configuración de flota establecida en el proyecto para usar la flota especificada. Para obtener más información, consulte Ejecución de compilaciones en flotas de capacidad reservada.

`buildspec-override: "<boolean>"`

- Ejemplo: `buildspec-override: "true"`

- Permite que la compilación ejecute comandos buildspec en las fases `INSTALL`, `PRE_BUILD` y `POST_BUILD` si está establecido en `true`.

Calcule imágenes compatibles con el ejecutor CodeBuild-hosted Buildkite

En la etiqueta que configuróSelf-managed BuildKite Runner en AWS CodeBuild, 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. Para obtener más información acerca de

Tipo de entorno	Identificador de imagen	Tamaño de instancia	Plataforma	Imagen resuelta	Definición

`linux`	`4.0`	`small`
		`medium`
		`large`
		`xlarge`
			Amazon Linux 2	`aws/codebuild/amazonlinux-x86_64-standard:4.0`	al/standard/4.0

`linux`	`5.0`	`2xlarge`
		`gpu_small`
		`gpu_large`
			Amazon Linux 2023	`aws/codebuild/amazonlinux-x86_64-standard:5.0`	al/standard/5.0

`linux-ec2`	`latest`	`small`
		`medium`
		`large`
			Amazon Linux 2023	`aws/codebuild/ami/amazonlinux-x86_64-base:latest`	Ninguno

`arm`	`2.0`	`small` `medium`	Amazon Linux 2	`aws/codebuild/amazonlinux-a`	al/aarch64/0standard/2.

Tipo de entorno	Identificador de imagen	Tamaño de instancia	Plataforma	Imagen resuelta	Definición
		large xlarge 2xlarge		arch64-standard:2.0	
arm	3.0		Amazon Linux 2023	aws/codebuild/amazonlinux-aarch64-standard:3.0	al/aarch64/0standard/3.
arm-ec2	latest	small medium large	Amazon Linux 2023	aws/codebuild/ami/amazonlinux-arm-base:latest	Ninguno
ubuntu	5.0	small medium large xlarge 2xlarge gpu_small gpu_large	Ubuntu 20.04	aws/codebuild/standard:5.0	ubuntu/standard/5.0
ubuntu	6.0		Ubuntu 22.04	aws/codebuild/standard:6.0	ubuntu/standard/6.0
ubuntu	7.0		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

Tipo de entorno	Identificador de imagen	Tamaño de instancia	Plataforma	Imagen resuelta	Definición
			Windows Server Core 2022	`aws/codebuild/windows-base:2022-1.0`	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
`windows-ec2`	`2022`	`medium` `large`	Windows Server Core 2022	`aws/codebuild/ami/windows-base:2022`	Ninguno

Además, puede anular la configuración del entorno de Lambda mediante los siguientes valores. Para obtener más información sobre la computación CodeBuild Lambda, consulte. Ejecutar se basa en AWS Lambda cálculo CodeBuild admite las siguientes imágenes de procesamiento Lambda:

Tipo de entorno	Identificador de imagen	Tamaño de instancia

`linux-lambda`	`dotnet6`
	`go1.21`
	`corretto11`
		`1GB`
		`2GB`
		`4GB`
		`8GB`

`arm-lambda`		

Tipo de entorno	Identificador de imagen	Tamaño de instancia			

	`corretto17`
	
	`corretto21`
	
	`nodejs18`
	
	`nodejs20`
	
	`python3.11`
	
	`python3.12`
	
	`ruby3.2`
		`10GB`			

Para obtener más información, consulte Modos y tipos de computación del entorno de compilación y Imágenes de Docker proporcionadas por CodeBuild.

Usa webhooks con AWS CodeBuild

AWS CodeBuild admite la integración de webhooks con GitHub Enterprise Server GitHub, Self Managed y Bitbucket. GitLab GitLab

Temas

- Mejores prácticas para usar webhooks con AWS CodeBuild
- Eventos de webhooks de Bitbucket
- GitHub webhooks globales y organizacionales
- GitHub webhooks manuales
- GitHub eventos de webhook
- GitLab webhooks grupales
- GitLab webhooks manuales

- GitLab eventos de webhook
- Webhooks manuales de Buildkite
- Aprobación de comentarios de solicitudes de extracción

Mejores prácticas para usar webhooks con AWS CodeBuild

Para los proyectos que utilizan repositorios públicos para configurar webhooks, recomendamos las siguientes opciones:

Configura los filtros `ACTOR_ACCOUNT_ID`

Añade `ACTOR_ACCOUNT_ID` filtros a los grupos de filtros de webhooks de tu proyecto para especificar qué usuarios pueden activar una compilación. Cada evento de webhook al que CodeBuild se envía viene con información del remitente que especifica el identificador del actor. CodeBuild filtrará los webhooks según el patrón de expresiones regulares 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.

Configura los filtros `FILE_PATH`

Añade `FILE_PATH` filtros a los grupos de filtros de webhooks de tu proyecto para incluir o excluir los archivos que pueden activar una compilación si 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, consulte ProjectSource.buildspec en la Referencia de la API de 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 debe habilitar el evento `declined` en su configuración de webhook de Bitbucket si crea un grupo de filtros de webhook que utilice el tipo de evento `PULL_REQUEST_CLOSED`.

Valor del encabezado **`X-Event-Key`**	Tipo de evento
`repo:push`	`PUSH`
`pullrequest:created`	`PULL_REQUEST_CREATED`
`pullrequest:updated`	`PULL_REQUEST_UPDATED`
`pullrequest:fulfilled`	`PULL_REQUEST_MERGED`
`pullrequest:rejected`	`PULL_REQUEST_CLOSED`

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.

CodeBuild evalúa los patrones de filtro mediante la sintaxis de expresiones RE2 regulares, que no admite mirar hacia adelante, hacia atrás, referencias inversas ni grupos atómicos. Para obtener más información, consulte la página de RE2 sintaxis del sitio web. GitHub

`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.

> **Note**
>
> El `FILE_PATH` filtro se evalúa comparándolo con los primeros 3000 archivos modificados en una solicitud push o pull. Para asegurarte de que el filtro se aplica según lo previsto, mantén el número de archivos modificados dentro de este límite. En el caso de los eventos de solicitud de extracción, también recomendamos usar una política de creación de solicitudes de extracción para controlar qué solicitudes de extracción pueden iniciar una compilación.

`COMMIT_MESSAGE`

Un webhook desencadena una compilación cuando el mensaje de confirmación del encabezado coincide con la expresión regular.

`WORKFLOW_NAME`

Un webhook desencadena una compilación cuando el nombre del flujo de trabajo coincide con el patrón de expresión regular.

> **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)
- Filtra los eventos de webhook de Bitbucket (CloudFormation)

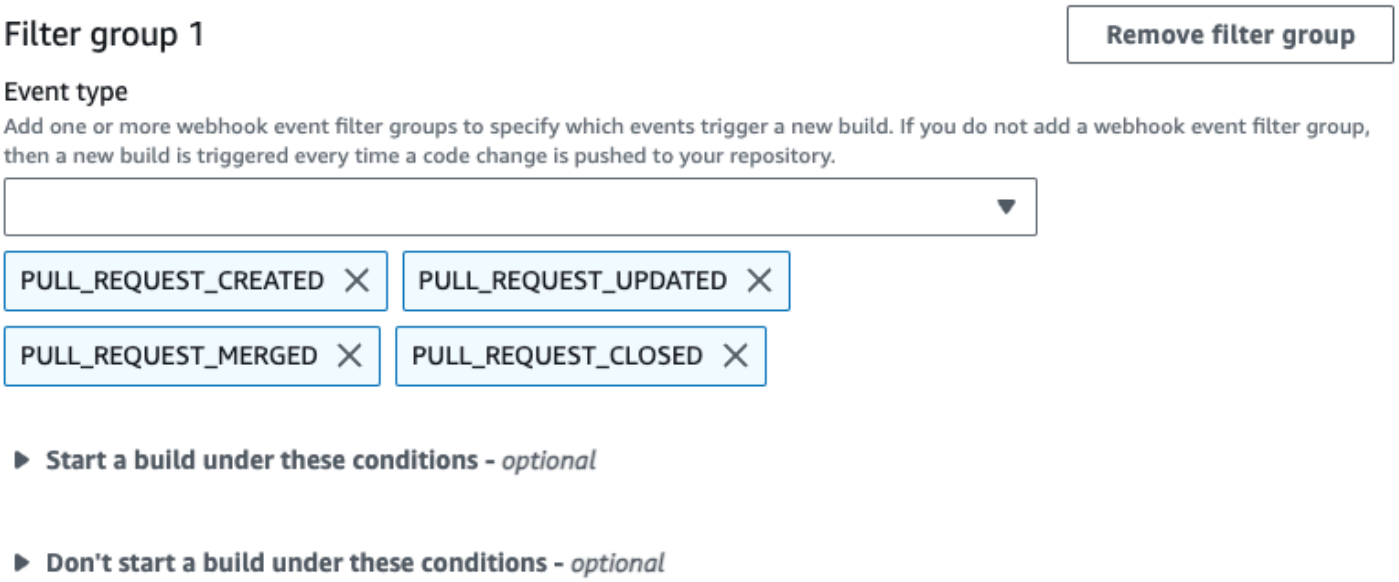
Filtrar eventos de webhooks de Bitbucket (consola)

Para usarla para filtrar Consola de administración de AWS los eventos de webhooks:

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.

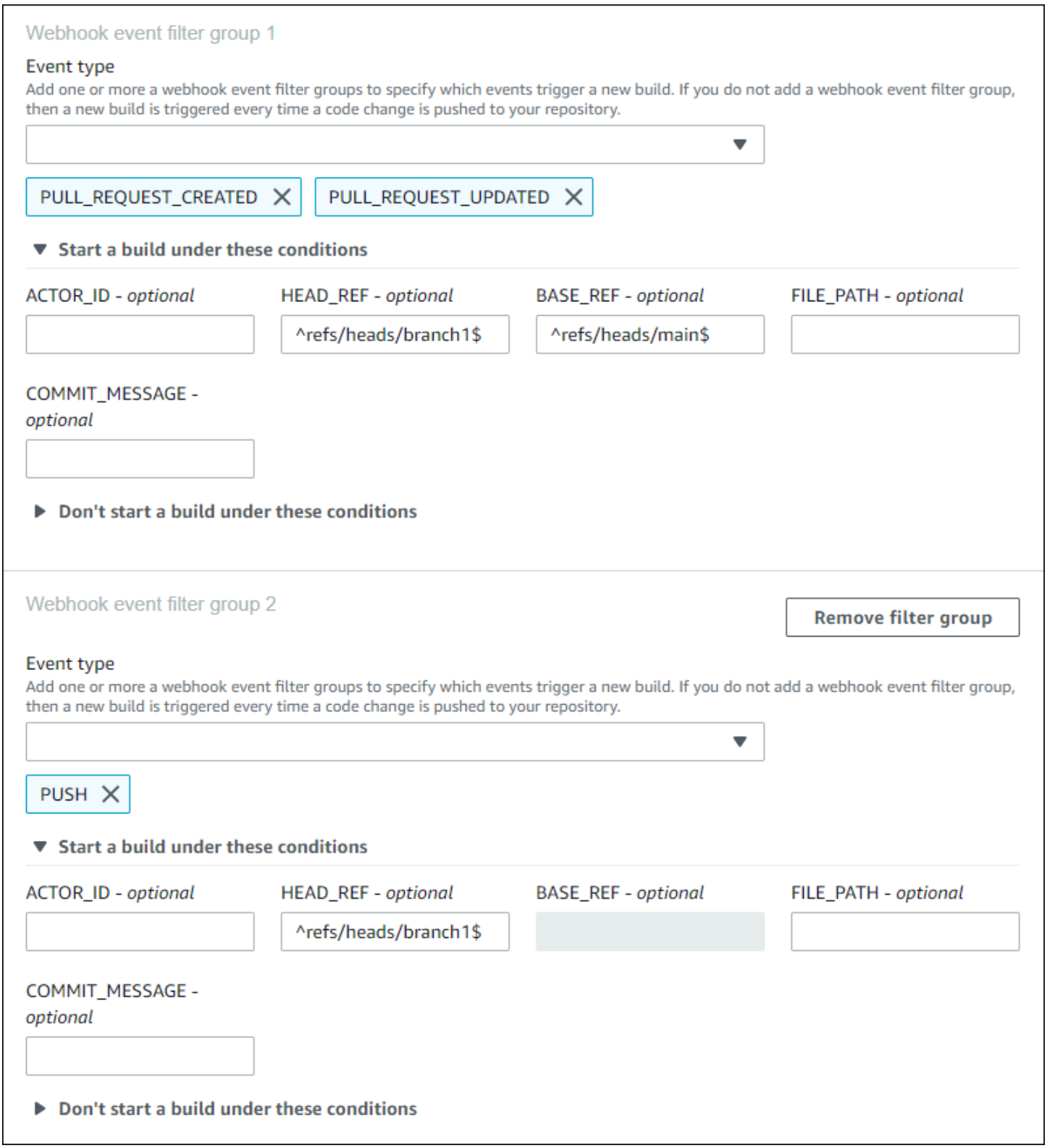
Para obtener más información consulte Creación de un proyecto de compilación (consola) y WebhookFilter en la Referencia de la API de AWS CodeBuild .

En este ejemplo, un grupo de filtros de webhooks desencadena una compilación únicamente con solicitudes de extracción:

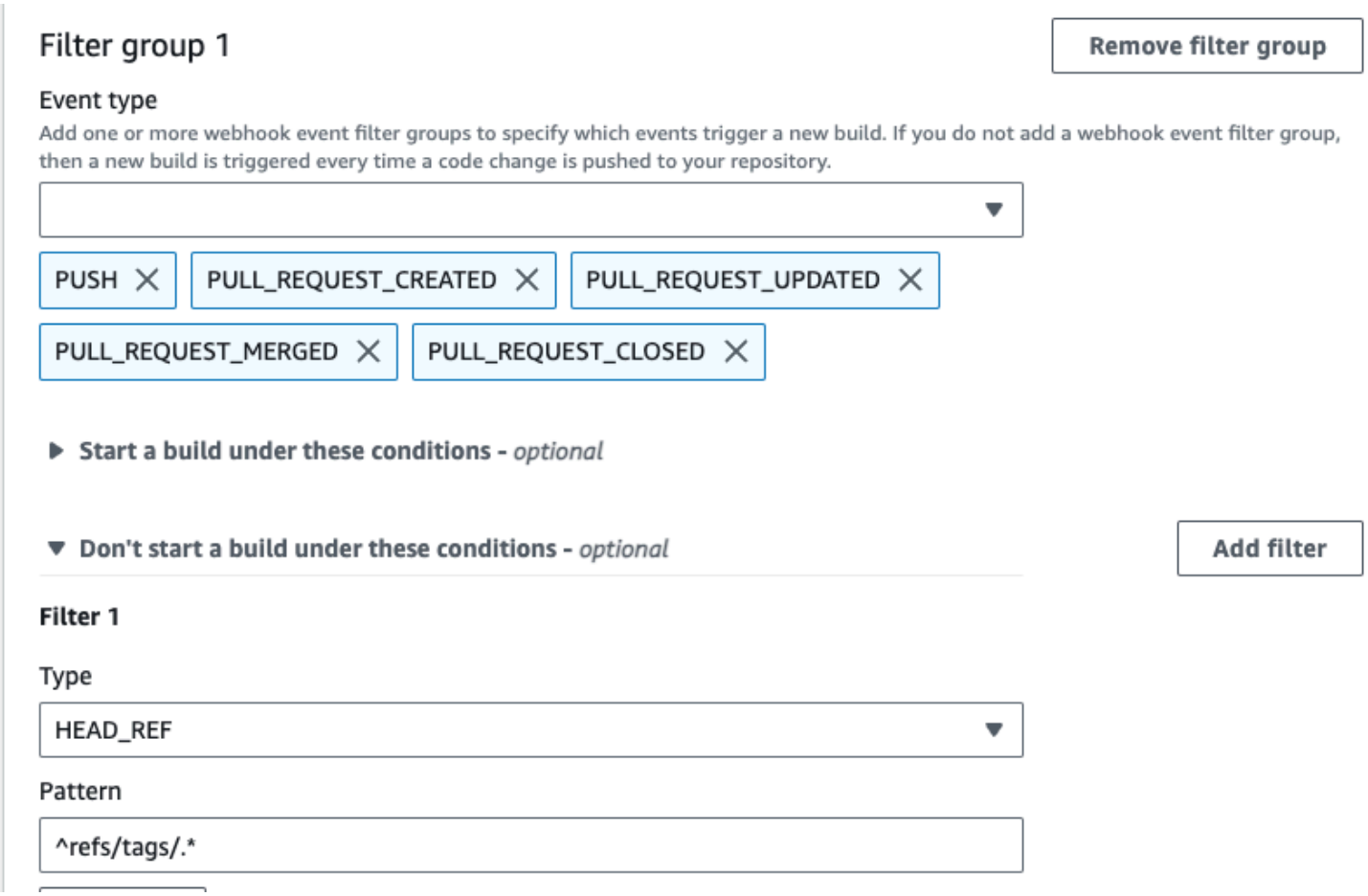


Si utilizamos un ejemplo con dos grupos de filtros, la compilación se desencadenaría cuando uno de los grupos o los dos se evalúen como true:

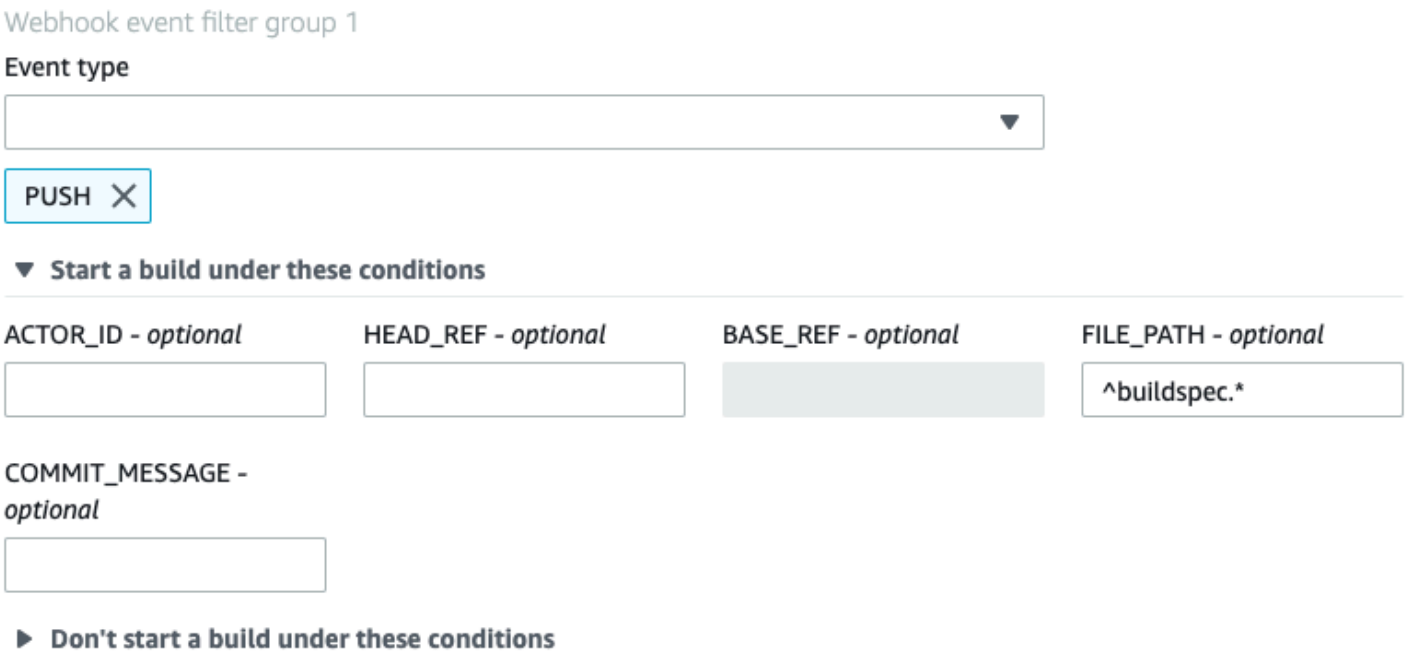
- El primer grupo de filtros especifica las solicitudes de extracción que se crean o actualizan en ramificaciones con nombres de referencia de Git que coinciden con la expresión regular `^refs/heads/main$` y las referencias de encabezado que coinciden con `^refs/heads/branch1!`.
- El segundo grupo de filtros especifica solicitudes de inserción en ramificaciones con nombres de referencia de Git que coinciden con la expresión regular `^refs/heads/branch1$`.



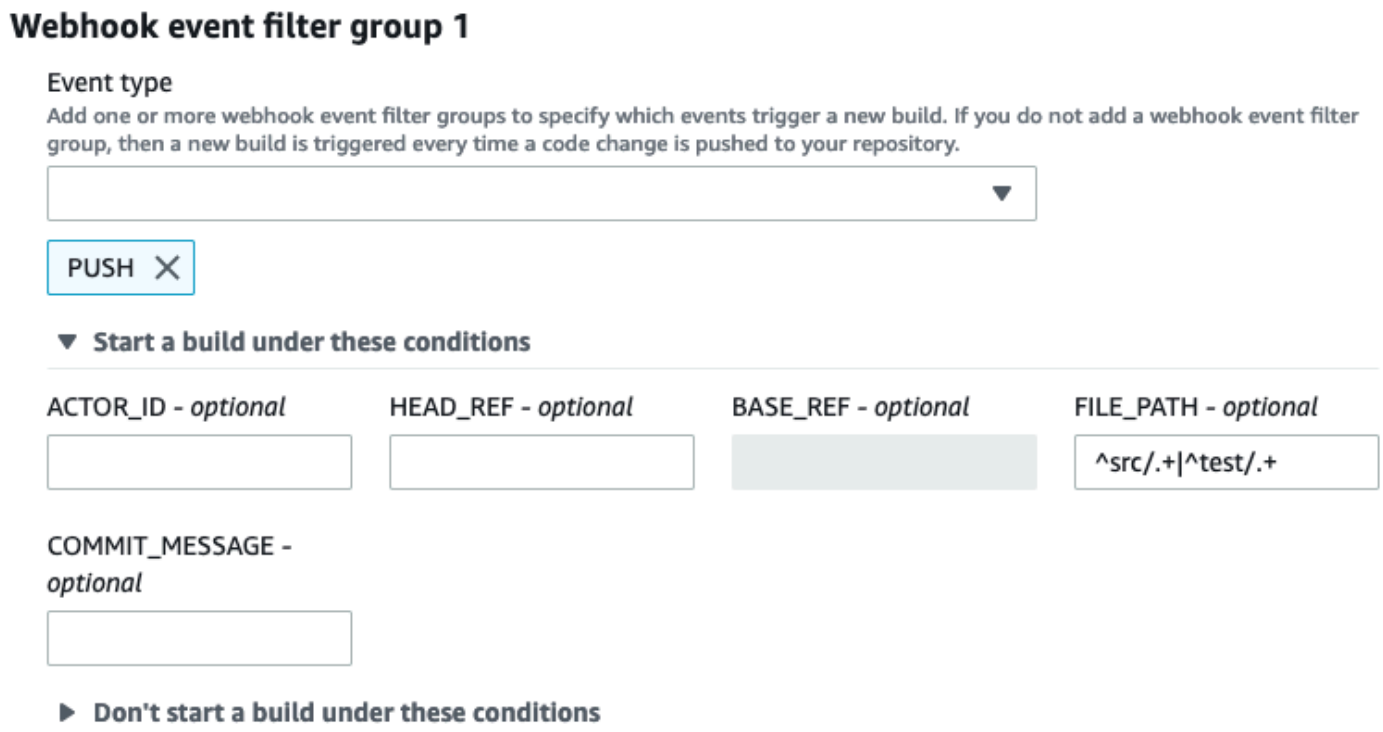
En este ejemplo, un grupo de filtros de webhooks desencadena una compilación de todas las solicitudes excepto los eventos de etiquetas.



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.*`.

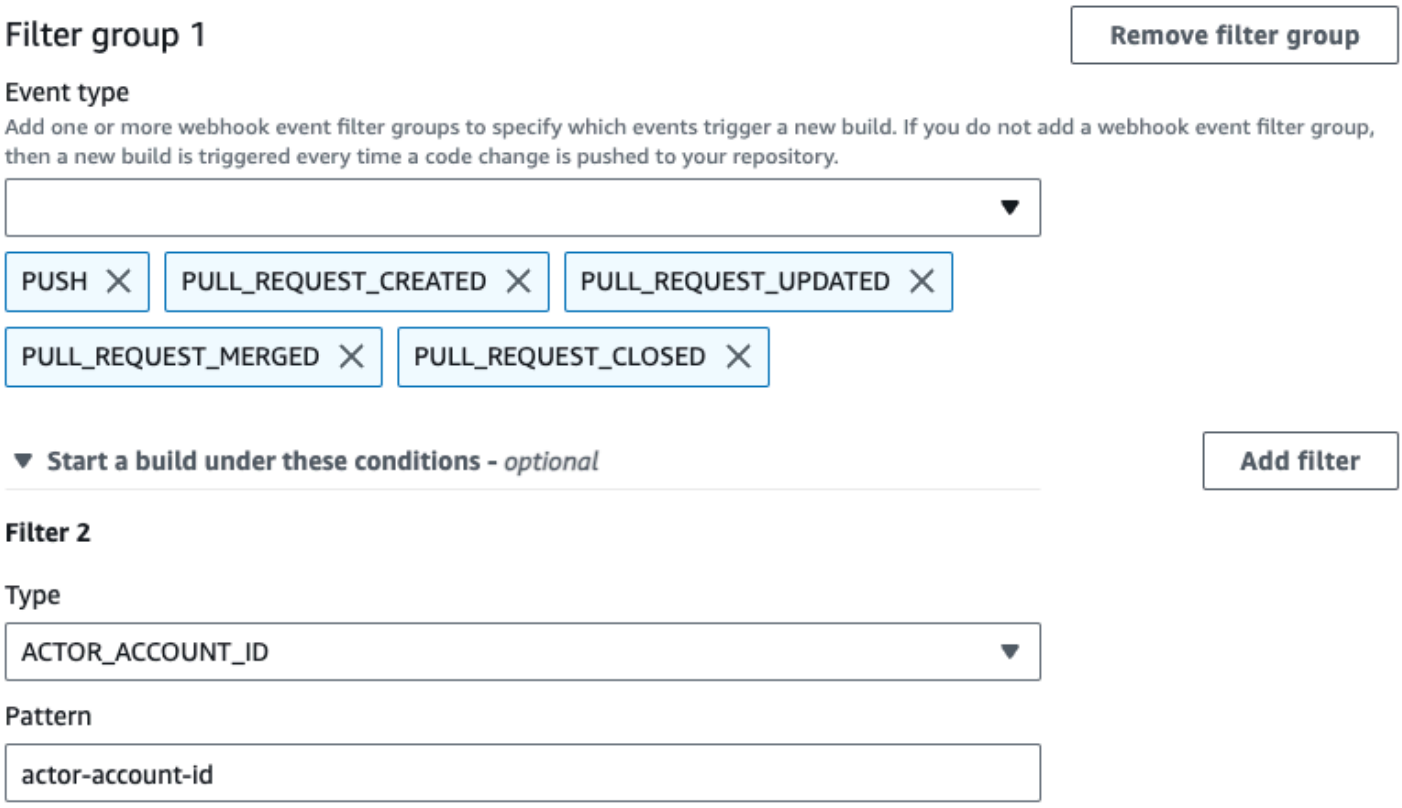


En este ejemplo, un grupo de filtros de webhook activa una compilación solo cuando se cambian archivos en las carpetas `src` o `test`.

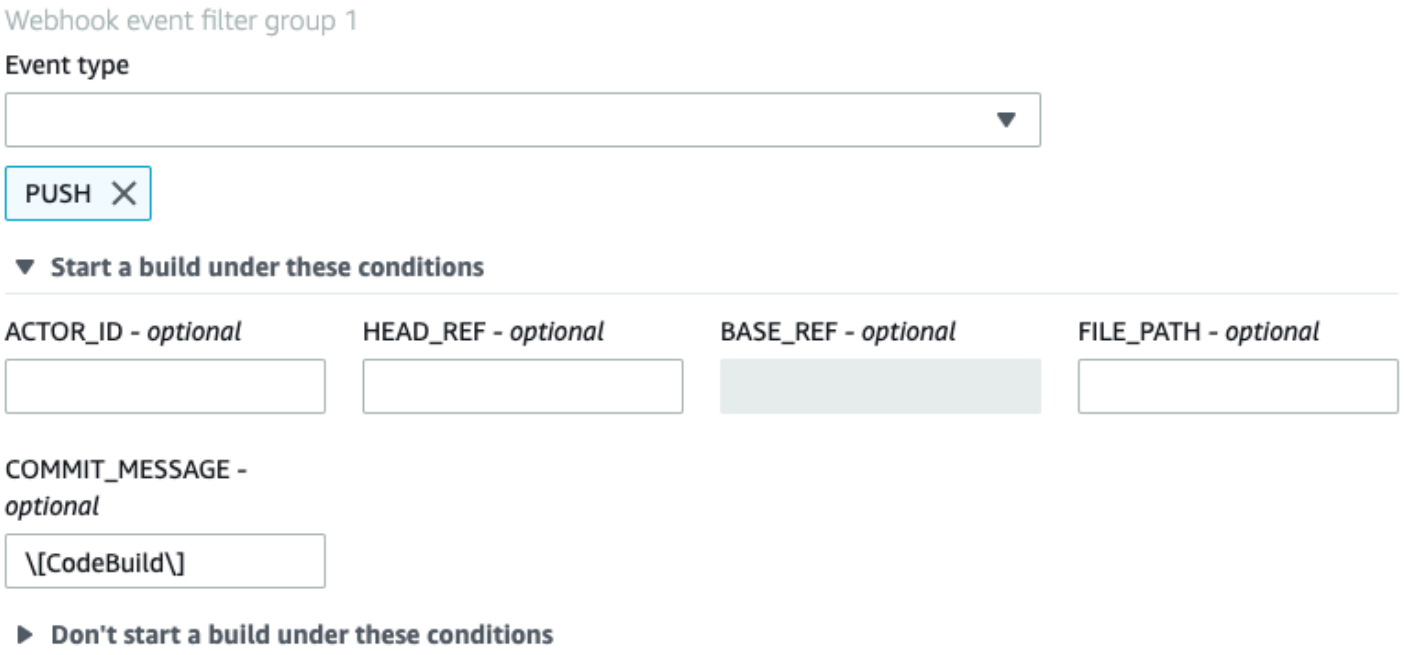


En este ejemplo, un grupo de filtros de webhooks desencadena una compilación únicamente cuando hay un cambio realizado por un usuario de Bitbucket que no tiene un ID de cuenta que coincida con la expresión regular `actor-account-id`.

> **Note**
>
> Para obtener información sobre cómo encontrar el identificador de tu cuenta de Bitbucket https://api.bitbucket.org/2.0/users/*user-name*, consulta ¿Dónde *user-name* está tu nombre de usuario de Bitbucket?



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 `\[CodeBuild\]`.



Filtrar eventos de webhooks de Bitbucket (SDK)

Para usar el AWS CodeBuild SDK para filtrar eventos de webhooks, usa el `filterGroups` campo de la sintaxis de solicitud de los métodos o de la `CreateWebhook` API. `UpdateWebhook` Para obtener más información, consulta WebhookFilter en la CodeBuild Referencia de la API de .

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 sobre cómo encontrar el ID de tu cuenta de Bitbucket https://api.bitbucket.org/2.0/users/*user-name*, consulta ¿Dónde *user-name* está tu 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": "\[CodeBuild\]"
```

```
      }
    ]
  ]
```

Filtra los eventos de webhook de Bitbucket (CloudFormation)

Para usar una CloudFormation plantilla para filtrar los eventos de webhook, usa la propiedad del AWS CodeBuild proyecto. `FilterGroups` La siguiente YAML-formatted parte de una CloudFormation plantilla 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: READ_ME
          ExcludeMatchedPattern: true
      - - Type: EVENT
          Pattern: PUSH
        - Type: COMMIT_MESSAGE
          Pattern: \[CodeBuild\]
        - Type: FILE_PATH
          Pattern: ^src/.+|^test/.+
```

GitHub webhooks globales y organizacionales

Puedes usar webhooks CodeBuild GitHub globales u organizacionales para crear compilaciones a partir de eventos de webhooks desde cualquier repositorio de una GitHub organización o empresa. Los webhooks globales y organizacionales funcionan con cualquiera de los tipos de eventos de GitHub webhook existentes y se pueden configurar añadiendo una configuración de alcance al crear un webhook. CodeBuild También puedes usar webhooks globales y organizativos para configurar ejecutores de GitHub acción autohospedados para recibir `WORKFLOW_JOB_QUEUED` eventos CodeBuild de varios repositorios dentro de un mismo proyecto.

Temas

- Configura un webhook global o organizacional GitHub
- Filtra los eventos de webhooks GitHub globales u organizacionales (consola)
- Filtrar los eventos GitHub de webhooks de la organización (CloudFormation)

Configura un webhook global o organizacional GitHub

Los pasos básicos para configurar un GitHub webhook global u organizacional son los siguientes. Para obtener más información sobre los GitHub webhooks globales y organizacionales, consulte. GitHub webhooks globales y organizacionales

1. Establezca la ubicación de origen del proyecto en `CODEBUILD_DEFAULT_WEBHOOK_SOURCE_LOCATION`.
2. En la configuración de ámbito del webhook, establezca el ámbito en `GITHUB_ORGANIZATION` o `GITHUB_GLOBAL`, en función de si debe ser un webhook global o de organización. Para obtener más información, consulte Tipos de webhooks.
3. Especifique un nombre como parte de la configuración del ámbito del webhook. En el caso de los webhooks de organización, este es el nombre de la organización y, para los webhooks globales, es el nombre de la empresa.

   > **Note**
   >
   > Si el tipo de origen del proyecto es `GITHUB_ENTERPRISE`, también debe especificar un dominio como parte de la configuración del ámbito del webhook.

4. (Opcional) Si solo quiere recibir eventos de webhook para repositorios específicos de su organización o empresa, puede especificar `REPOSITORY_NAME` como filtro al crear el webhook.
5. Si vas a crear un webhook para una organización, asegúrate de que CodeBuild cuenta con los permisos necesarios para crear webhooks a nivel de organización. GitHub Puedes crear un token de acceso GitHub personal con los permisos de webhook de la organización o usar OAuth. CodeBuild Para obtener más información, consulte GitHub y un token de acceso a GitHub Enterprise Server.

   Ten en cuenta que los webhooks de la organización funcionan con cualquiera de los tipos de eventos de webhook existentes GitHub .
6. Si va a crear un webhook global, deberá crearlo de forma manual. Para obtener más información sobre cómo crear manualmente un webhook desde dentro GitHub, consulta. GitHub webhooks manuales

   Tenga en cuenta que los webhooks globales solo admiten el tipo de evento `WORKFLOW_JOB_QUEUED`. Para obtener más información, consulte Tutorial: Configurar un ejecutor de CodeBuild-hosted GitHub acciones.

Filtra los eventos de webhooks GitHub globales u organizacionales (consola)

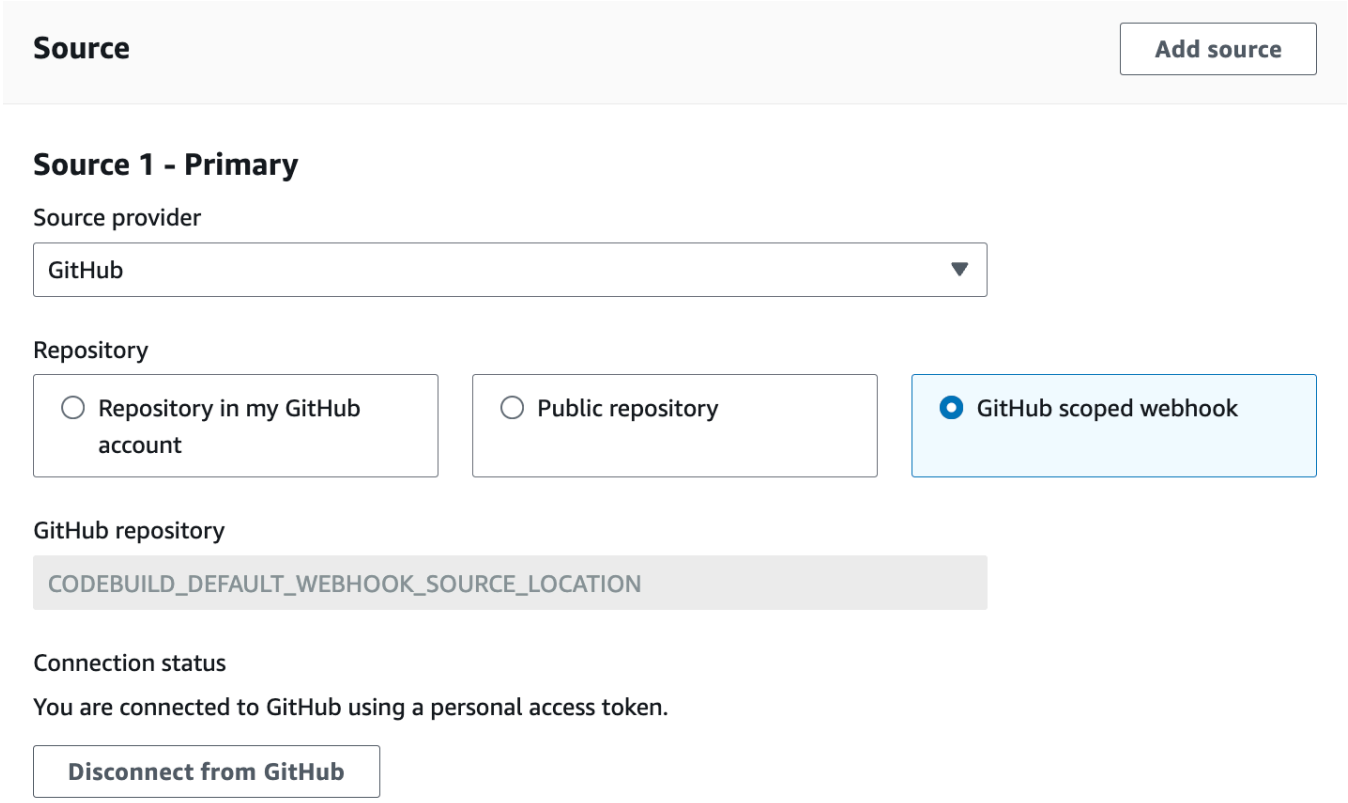
Al crear un GitHub proyecto a través de la consola, selecciona las siguientes opciones para crear un webhook GitHub global u organizacional dentro del proyecto. Para obtener más información sobre los GitHub webhooks globales y organizativos, consulte. GitHub webhooks globales y organizacionales

1. Abra 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):
     - Para el proveedor de origen, elija GitHub o GitHub Enterprise.
     - En Repository, selecciona GitHub Sped Webhook.

     El GitHub repositorio se configurará automáticamente en`CODEBUILD_DEFAULT_WEBHOOK_SOURCE_LOCATION`, que es la ubicación de origen requerida para los webhooks globales y organizacionales.

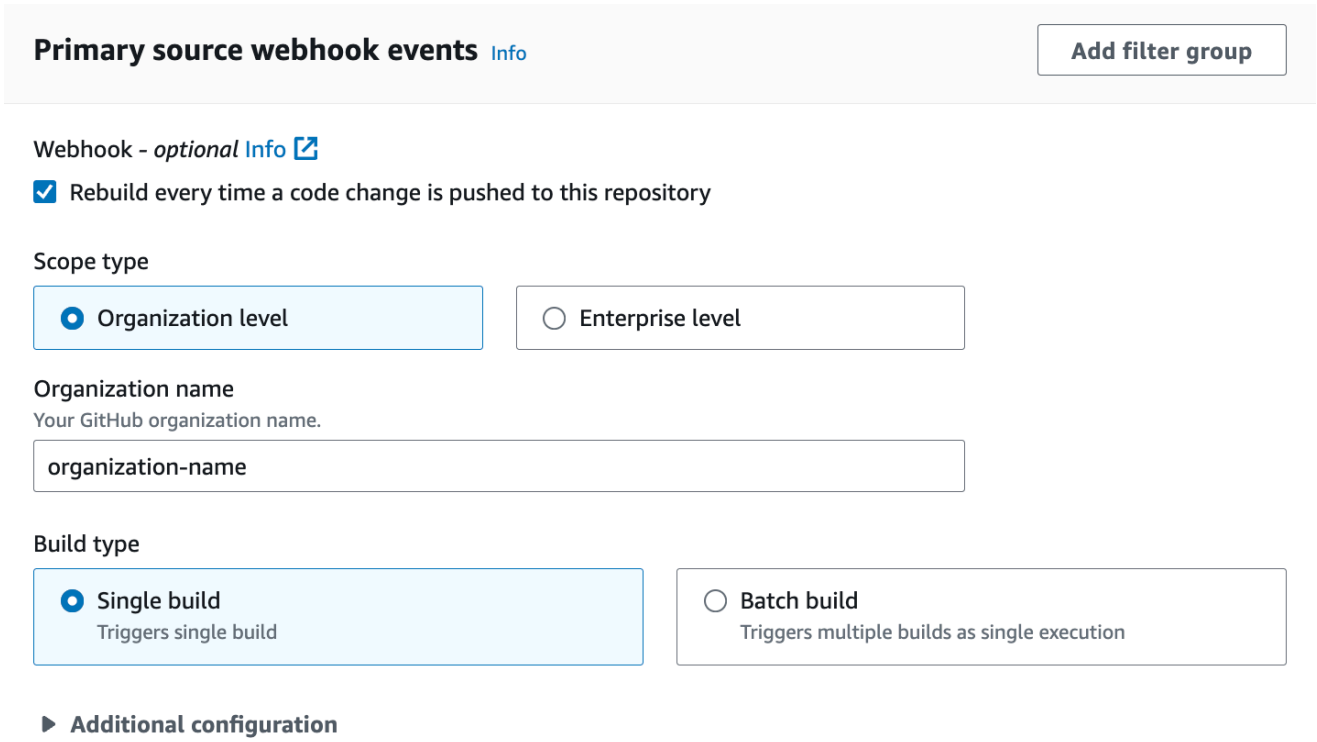
     > **Note**
     >
     > Si utilizas webhooks de organizaciones, asegúrate de que CodeBuild tienen permisos para crear webhooks a nivel de organización. GitHub Si utilizas una conexión de OAuth existente, es posible que tengas que volver a generar la conexión para conceder este permiso. CodeBuild Como alternativa, puedes crear el webhook manualmente mediante la función de webhooks manuales. CodeBuild Ten en cuenta que si ya tienes un token de GitHub OAuth y quieres añadir permisos de organización adicionales, puedes revocar el permiso del token de OAuth y volver a conectarlo a través de la consola. CodeBuild



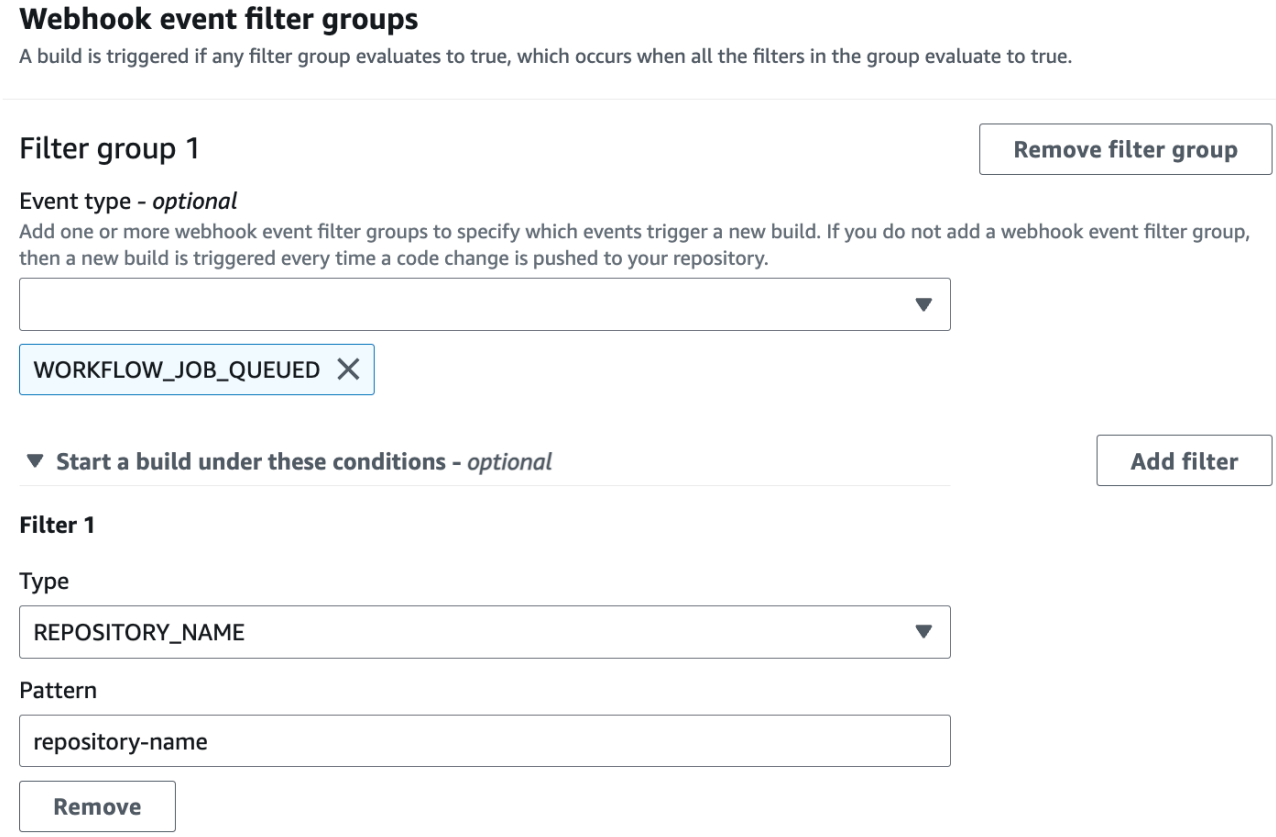
- En Eventos de webhook de origen principal:
  - En Tipo de ámbito, elija Nivel de organización si va a crear un webhook de organización o Nivel empresarial si va a crear un webhook global.
  - En Nombre, especifique el nombre de la empresa o de la organización en función de si el webhook es global o de organización.

    Si el tipo de origen del proyecto es `GITHUB_ENTERPRISE`, también debe especificar un dominio como parte de la configuración de la organización del webhook. Por ejemplo, si la URL de su organización es **`https://domain.com/orgs/org-name`**, el dominio es **`https://domain.com`**.

    > **Note**
    >
    > Este nombre no puede modificarse una vez que se haya creado el webhook. Para cambiar el nombre, puede eliminar el webhook y volver a crearlo. Si quieres eliminar el webhook por completo, también puedes actualizar la ubicación de origen del proyecto a un repositorio. GitHub



- (Opcional) En Grupos de filtros de eventos de webhook, puede especificar qué eventos quiere que desencadenen una nueva compilación. También puede especificar `REPOSITORY_NAME` como filtro para que solo se desencadenen compilaciones a partir de eventos de webhook de repositorios específicos.



También puedes configurar el tipo de evento `WORKFLOW_JOB_QUEUED` para configurar ejecutores de GitHub acciones autohospedados. Para obtener más información, consulte Tutorial: Configurar un ejecutor de CodeBuild-hosted GitHub acciones.

3. Continúe con los valores predeterminados y, a continuación, elija Crear el proyecto de compilación.

Filtrar los eventos GitHub de webhooks de la organización (CloudFormation)

Para usar una CloudFormation plantilla para filtrar los eventos de webhooks de la organización, usa la propiedad del AWS CodeBuild `ScopeConfiguration` proyecto. Para obtener más información sobre los GitHub webhooks globales y organizacionales, consulte. GitHub webhooks globales y organizacionales

> **Note**
>
> Los webhooks globales y GitHub empresariales no son compatibles con. CloudFormation

La siguiente YAML-formatted parte de una CloudFormation plantilla crea cuatro grupos de filtros. Juntos, desencadenan una compilación cuando uno o todos se evalúan como true:

- El primer grupo de filtros especifica que un GitHub usuario que no tiene un identificador de cuenta cree o actualice las solicitudes de extracción en ramas con nombres `^refs/heads/main$` de referencia de Git que coincidan con la expresión regular`12345`.
- El segundo grupo de filtros especifica las solicitudes de inserción que se crean en archivos cuyos nombres coinciden con la expresión regular `READ_ME` 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 del 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
      ScopeConfiguration:
        Name: organization-name
        Scope: GITHUB_ORGANIZATION
      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: READ_ME
            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: \[CI-CodeBuild\]
```

GitHub webhooks manuales

Puedes configurar los GitHub webhooks manuales para CodeBuild evitar que intenten crear automáticamente un webhook en ellos. GitHub CodeBuild devuelve una URL de carga útil como parte de la llamada para crear el webhook, que se puede usar para crear manualmente el webhook en su interior. GitHub Aunque no CodeBuild esté autorizado a crear un webhook en tu GitHub cuenta, puedes crear manualmente un webhook para tu proyecto de compilación.

Usa el siguiente procedimiento para crear un GitHub webhook manual.

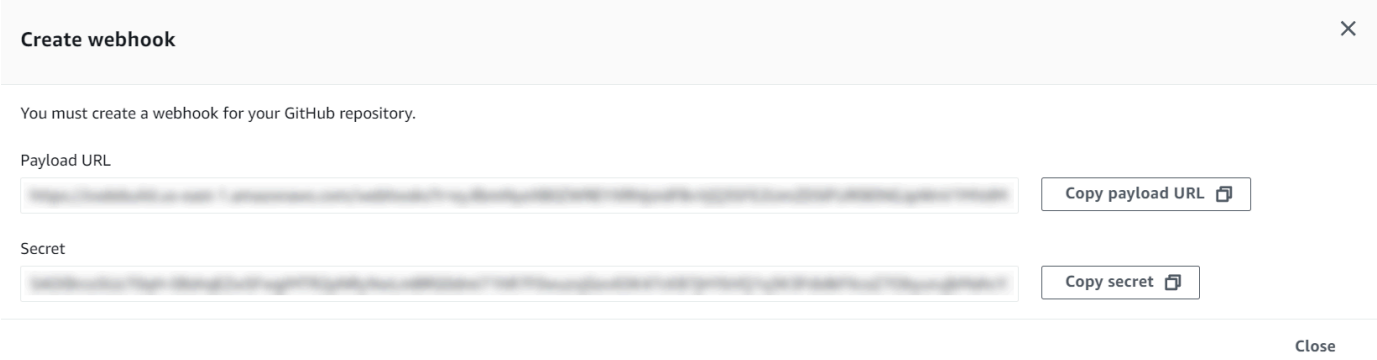
Para crear un webhook GitHub manual

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 el caso del proveedor de origen, elija GitHub.
     - En Repositorio, seleccione Repositorio en mi GitHub cuenta.

- En URL de repositorio, especifique **`https://github.com/user-name/repository-name`**

- En Eventos de webhook de origen principal:
  - En Webhook (opcional), seleccione Volver a compilar cada vez que un cambio de código se envía a este repositorio.
  - Selecciona Configuración adicional y, para Creación manual (opcional), elige Crear manualmente un webhook para este repositorio en la GitHub consola. .

3. Continúe con los valores predeterminados y, a continuación, elija Crear el proyecto de compilación. Anote los valores de URL de carga y Secreto, ya que los utilizará más adelante.



4. Abre la GitHub consola en `https://github.com/user-name/repository-name/settings/hooks` y selecciona Agregar webhook.

   - En Payload URL, especifique el valor de la URL de carga útil que ha anotado anteriormente.
   - En Tipo de contenido, elige application/json.
   - En Secret, especifique el valor de secreto que ha anotado anteriormente.
   - Configura los eventos individuales a los que se enviará una carga útil de webhook. CodeBuild En Which events would you like to trigger this webhook?, elija Let me select individual events y, a continuación, seleccione entre los eventos siguientes: Pushes, Pull requests y Releases. Si quiere iniciar compilaciones para eventos `WORKFLOW_JOB_QUEUED`, seleccione Workflow jobs. Para obtener más información sobre GitHub Actions Runners, consulta. Tutorial: Configurar un ejecutor de CodeBuild-hosted GitHub acciones Para obtener más información sobre los tipos de eventos que admite CodeBuild, consulteGitHub eventos de webhook.

5. Elija Add webhook.

GitHub eventos de webhook

Puedes usar los grupos de filtros de webhooks 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, puedes elegir uno o más de los siguientes eventos:PUSH,,PULL_REQUEST_CREATED,PULL_REQUEST_UPDATED,PULL_REQUEST_REOPENED,PULL PULL_REQUEST_CLOSED RELEASEDPRERELEASED, 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`

Valor del encabezado **`X-GitHub-Event`**	Valor **`action`** de la carga del evento de webhook	Tipo de evento
`push`	n/a	`PUSH`
`release`	released	`RELEASED`
`release`	prereleased	`PRERELEASED`
`workflow_job`	queued	`WORKFLOW_JOB_QUEUED`

> **Note**
>
> El tipo de `PULL_REQUEST_REOPENED` evento solo se puede usar con GitHub un servidor GitHub empresarial. El tipo de `PRERELEASED` evento `RELEASED` y GitHub solo se puede usar con. Para obtener más información sobre `WORKFLOW_JOB_QUEUED`, consulte Tutorial: Configurar un ejecutor de CodeBuild-hosted GitHub acciones.

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.

CodeBuild evalúa los patrones de filtro mediante la sintaxis de expresiones RE2 regulares, que no admite grupos atómicos, retroreferencias ni referencias anticipadas. Para obtener más información, consulte la página de RE2 sintaxis del sitio web. GitHub

`ACTOR_ACCOUNT_ID` (`ACTOR_ID` en la consola)

Un evento de webhook desencadena una compilación cuando el identificador de una cuenta GitHub o de GitHub Enterprise Server coincide con el patrón de expresión regular. 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 solicitud GitHub push y pull y con los eventos push de GitHub Enterprise Server. No se puede usar con los eventos de solicitud de extracción de GitHub Enterprise Server.

> **Note**
>
> El `FILE_PATH` filtro se evalúa comparándolo con los 100 primeros archivos modificados en una solicitud de inserción o extracción. Para asegurarte de que el filtro se aplica según lo previsto, mantén el número de archivos modificados dentro de este límite. En el caso de los eventos de solicitud de extracción, también recomendamos usar una política de creación de solicitudes de extracción para controlar qué solicitudes de extracción pueden iniciar una compilación.

`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 solicitud de GitHub inserción y extracción y con los eventos de inserción de GitHub Enterprise Server. No se puede usar con los eventos de solicitud de extracción de GitHub Enterprise Server.

`TAG_NAME`

Un webhook desencadena una compilación cuando el nombre de etiqueta de la versión coincide con el patrón de expresión regular. Se puede usar un `TAG_NAME` filtro con los eventos de solicitud GitHub publicados y prepublicados.

`RELEASE_NAME`

Un webhook desencadena una compilación cuando el nombre de la versión coincide con el patrón de expresión regular. Se puede usar un `RELEASE_NAME` filtro con eventos de solicitud GitHub publicados y prepublicados.

`REPOSITORY_NAME`

Un webhook desencadena una compilación cuando el nombre del repositorio coincide con el patrón de expresión regular. Un `REPOSITORY_NAME` filtro solo se puede usar con webhooks GitHub globales u organizacionales.

`ORGANIZATION_NAME`

Un webhook desencadena una compilación cuando el nombre de la organización coincide con el patrón de expresión regular. Un `ORGANIZATION_NAME` filtro solo se puede usar con webhooks GitHub globales.

`WORKFLOW_NAME`

Un webhook desencadena 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 solicitudes de trabajos en cola del flujo de trabajo de GitHub Actions.

> **Note**
>
> Puedes encontrar la carga útil de los webhooks en la configuración de los webhooks de tu repositorio. GitHub

Temas

- Filtra los eventos de GitHub webhooks (consola)
- Filtra los eventos de GitHub webhooks (SDK)
- Filtrar eventos de GitHub webhooks (CloudFormation)

Filtra los eventos de GitHub webhooks (consola)

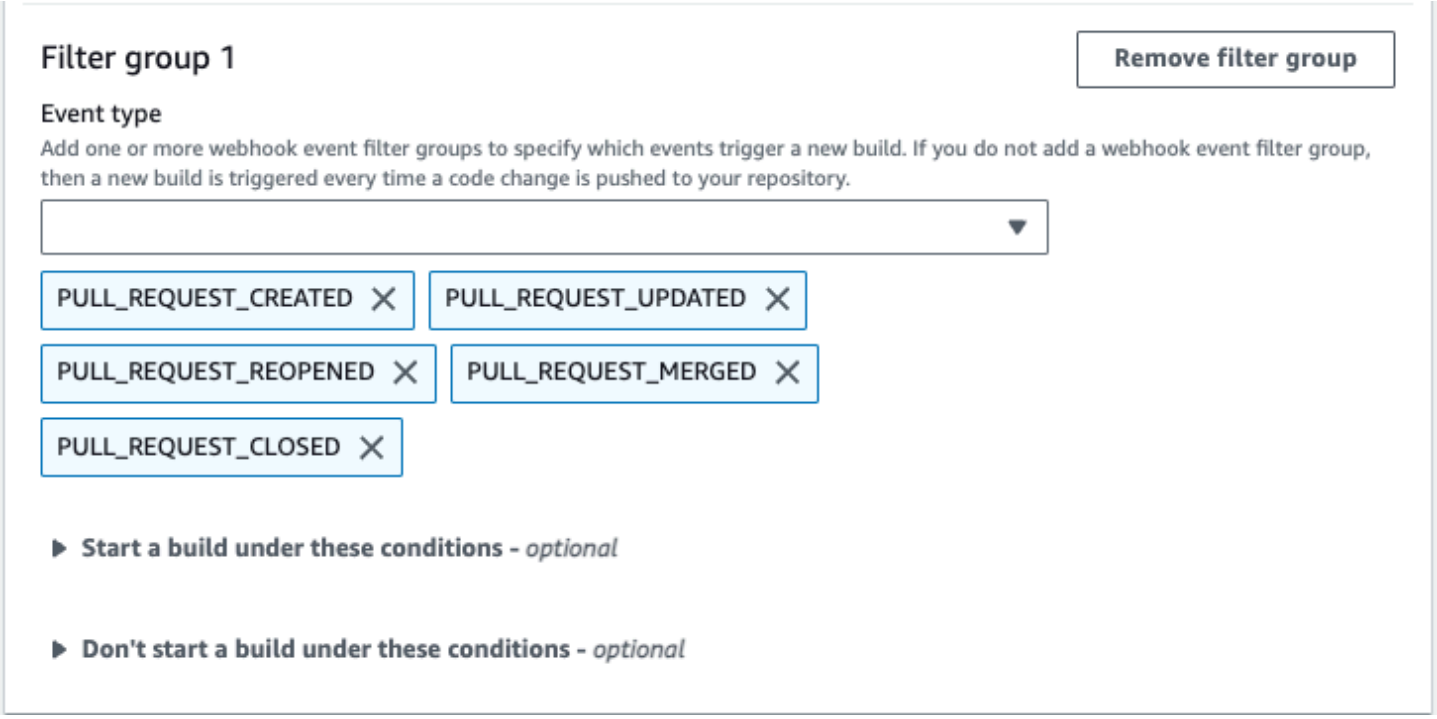
Siga las instrucciones siguientes para filtrar los eventos de GitHub webhook mediante. Consola de administración de AWS Para obtener más información sobre los eventos de GitHub webhook, consulte. GitHub eventos de webhook

En Eventos de webhook de origen principal, seleccione lo siguiente. Esta sección solo está disponible si eliges Repositorio en mi GitHub cuenta para el 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.

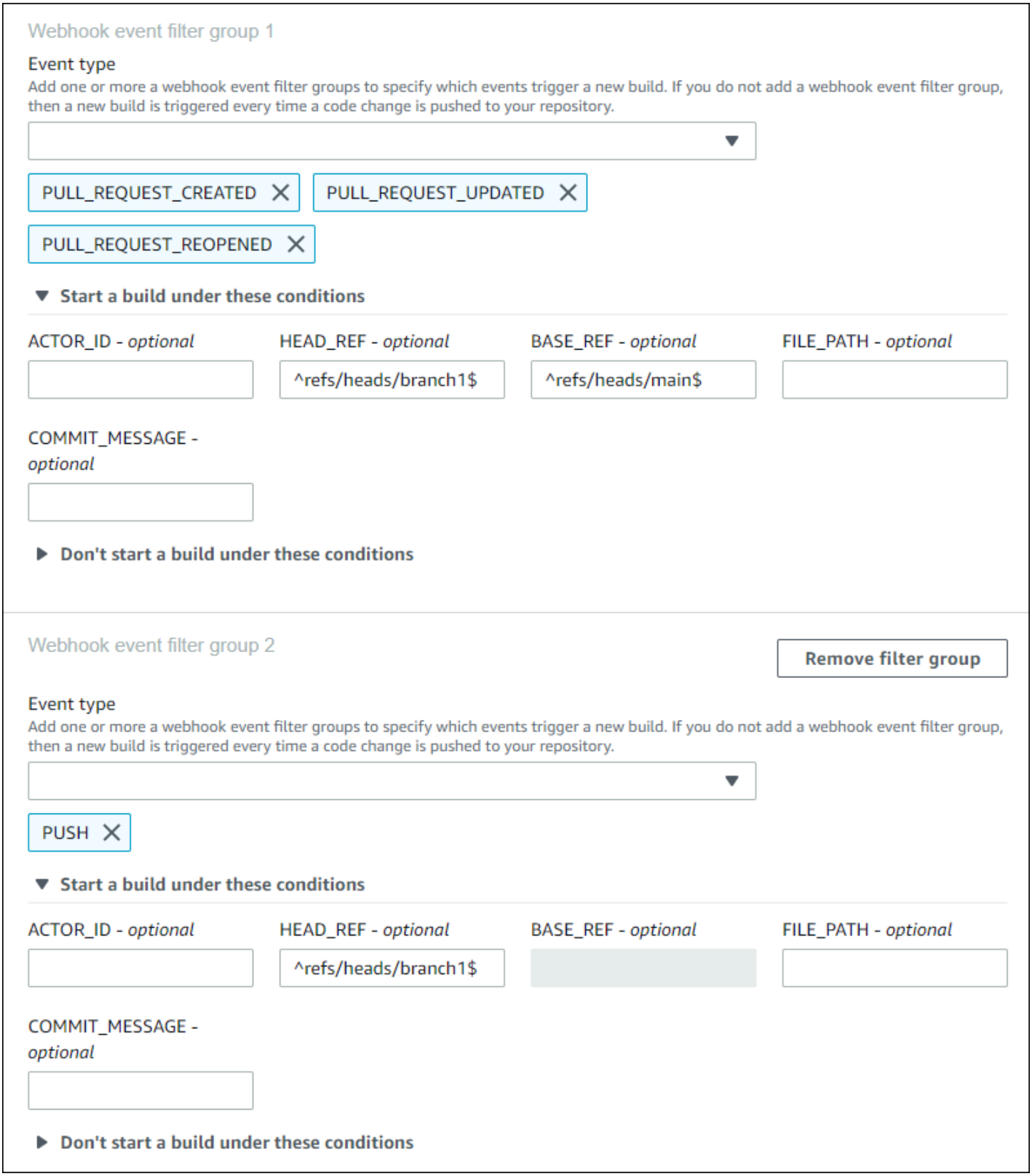
Para obtener más información consulte Creación de un proyecto de compilación (consola) y WebhookFilter en la Referencia de la API de AWS CodeBuild .

En este ejemplo, un grupo de filtros de webhooks desencadena una compilación únicamente con solicitudes de extracción:

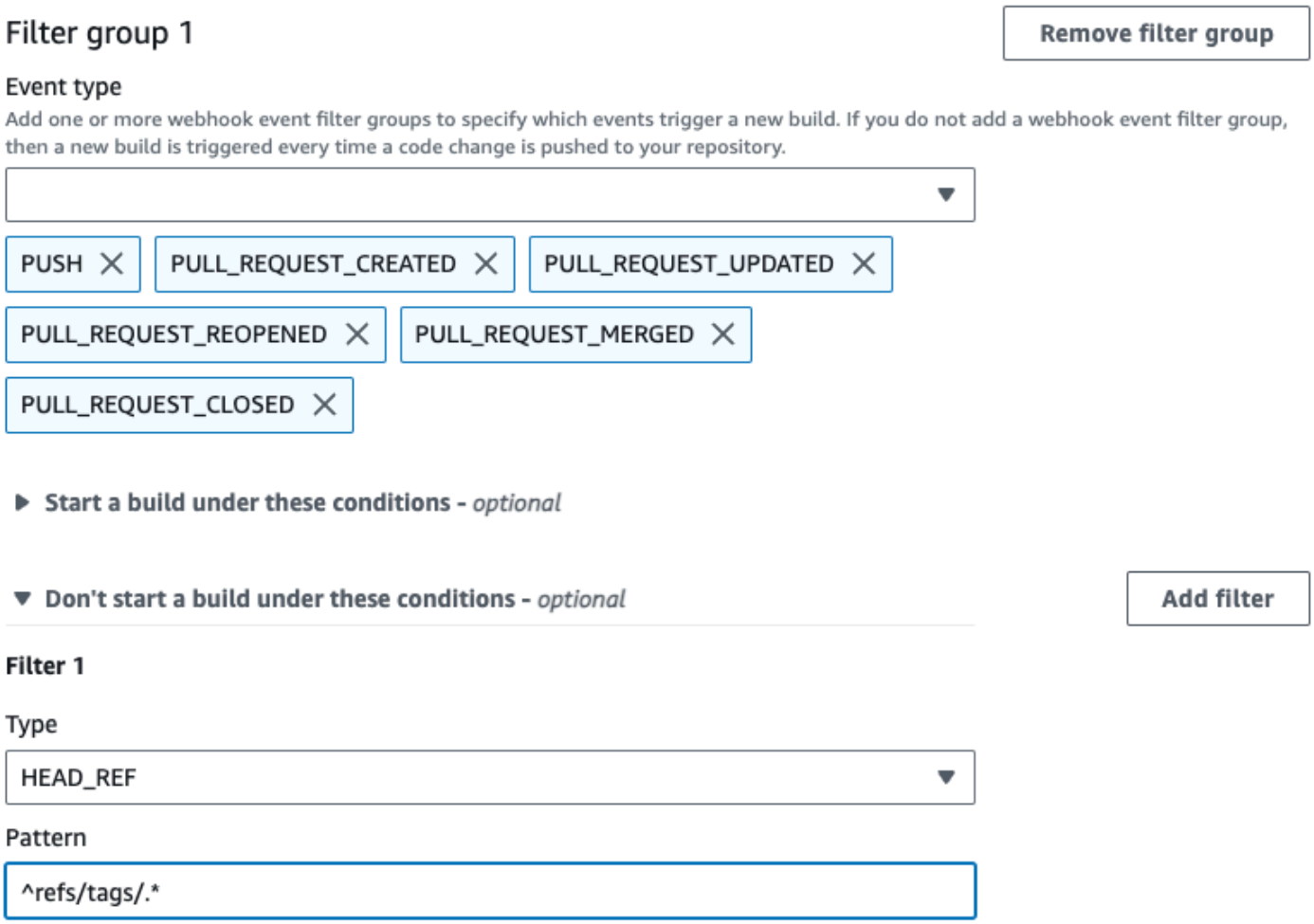


Si utilizamos un ejemplo con dos grupos de filtros de webhooks, la compilación se desencadenará cuando uno de los grupos o los dos se evalúen como true:

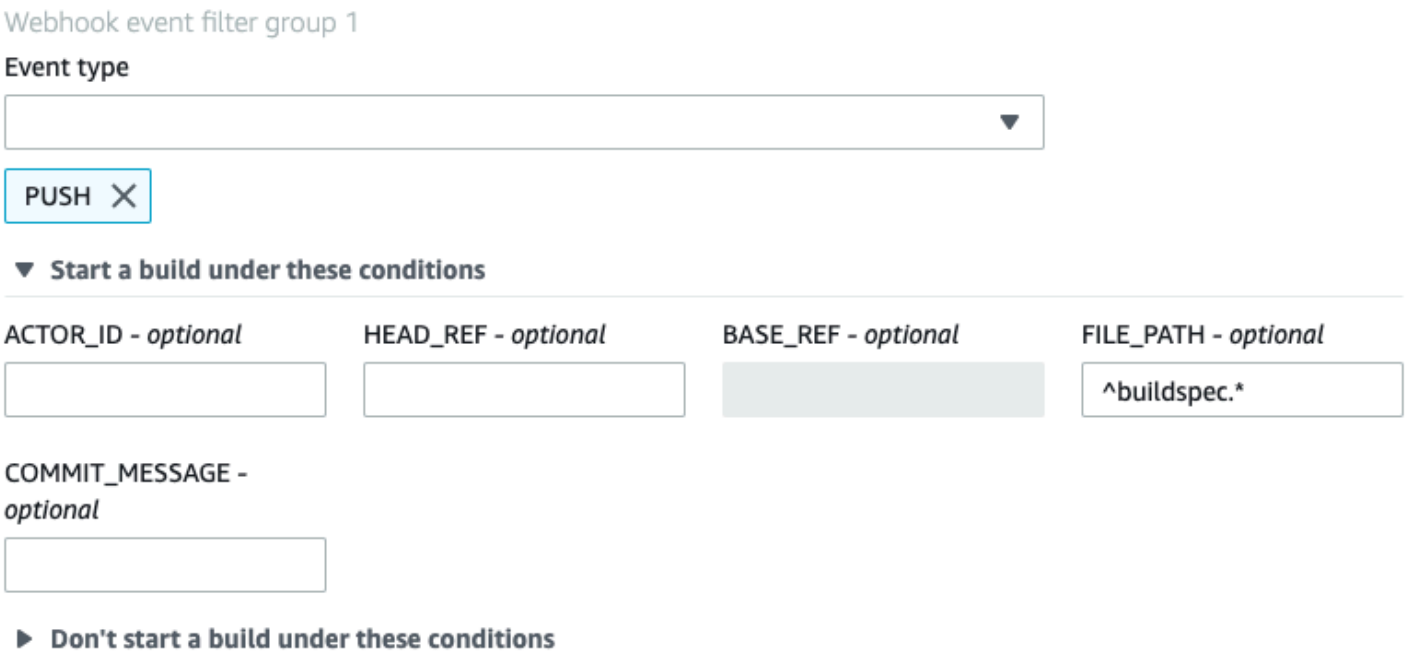
- El primer grupo de filtros especifica las solicitudes de extracción que se crean, actualizan o vuelven a abrirse en ramificaciones con nombres de referencia de Git que coinciden con la expresión regular `^refs/heads/main$` y las referencias de encabezado que coinciden con `^refs/heads/branch1$`.
- 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
PULL_REQUEST_UPDATED
PULL_REQUEST_REOPENED
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
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

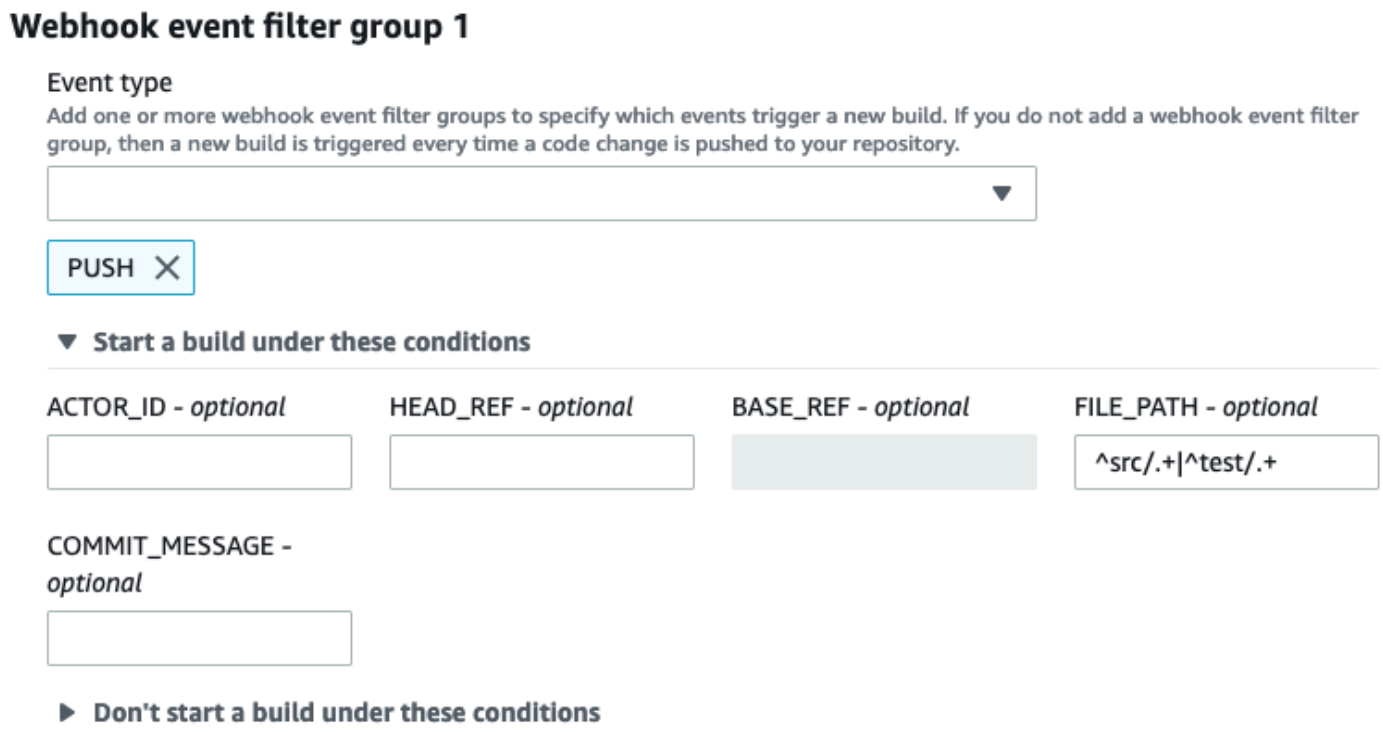
En este ejemplo, un grupo de filtros de webhooks desencadena una compilación de todas las solicitudes excepto los eventos de etiquetas.



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.*`.

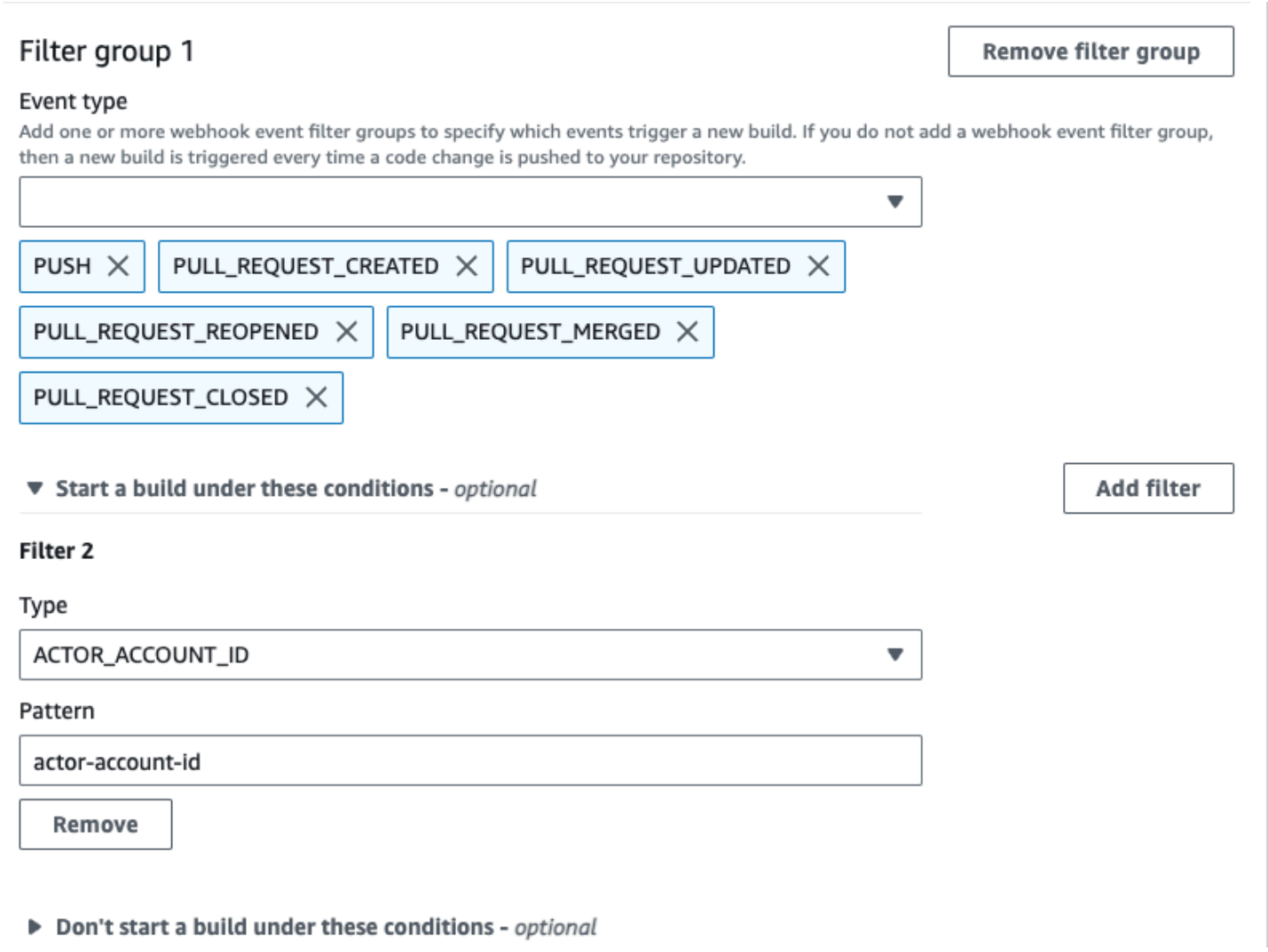


En este ejemplo, un grupo de filtros de webhook activa una compilación solo cuando se cambian archivos en las carpetas `src` o `test`.

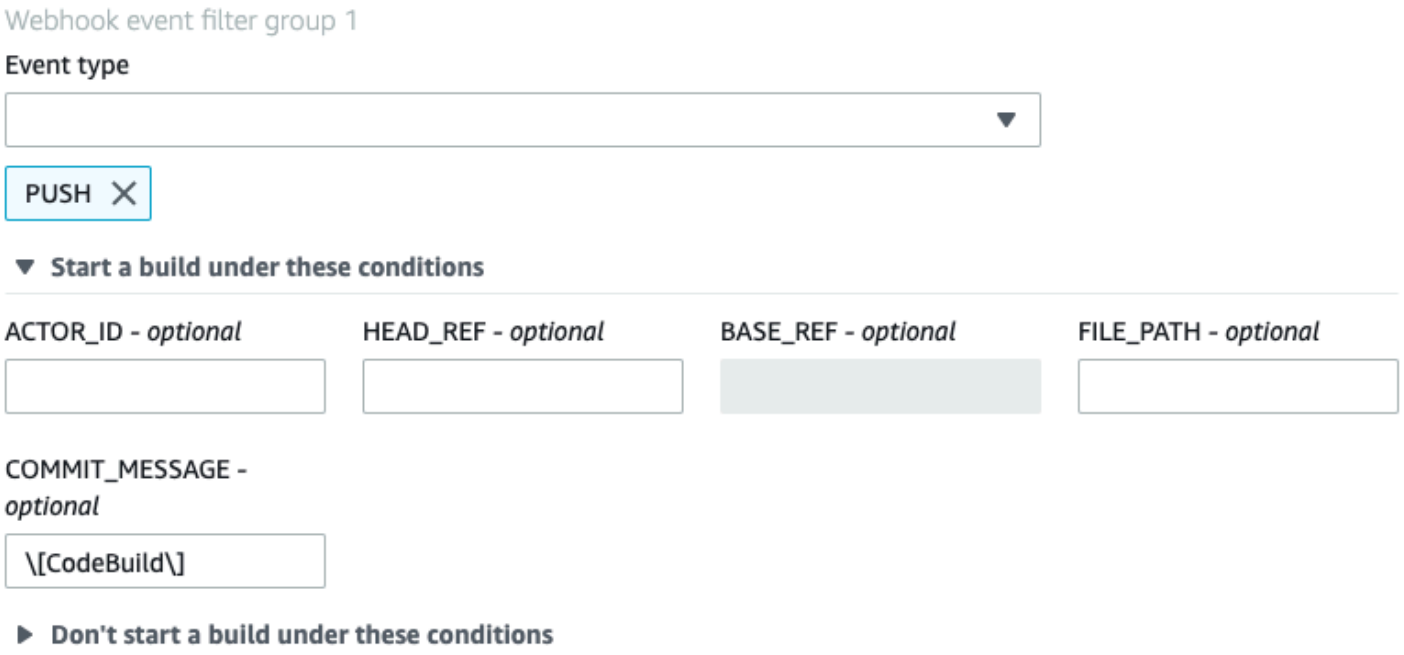


En este ejemplo, un grupo de filtros de webhooks activa una compilación solo cuando un usuario específico GitHub o de GitHub Enterprise Server realiza un cambio con un identificador de cuenta que coincide con la expresión `actor-account-id` regular.

> **Note**
>
> Para obtener información sobre cómo encontrar el ID de su GitHub cuenta https://api.github.com/users/*user-name*, consulte ¿Dónde *user-name* está su nombre GitHub de usuario?

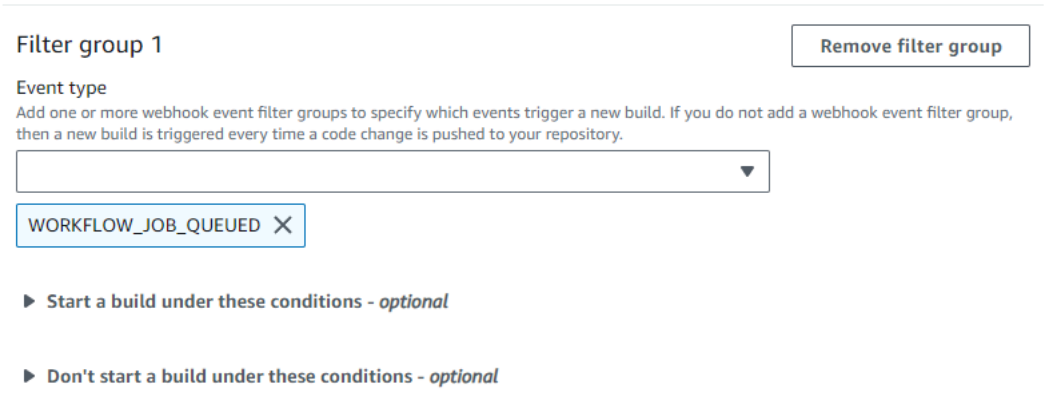


En este ejemplo, un grupo de filtros de webhook desencadena una compilación para un evento de inserción cuando el mensaje de confirmación de la cabeza coincide con la expresión regular `\[CodeBuild\]`.

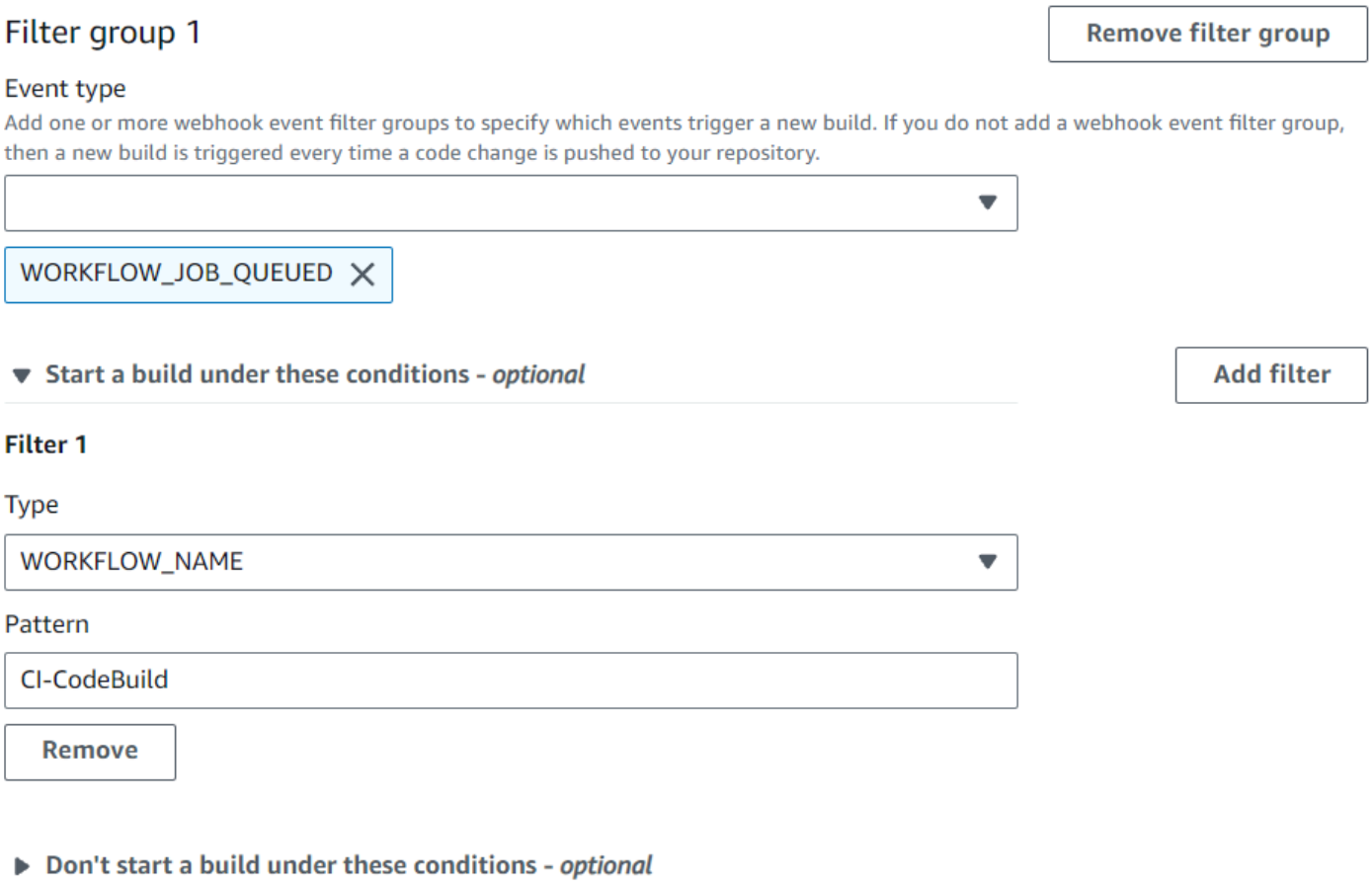


En este ejemplo, un grupo de filtros de webhooks activa únicamente los eventos de trabajo del flujo de trabajo de Build for GitHub Actions.

> **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.



En este ejemplo, un grupo de filtros de webhook desencadena una compilación para un nombre de flujo de trabajo que coincide con la expresión regular `CI-CodeBuild`.



Filtra los eventos de GitHub webhooks (SDK)

Para usar el AWS CodeBuild SDK para filtrar los eventos de webhooks, usa el `filterGroups` campo de la sintaxis de solicitud de los métodos o de la `CreateWebhook` API. `UpdateWebhook` Para obtener más información, consulta WebhookFilter en la CodeBuild Referencia de la API de .

Para obtener más información sobre los eventos de GitHub webhook, consulte. GitHub eventos de webhook

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/.+"
        }
    ]
]
```

Puede crear un filtro que active una compilación solo cuando un usuario específico GitHub o de GitHub Enterprise Server con un identificador `actor-account-id` de cuenta realice un cambio.

> **Note**
>
> Para obtener información sobre cómo encontrar el ID de su GitHub cuenta https://api.github.com/users/*user-name*, consulte Dónde *user-name* está su nombre GitHub de 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 webhooks que active únicamente trabajos de flujo de trabajo de compilación para GitHub acciones, inserta lo siguiente en la sintaxis de la solicitud:

```
"filterGroups": [
   [
        {
            "type": "EVENT",
            "pattern": "WORKFLOW_JOB_QUEUED"
        }
    ]
]
```

Filtrar eventos de GitHub webhooks (CloudFormation)

Para usar una CloudFormation plantilla para filtrar los eventos de webhooks, usa la propiedad del AWS CodeBuild `FilterGroups` proyecto.

Para obtener más información sobre los eventos de GitHub webhook, consulte. GitHub eventos de webhook

La siguiente YAML-formatted parte de una CloudFormation plantilla 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 un GitHub usuario que no tiene un identificador de cuenta cree o actualice las solicitudes de extracción en ramas con nombres `^refs/heads/main$` de referencia de Git que coincidan con la expresión regular`12345`.
- El segundo grupo de filtros especifica las solicitudes de inserción que se crean en archivos cuyos nombres coinciden con la expresión regular `READ_ME` 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 del 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: READ_ME
            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 webhooks grupales

Puedes usar webhooks de CodeBuild GitLab grupo para iniciar compilaciones a partir de eventos de webhook desde cualquier repositorio de un grupo. GitLab Los webhooks de grupo funcionan con cualquiera de los tipos de eventos de GitLab webhook existentes y se pueden configurar añadiendo una configuración de ámbito al crear un webhook. CodeBuild También puedes usar webhooks grupales para configurar GitLab ejecutores autohospedados con el fin de recibir `WORKFLOW_JOB_QUEUED` eventos de varios repositorios dentro CodeBuild de un mismo proyecto.

Temas

- Configura un webhook grupal GitLab
- Filtra los eventos GitLab de webhooks grupales (consola)
- Filtrar GitLab eventos grupales de webhooks (CloudFormation)

Configura un webhook grupal GitLab

Los pasos generales para configurar un GitLab webhook grupal son los siguientes. Para obtener más información sobre los GitLab webhooks grupales, consulte. GitLab webhooks grupales

1. Establezca la ubicación de origen del proyecto en `CODEBUILD_DEFAULT_WEBHOOK_SOURCE_LOCATION`.
2. En la configuración de ámbito del webhook, establezca el ámbito en `GITLAB_GROUP`.
3. Especifique un nombre como parte de la configuración del ámbito del webhook. En el caso de los webhooks de grupo, es el nombre del grupo.

> **Note**
>
> Si el tipo de origen del proyecto es `GITLAB_SELF_MANAGED`, también debe especificar un dominio como parte de la configuración del ámbito del webhook.

4. (Opcional) Si solo quiere recibir eventos de webhook para repositorios específicos de su organización o empresa, puede especificar `REPOSITORY_NAME` como filtro al crear el webhook.
5. Al crear un webhook de grupo, asegúrate de que CodeBuild tiene permisos para crear webhooks a nivel de grupo. GitLab Sin embargo, para hacerlo, puedes usar CodeBuild OAuth. AWS CodeConnections Para obtener más información, consulte GitLab acceder en CodeBuild.

   Ten en cuenta que los webhooks grupales funcionan con cualquiera de los tipos de eventos de GitLab webhook existentes.

Filtra los eventos GitLab de webhooks grupales (consola)

Al crear un GitLab proyecto a través de la consola, selecciona las siguientes opciones para crear un webhook GitLab grupal dentro del proyecto. Para obtener más información sobre los GitLab webhooks grupales, consulte. GitLab webhooks grupales

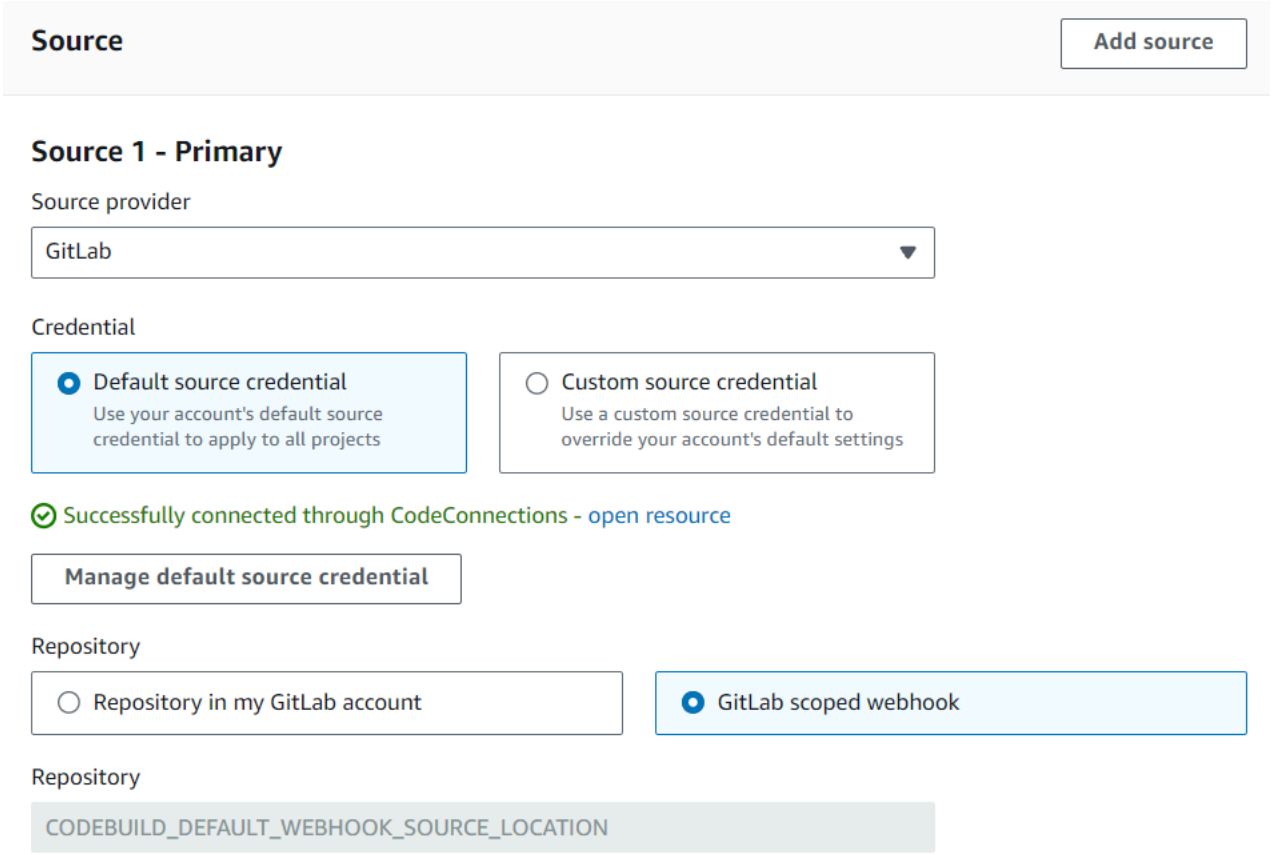
1. Abra la AWS CodeBuild consola en. https://console.aws.amazon.com/codesuite/codebuild/home
2. Cree un proyecto de compilación. Para obtener más información, consulte Creación de un proyecto de compilación (consola) y Ejecutar una compilación (consola).

   - En Source (Código fuente):
     - En Proveedor de código fuente, elija GitLabo GitLabAutogestionado.
     - En Repositorio, elige GitLabwebhook con alcance.

       El GitLab repositorio se configurará automáticamente en`CODEBUILD_DEFAULT_WEBHOOK_SOURCE_LOCATION`, que es la ubicación de origen requerida para los webhooks grupales.

       > **Note**
       >
       > Cuando utilices webhooks de grupo, asegúrate de tener permisos para crear webhooks a nivel de grupo. CodeBuild GitLab Si utilizas una conexión OAuth

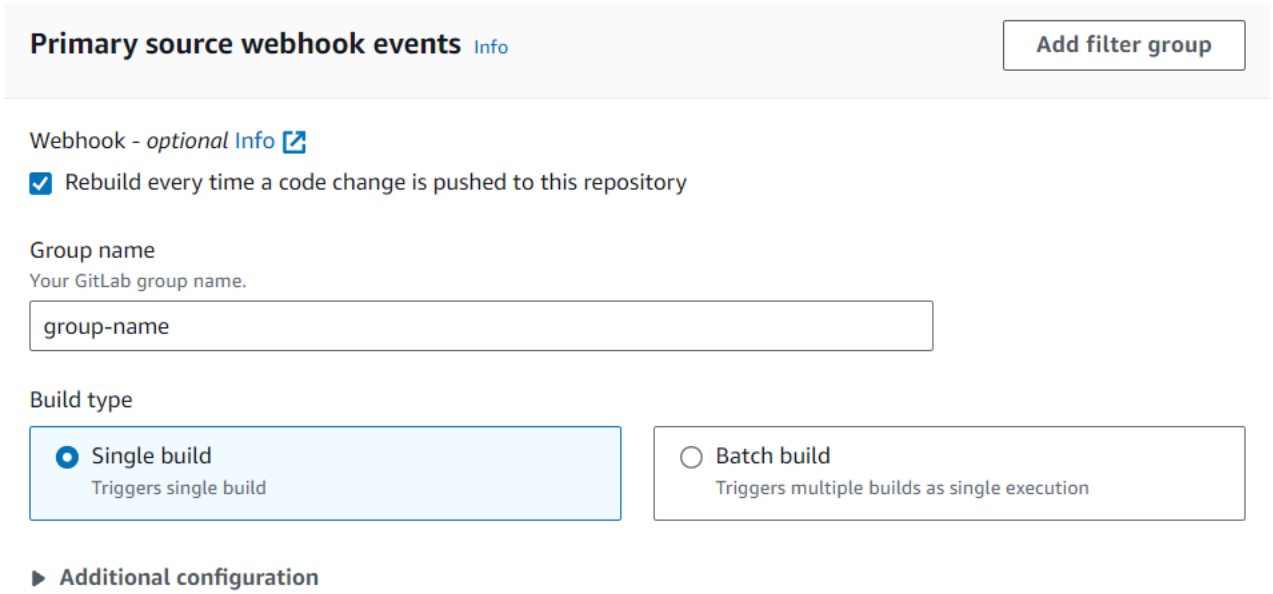
existente, es posible que tengas que volver a generarla para conceder este permiso. CodeBuild



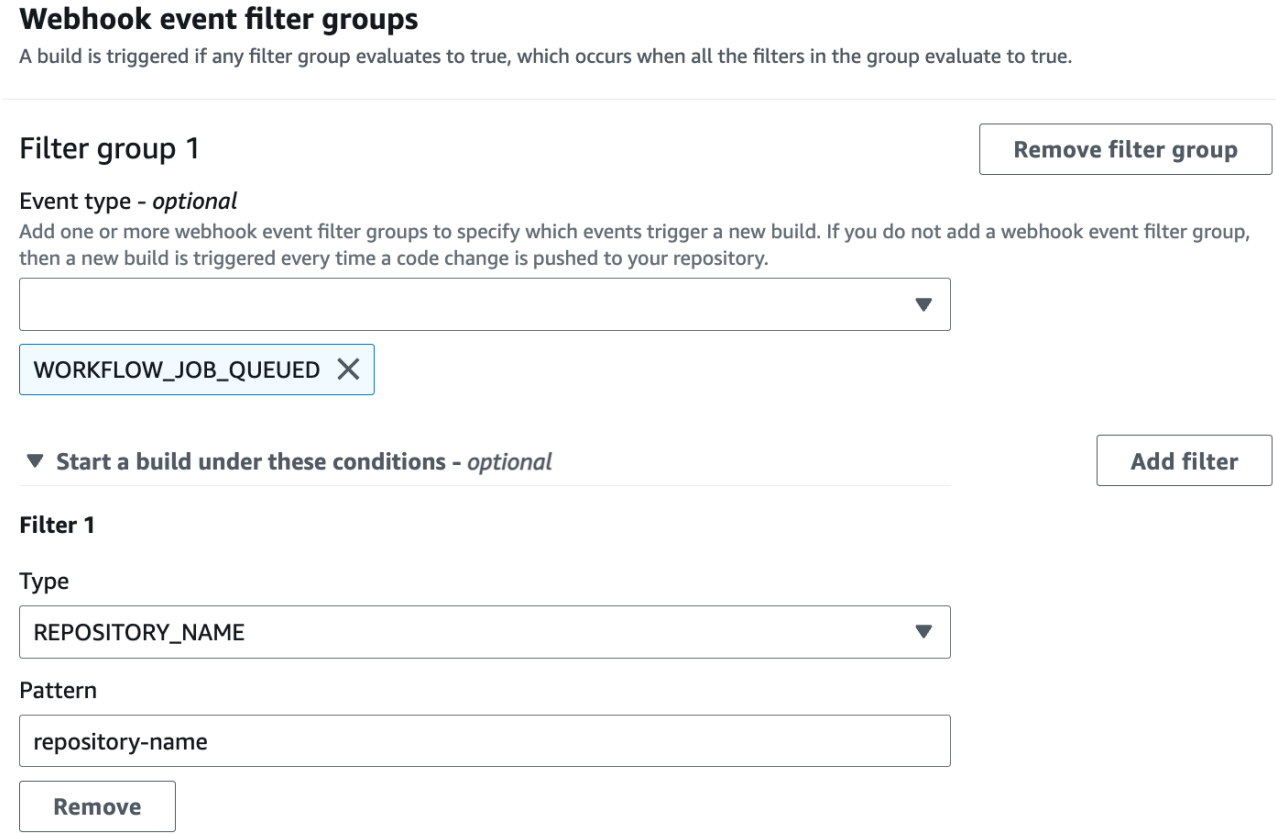
- En Eventos de webhook de origen principal:
  - Escriba el nombre del grupo en Nombre del grupo.

    Si el tipo de origen del proyecto es `GITLAB_SELF_MANAGED`, también debe especificar un dominio como parte de la configuración de grupo del webhook. Por ejemplo, si la URL de su grupo es **`https://domain.com/group/group-name`**, el dominio es **`https://domain.com`**.

    > **Note**
    >
    > Este nombre no puede modificarse una vez que se haya creado el webhook. Para cambiar el nombre, puede eliminar el webhook y volver a crearlo. Si quieres eliminar el webhook por completo, también puedes actualizar la ubicación de origen del proyecto a un repositorio. GitLab



- (Opcional) En Grupos de filtros de eventos de webhook, puede especificar qué eventos quiere que desencadenen una nueva compilación. También puede especificar `REPOSITORY_NAME` como filtro para que solo se desencadenen compilaciones a partir de eventos de webhook de repositorios específicos.



También puedes configurar el tipo de evento como `WORKFLOW_JOB_QUEUED` para configurar corredores autohospedados GitLab . Para obtener más información, consulte Self-managed GitLab se ejecuta en AWS CodeBuild.

3. Continúe con los valores predeterminados y, a continuación, elija Crear el proyecto de compilación.

Filtrar GitLab eventos grupales de webhooks (CloudFormation)

Para usar una CloudFormation plantilla para filtrar eventos de webhook de grupo, usa la propiedad del AWS CodeBuild `ScopeConfiguration` proyecto. Para obtener más información sobre los GitLab webhooks de grupo, consulte. GitLab webhooks grupales

La siguiente YAML-formatted parte de una CloudFormation plantilla crea cuatro grupos de filtros. Juntos, desencadenan una compilación cuando uno o todos 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 cuenta`12345`.

- El segundo grupo de filtros especifica las solicitudes de inserción que se crean en archivos cuyos nombres coinciden con la expresión regular `READ_ME` 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 GitLab CI/CD canalización con un nombre de CI/CD canalización 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: GITLAB
      Location: source-location
    Triggers:
      Webhook: true
      ScopeConfiguration:
        Name: group-name
        Scope: GITLAB_GROUP
      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: READ_ME
          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 webhooks manuales

Puedes configurar los GitLab webhooks manuales para CodeBuild evitar que intenten crear automáticamente un webhook en ellos. GitLab CodeBuild devuelve una URL de carga útil como parte de la llamada para crear el webhook, que se puede usar para crear manualmente el webhook en su interior. GitLab Aunque no CodeBuild esté autorizado a crear un webhook en tu GitLab cuenta, puedes crear manualmente un webhook para tu proyecto de compilación.

Usa el siguiente procedimiento para crear un GitLab webhook manual.

Para crear un webhook GitLab manual

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 el caso del proveedor de origen, elija GitLab.
     - En Repositorio, seleccione Repositorio en mi GitLab cuenta.
     - En URL de repositorio, especifique **`https://gitlab.com/`*`user-name`*`/`*`repository-name`***
   - En Eventos de webhook de origen principal:
     - En Webhook (opcional), seleccione Volver a compilar cada vez que un cambio de código se envía a este repositorio.
     - Selecciona Configuración adicional y, para Creación manual (opcional), elige Crear manualmente un webhook para este repositorio en la GitLab consola. .

3. Continúe con los valores predeterminados y, a continuación, elija Crear el proyecto de compilación. Anote los valores de URL de carga y Secreto, ya que los utilizará más adelante.
4. Abre la GitLab consola en `https://gitlab.com/user-name/repository-name/-/hooks` y elige Agregar nuevo webhook.
   - En URL, especifique el valor de la URL de carga útil que ha anotado anteriormente.
   - En Token de secreto, especifique el valor de Secreto que ha anotado anteriormente.
   - Configura los eventos individuales a los que se enviará la carga útil de un webhook. CodeBuild En Disparador, elija uno de los siguientes eventos: Eventos de inserción, Eventos de solicitud de combinación, Eventos de lanzamientos y Eventos de trabajo. Para obtener más información sobre los tipos de eventos que admite CodeBuild, consulte. GitLab eventos de webhook
5. Elija Add webhook.

GitLab eventos de webhook

Puedes usar los grupos de filtros de webhooks 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, puedes elegir uno o más de los siguientes eventos:`PUSH`,,`PULL_REQUEST_CREATED`,`PULL_REQUEST_UPDATED`,`PULL_REQUEST_MERGED`, `PULL_REQUEST_REOPENED` `PULL_REQUEST_CLOSEDRELEASED`, y. `WORKFLOW_JOB_QUEUED`

El tipo de evento de webhook está en su encabezado en el campo `X-GitLab-Event`. En la siguiente tabla se muestra cómo los valores del encabezado `X-GitLab-Event` se asignan a los tipos de eventos. En el evento de webhook `Merge Request Hook`, el valor `object_atttributes.action` de la carga útil contendrá información adicional sobre el tipo de solicitud de combinación.

Valor del encabezado **X-GitLab-Event**	**object_atttributes.action**	Tipo de evento
`Push Hook`	N/A	`PUSH`
`Merge Request Hook`	open	`PULL_REQUEST_CREATED`
`Merge Request Hook`	actualización	`PULL_REQUEST_UPDATED`
`Merge Request Hook`	merge	`PULL_REQUEST_MERGED`
`Merge Request Hook`	reabrir	`PULL_REQUEST_REOPENED`
`Merge Request Hook`	close	`PULL_REQUEST_CLOSED`
`Release Hook`	crear, actualizar	`RELEASED`
`Job Hook`	N/A	`WORKFLOW_JOB_QUEUED`

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.

CodeBuild evalúa los patrones de filtro mediante la sintaxis de expresiones RE2 regulares, que no admite referencias anticipadas, retrospectivas ni grupos atómicos. Para obtener más información, consulte la página de RE2 sintaxis del sitio web. GitHub

`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.

> **Note**
>
> El `FILE_PATH` filtro se evalúa comparándolo con los 100 primeros archivos modificados en una solicitud push o pull. Para asegurarte de que el filtro se aplica según lo previsto, mantén el número de archivos modificados dentro de este límite. En el caso de los eventos de solicitud de extracción, también recomendamos usar una política de creación de solicitudes de extracción para controlar qué solicitudes de extracción pueden iniciar una compilación.

`COMMIT_MESSAGE`

Un webhook desencadena una compilación cuando el mensaje de confirmación del encabezado coincide con la expresión regular.

`WORKFLOW_NAME`

Un webhook desencadena una compilación cuando el nombre del flujo de trabajo coincide con el patrón de expresión regular.

> **Note**
>
> Puedes encontrar la carga útil de los webhooks en la configuración de los webhooks de tu repositorio. GitLab

Temas

- Filtra los eventos de GitLab webhooks (consola)
- Filtrar eventos de GitLab webhooks (SDK)
- Filtrar eventos de GitLab webhook (CloudFormation)

Filtra los eventos de GitLab webhooks (consola)

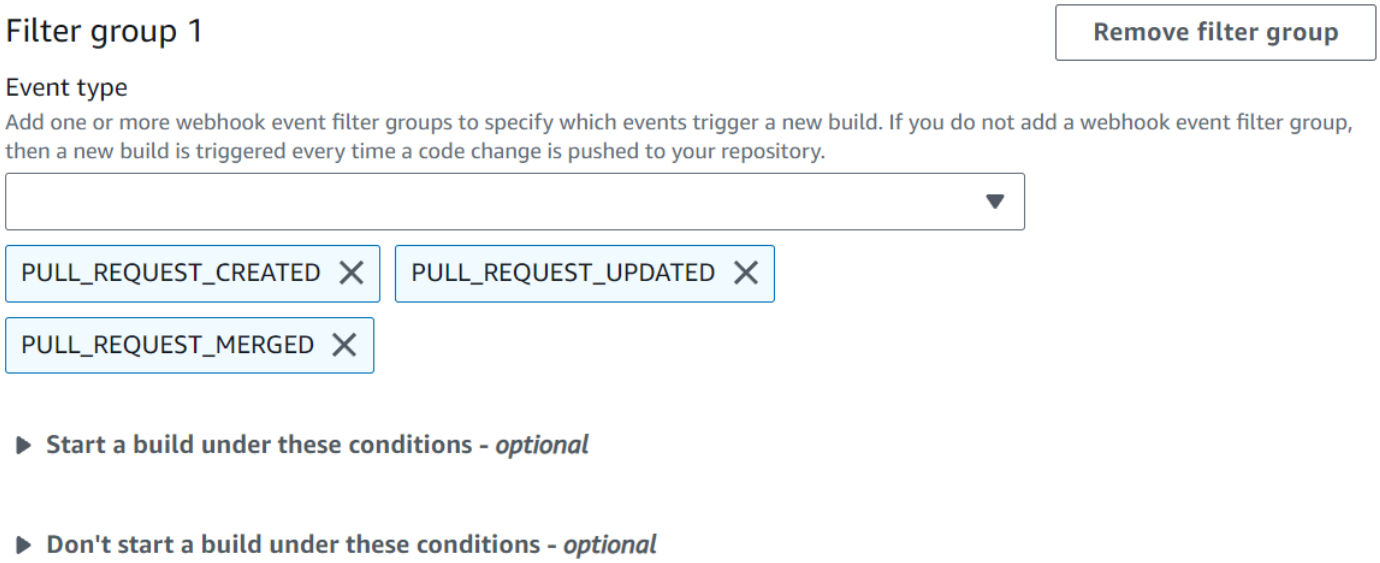
Siga las instrucciones siguientes para filtrar los eventos Consola de administración de AWS de webhooks. Para obtener más información sobre los eventos de GitLab webhook, consulte. GitLab eventos de webhook

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 consulte Creación de un proyecto de compilación (consola) y WebhookFilter en la Referencia de la API de AWS CodeBuild .

En este ejemplo, un grupo de filtros de webhooks desencadena una compilación únicamente con solicitudes de extracción:



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

HEAD_REF

Pattern

^refs/heads/branch1$

Remove

Filter 2

Type

BASE_REF

Pattern

^refs/heads/main

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.

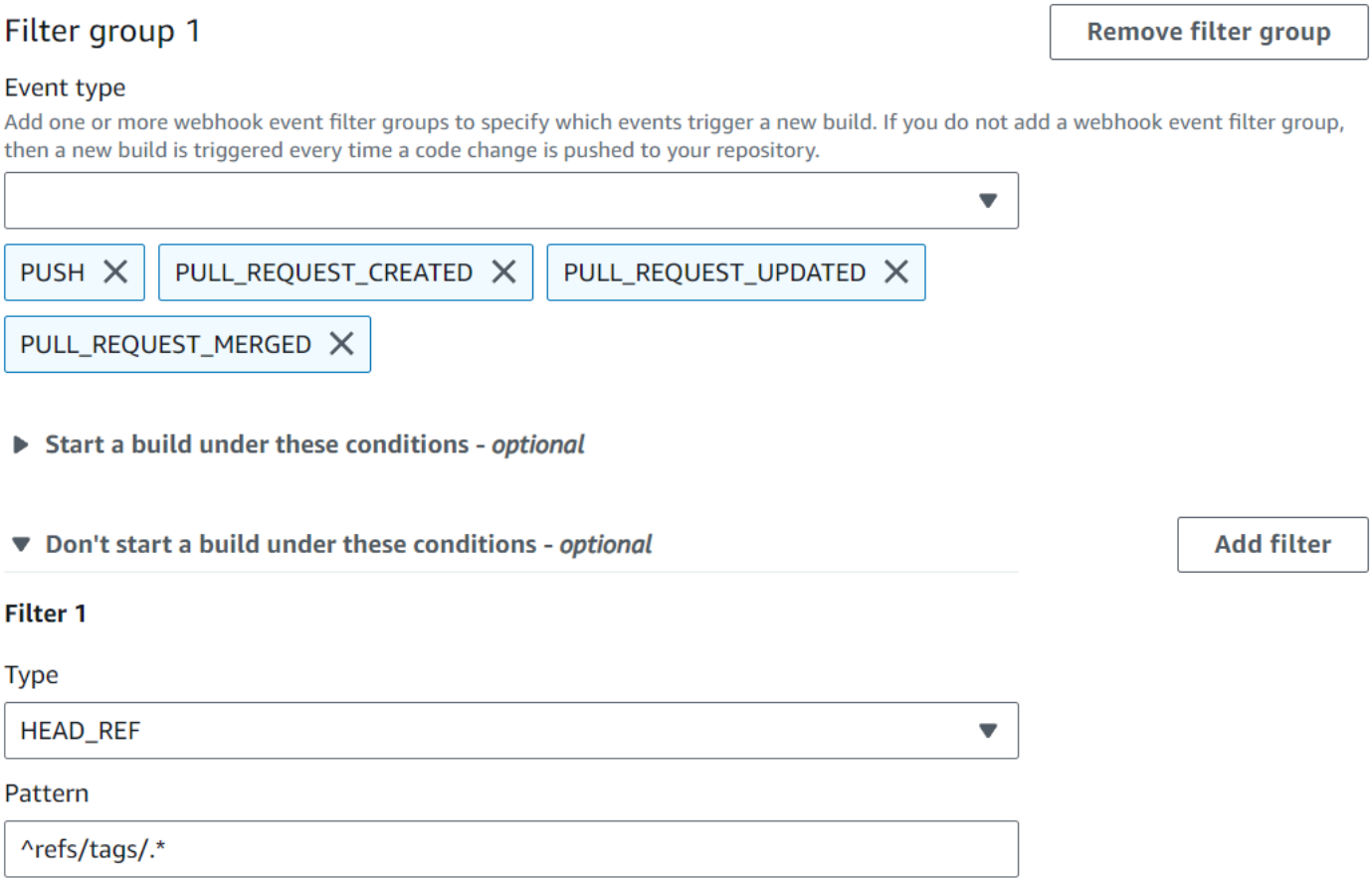
PUSH ✕

▼ **Start a build under these conditions** - *optional*

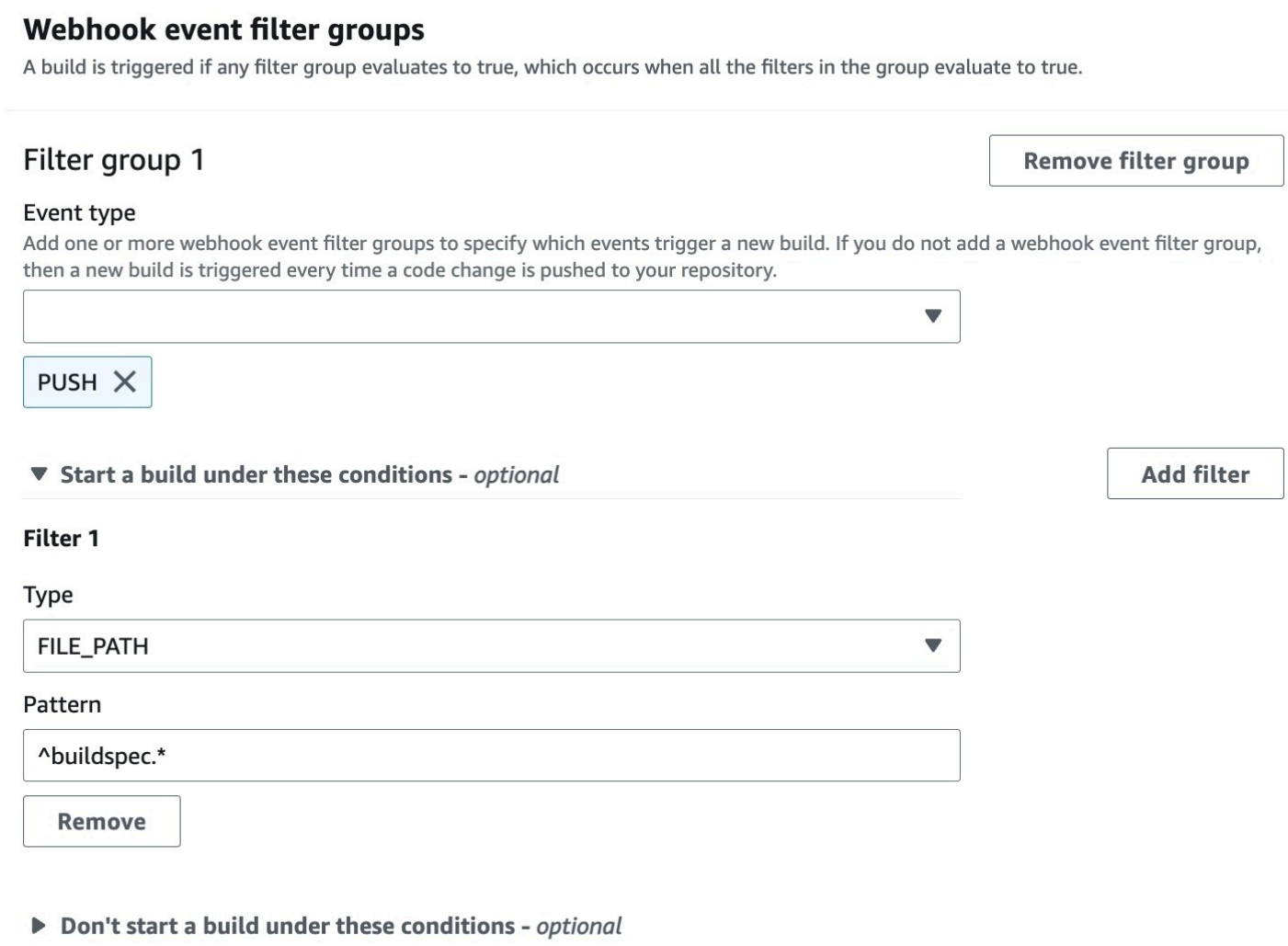
Add filter

Filter 1

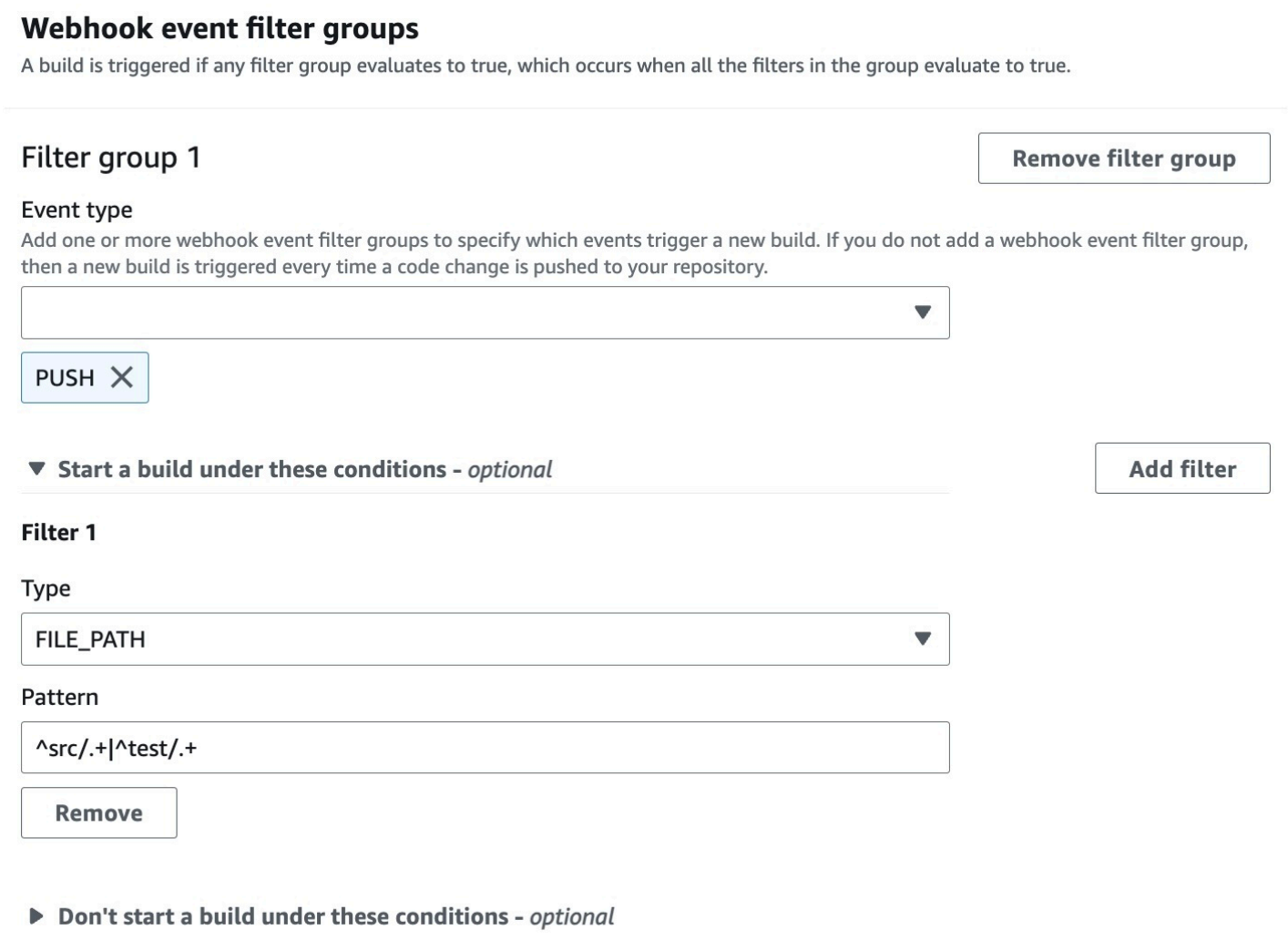
En este ejemplo, un grupo de filtros de webhooks desencadena una compilación de todas las solicitudes excepto los eventos de etiquetas.



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.*`.

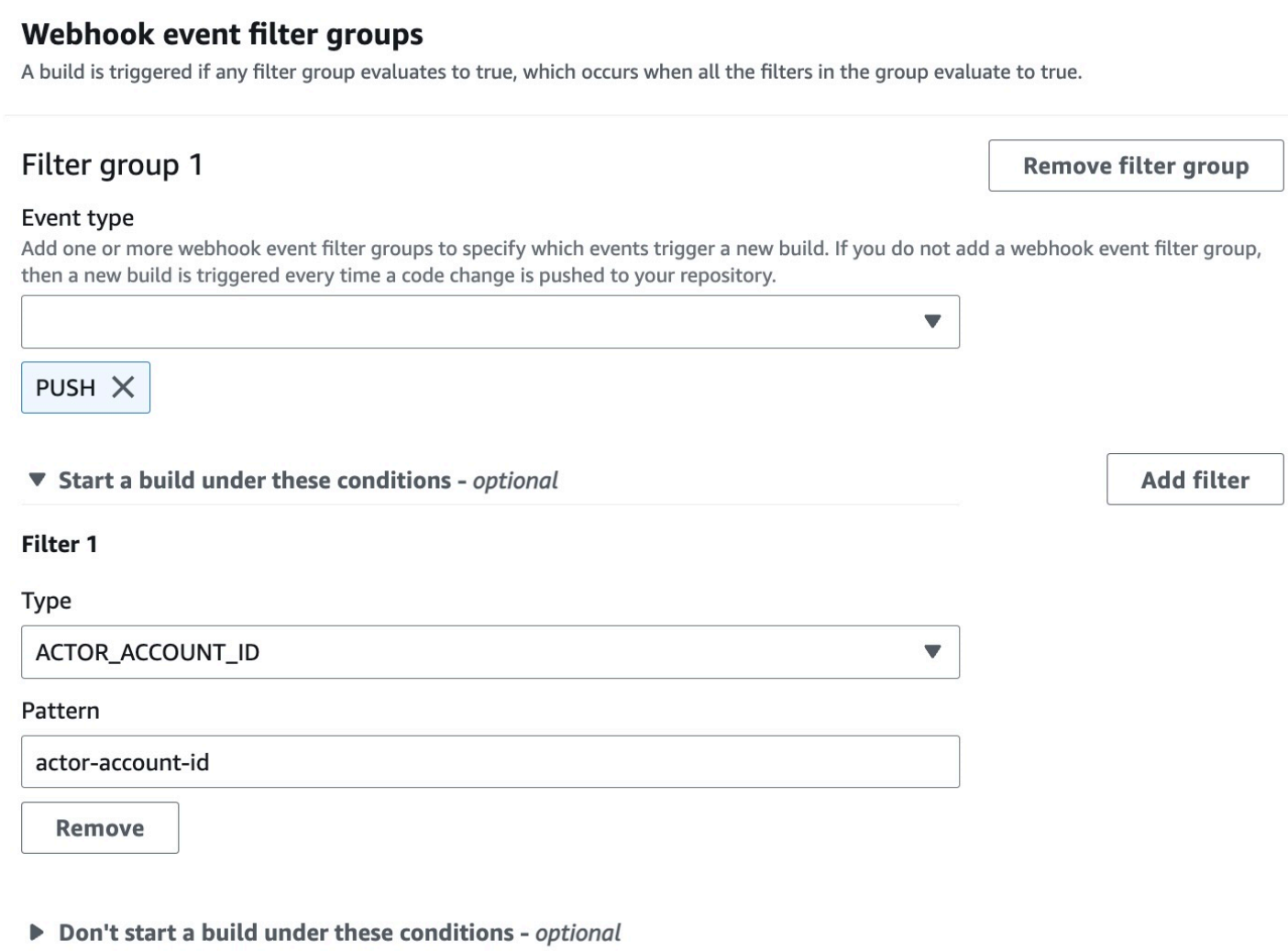


En este ejemplo, un grupo de filtros de webhook activa una compilación solo cuando se cambian archivos en las carpetas `src` o `test`.

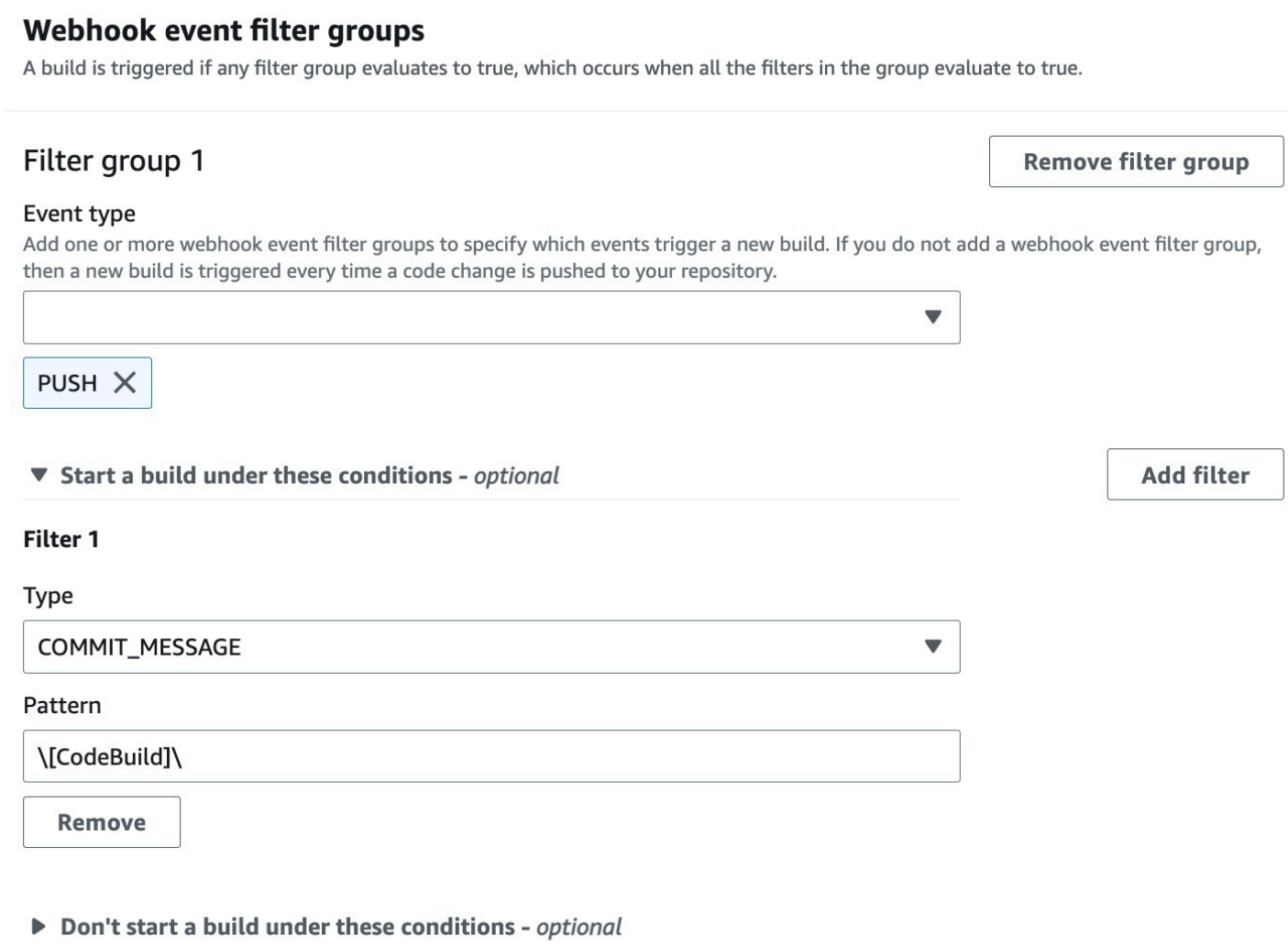


En este ejemplo, un grupo de filtros de webhooks activa una compilación solo cuando un GitLab usuario que no tiene un identificador de cuenta que coincida con la expresión regular realiza un cambio. `actor-account-id`

> **Note**
>
> Para obtener información sobre cómo encontrar el ID de tu GitLab cuenta https://api.github.com/users/*`user-name`*, consulta ¿Dónde *`user-name`* está tu nombre GitLab de usuario?



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 `\[CodeBuild\]`.



Filtrar eventos de GitLab webhooks (SDK)

Para usar el AWS CodeBuild SDK para filtrar los eventos de webhooks, usa el `filterGroups` campo de la sintaxis de solicitud de los métodos o de la `CreateWebhook` API. `UpdateWebhook` Para obtener más información, consulta WebhookFilter en la CodeBuild Referencia de la API de .

Para obtener más información sobre los eventos de GitLab webhook, consulte. GitLab eventos de webhook

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 identificador `actor-account-id` de cuenta realice un cambio.

> **Note**
>
> Para obtener información sobre cómo encontrar el ID de tu GitLab cuenta https://api.github.com/users/*user-name*, consulta ¿Dónde *user-name* está tu nombre GitLab de 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 (CloudFormation)

Para usar una CloudFormation plantilla para filtrar los eventos de webhooks, usa la propiedad del AWS CodeBuild `FilterGroups` proyecto. Para obtener más información sobre los eventos de GitLab webhook, consulte. GitLab eventos de webhook

La siguiente YAML-formatted parte de una CloudFormation plantilla 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 un GitLab usuario que no tiene un identificador de cuenta cree o actualice las solicitudes de extracción en ramas con nombres `^refs/heads/main$` de referencia de Git que coincidan con la expresión regular`12345`.
- 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 `\[CodeBuild\]`.
- El cuarto grupo de filtros especifica una solicitud de trabajo del 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: 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: \[CodeBuild\]
        - - Type: EVENT
            Pattern: WORKFLOW_JOB_QUEUED
          - Type: WORKFLOW_NAME
            Pattern: \[CI-CodeBuild\]
```

Webhooks manuales de Buildkite

Actualmente, CodeBuild requiere que todos los webhooks de Buildkite se creen manualmente. CodeBuilddevuelve una URL de carga útil como parte de la llamada para crear el webhook, que se puede usar para crear manualmente el webhook en Buildkite.

Utilice el siguiente procedimiento para crear un webhook manual de Buildkite.

Para crear un proyecto con un webhook CodeBuild

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).
3. En Configuración del proyecto, seleccione Proyecto de ejecutor.

   En Ejecutor:

   - En Proveedor de ejecutor, elija Buildkite.
   - En Token de agente de Buildkite, seleccione Crear un nuevo token de agente mediante la página de creación de secreto. Se te pedirá que crees un nuevo secreto en AWS Secrets Manager con un valor secreto igual al token de agente de Buildkite que generaste anteriormente.
   - (Opcional) Si quieres usar credenciales CodeBuild administradas para tu trabajo, selecciona el proveedor del repositorio de origen de tu trabajo en las opciones de credenciales de origen de Buildkite y verifica que las credenciales estén configuradas para tu cuenta. Además, verifique que la canalización de Buildkite utilice Extracción mediante HTTPS.
4. - En Environment (Entorno):
     - Elija una Imagen del entorno y una Computación compatibles. Ten en cuenta que tienes la opción de anular la configuración de la imagen y la instancia mediante el uso de una etiqueta en el YAML de tu GitHub flujo de trabajo de Actions. Para obtener más información, consulte Paso 2: actualiza el flujo de trabajo de GitHub Actions (YAML)
   - En Buildspec:
     - Tenga en cuenta que su especificación de compilación se ignorará a menos que se agregue `buildspec-override:true` como etiqueta. En su lugar, la CodeBuild anularás para usar comandos que configurarán el ejecutor autohospedado.
5. Continúe con los valores predeterminados y, a continuación, elija Crear el proyecto de compilación.
6. Guarde los valores de URL de carga útil y Secreto de la ventana emergente Crear webhook. Siga las instrucciones de la ventana emergente para crear un nuevo webhook de organización de Buildkite.

Aprobación de comentarios de solicitudes de extracción

CodeBuild admite políticas de creación de solicitudes de incorporación de cambios que proporcionan un control adicional sobre las compilaciones activadas por solicitudes de incorporación de cambios. No conviene compilar automáticamente solicitudes de extracción de usuarios desconocidos hasta que se puedan revisar sus cambios. Esta característica permite solicitar a uno de los miembros del equipo que primero revise el código y, a continuación, ejecute la canalización. Esto se suele utilizar como medida de seguridad cuando se crea código enviado por colaboradores desconocidos.

Las políticas de creación de solicitudes de extracción te permiten controlar cuándo se CodeBuild activan las compilaciones de las solicitudes de extracción en función de los permisos y el estado de aprobación del colaborador. Esto es algo particularmente importante en los repositorios públicos o los repositorios que aceptan contribuciones de colaboradores externos.

Cuando está habilitada, esta característica garantiza que las compilaciones solo se desencadenen para solicitudes de extracción cuando:

- La solicitud de extracción la crea un colaborador de confianza.
- Un colaborador de confianza aprueba la solicitud de extracción publicando un comentario específico.

Funcionamiento

Colaboradores de confianza

Un colaborador de confianza es un usuario cuya función actual en el sistema de control de código fuente se establece en la política basada en solicitudes de extracción como función de aprobador. Cuando un colaborador de confianza crea una solicitud de cambios, CodeBuild activa la compilación automáticamente, manteniendo el comportamiento actual.

Colaboradores que no son de confianza

Un colaborador que no es de confianza es un usuario cuya función no está establecida en la lista de funciones de aprobador. Cuando un colaborador que no es de confianza crea una solicitud de cambios:

1. CodeBuild marca el estado de la compilación como «Fallido» con el mensaje «Se requiere la aprobación de la solicitud de extracción para iniciar una compilación».

2. Un colaborador de confianza debe revisar los cambios y publicar un comentario con `/codebuild_run(<SHA_OF_THE_LATEST_COMMIT>)` para desencadenar la compilación. Por ejemplo, `/codebuild_run(046e8b67481d53bdc86c3f6affdd5d1afae6d369)`.
3. CodeBuild valida los permisos del comentarista y activa la compilación si se aprueba.
4. Los resultados de la compilación se devuelven a la página de solicitudes de extracción.

Sintaxis de aprobación de comentarios

Los colaboradores de confianza pueden aprobar compilaciones mediante los siguientes formatos de comentario:

- `/codebuild_run(046e8b67481d53bdc86c3f6affdd5d1afae6d369)`: desencadena la compilación en el SHA de confirmación especificado.

Configuración

Comportamiento predeterminado

La política de creación de solicitudes de extracción está habilitada de forma predeterminada para todos los proyectos recién creados CodeBuild .

Parámetros de la API

La política de compilación de solicitudes de extracción se configura mediante el parámetro `PullRequestBuildPolicy` en las siguientes acciones:

- `CreateWebhook`
- `UpdateWebhook`

Estructura de **`PullRequestBuildPolicy`**

```
{
    "requiresCommentApproval": "string",
    "approverRoles": ["string", ...]
}
```

`requiresCommentApproval`

Especifica cuándo se requiere aprobación basada en comentarios antes de desencadenar una compilación a partir de solicitudes de extracción. Esta configuración determina si las compilaciones se ejecutan automáticamente o si requieren una aprobación explícita mediante comentarios.

Tipo: cadena

Valores válidos:

- `DISABLED`: las compilaciones se desencadenan automáticamente sin requerir aprobación de comentarios.
- `FORK_PULL_REQUESTS`: solo las solicitudes de extracción de repositorios bifurcados requieren aprobación de comentarios (a menos que colaborador sea uno de los roles de aprobador).
- `ALL_PULL_REQUESTS`: todas las solicitudes de extracción requieren aprobación de comentarios antes de ejecutar compilaciones (a menos que colaborador sea uno de los roles de aprobador). Este es el valor predeterminado.

`approverRoles`

Lista de roles de repositorio que tienen privilegios de aprobación para compilaciones de solicitudes de extracción cuando se requiere aprobación de comentarios. Solo los usuarios con estos roles pueden proporcionar aprobaciones de comentarios válidas. Si un colaborador de solicitudes de extracción tiene uno de estos roles, las compilaciones de solicitudes de extracción se desencadenarán automáticamente.

Tipo: matriz de cadenas

Valores válidos para los GitHub proyectos (los valores se asignan a los GitHub roles):

- `GITHUB_ADMIN`: administradores del repositorio
- `GITHUB_MAINTAIN`: mantenedores del repositorio
- `GITHUB_WRITE`: usuario con permisos de escritura
- `GITHUB_TRIAGE`: usuario con permisos de clasificación
- `GITHUB_READ`: usuario con permisos de lectura
- Valor predeterminado: `["GITHUB_ADMIN", "GITHUB_MAINTAIN", "GITHUB_WRITE"]`

Valores válidos para los GitLab proyectos (los valores se asignan a los GitLab roles):

- `GITLAB_OWNER`: propietario del repositorio
- `GITLAB_MAINTAINER`: mantenedor del repositorio
- `GITLAB_DEVELOPER`: usuario con permisos de desarrollador
- `GITLAB_REPORTER`: usuario con permisos de informador
- `GITLAB_PLANNER`: usuario con permisos de planificador
- `GITLAB_GUEST` : usuario con permisos de invitado

- Valor predeterminado: `["GITLAB_OWNER", "GITLAB_MAINTAINER", "GITLAB_DEVELOPER"]`

Valores válidos para proyectos de Bitbucket (los valores se asignan a roles de Bitbucket):

- `BITBUCKET_ADMIN` : administrador del repositorio
- `BITBUCKET_WRITE`: usuario con permisos de escritura
- `BITBUCKET_READ` : usuario con permisos de lectura
- Valor predeterminado: `["BITBUCKET_ADMIN", "BITBUCKET_WRITE"]`

Funciones GitHub empresariales personalizadas

CodeBuild asigna los roles de repositorio GitHub empresariales (GHE) personalizados a `GITHUB_*` valores estándar en función del nivel de permiso más alto otorgado al rol personalizado. CodeBuild evalúa los permisos de mayor a menor, y el primer permiso habilitado determina el rol asignado.

- `admin` → `GITHUB_ADMIN`
- `maintain` → `GITHUB_MAINTAIN`
- `push`(escribir) → `GITHUB_WRITE`
- `triage` → `GITHUB_TRIAGE`
- `pull`(leer) → `GITHUB_READ`

Por ejemplo, un rol personalizado que otorga `push` tanto `triage` permisos como a los que se asigna `GITHUB_WRITE` porque `push` tiene más privilegios que`triage`.

Si un rol personalizado no tiene ninguno de los permisos reconocidos (admin, keep, push, triage, pull all desactivado), el usuario no cumple con ningún rol de aprobador y se lo considera un colaborador que no es de confianza.

Las GitHub funciones estándar (administración, mantenimiento, escritura, clasificación, lectura) se asignan directamente a sus `GITHUB_*` valores correspondientes sin necesidad de una resolución basada en permisos.

Ejemplos

Habilitación de aprobación de comentarios para todas las solicitudes de extracción

Para usar el AWS CodeBuild SDK para habilitar o deshabilitar la política de creación de solicitudes de extracción para un webhook, usa el `pullRequestBuildPolicy` campo de la

sintaxis de solicitud de los métodos o de la API. `CreateWebhook` `UpdateWebhook` Para obtener más información, consulta WebhookFilter en la CodeBuild Referencia de la API de .

Los usuarios con GitHub las funciones de administrador, mantenimiento y escritura serán tratados como colaboradores de confianza.

```
"pullRequestBuildPolicy": {
    "requiresCommentApproval": "ALL_PULL_REQUESTS",
    "approverRoles": ["GITHUB_ADMIN", "GITHUB_MAINTAIN", "GITHUB_WRITE"]
}
```

Habilitación de la aprobación de comentarios solo para administradores y mantenedores de repositorios

Los usuarios con GitHub las funciones de administrador y mantenimiento serán tratados como colaboradores de confianza.

```
"pullRequestBuildPolicy": {
    "requiresCommentApproval": "FORK_PULL_REQUESTS",
    "approverRoles": ["GITHUB_ADMIN", "GITHUB_MAINTAIN"]
}
```

Deshabilitación de la aprobación de comentarios

```
"pullRequestBuildPolicy": {
    "requiresCommentApproval": "DISABLED"
}
```

AWS CloudFormation

Si quieres usar una AWS CloudFormation plantilla para habilitar o deshabilitar la política de creación de Pull Request para un webhook, usa la PullRequestBuildPolicy propiedad. La siguiente YAML-formatted parte de una AWS CloudFormation plantilla crea un proyecto con un webhook que tiene la política de creación de solicitudes de extracción habilitada para todas las solicitudes de extracción. Los roles de mantenimiento y administración se especifican como aprobadores.

```
CodeBuildProject:
  Type: AWS::CodeBuild::Project
  Properties:
```

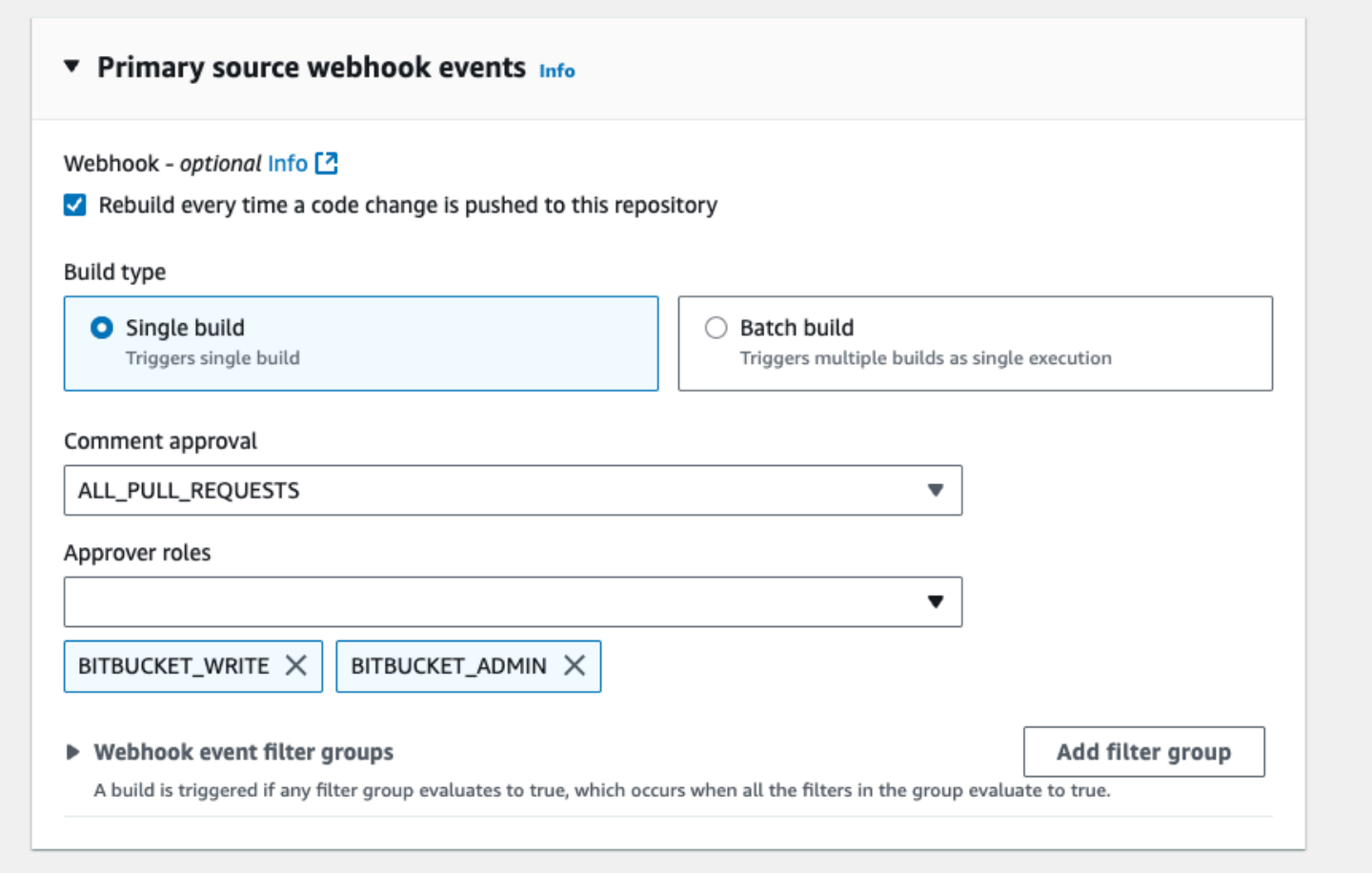
```
    Name: MyProject
    ServiceRole: service-role
    Artifacts:
      Type: NO_ARTIFACTS
    Environment:
      Type: LINUX_CONTAINER
      ComputeType: BUILD_GENERAL1_SMALL
      Image: aws/codebuild/standard:5.0
    Source:
      Type: BITBUCKET
      Location: source-location
    Triggers:
      Webhook: true
      FilterGroups:
        - - Type: EVENT
            Pattern: PULL_REQUEST_CREATED,PULL_REQUEST_UPDATED
          - Type: BASE_REF
            Pattern: ^refs/heads/main$
            ExcludeMatchedPattern: false
      PullRequestBuildPolicy:
        RequiresCommentApproval: ALL_PULL_REQUESTS
        ApproverRoles:
          - GITHUB_MAINTAIN
          - GITHUB_ADMIN
```

Configuración de la consola

Para usar la consola AWS de administración para filtrar los eventos de webhooks:

1. En Aprobación de comentarios, seleccione deshabilitado o habilitado para todas las solicitudes de extracción (`ALL_PULL_REQUEST`) o solo para las solicitudes de extracción de bifurcaciones (`FORK_PULL_REQUEST`).
2. En Roles de aprobador, seleccione los roles de repositorio que tengan privilegios de aprobación para compilaciones de solicitudes de extracción cuando se requiera aprobación de comentarios.

Para obtener más información consulte Creación de un proyecto de compilación (consola) y WebhookFilter en la Referencia de la API de CodeBuild .



Vea los detalles de un proyecto de construcción en AWS CodeBuild

Puedes usar la AWS CodeBuild consola o AWS CLI AWS los SDK para ver los detalles de un proyecto de compilación. 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 (AWS SDK)

Ver los detalles de un proyecto de compilación (consola)

1. Abre la AWS CodeBuild consola en. https://console.aws.amazon.com/codesuite/codebuild/home
2. En el panel de navegación, elija Proyectos de compilación.

> **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 *project-name* página Crear proyecto:, seleccione Detalles de construcción.

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 más nombres de proyectos de construcción sobre los que 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 Cree 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 el uso del AWS CLI con AWS CodeBuild, consulte laReferencia de la línea de comandos.

Ver los detalles de un proyecto de compilación (AWS SDK)

Para obtener más información sobre su uso AWS CodeBuild con los AWS SDK, consulta la. AWS Referencia de SDK y herramientas

Ver los nombres de los proyectos de construcción en AWS CodeBuild

Puedes usar la AWS CodeBuild consola o los AWS CLI AWS SDK para ver una lista de proyectos de compilación en CodeBuild ellos.

Temas

- Ver una lista de nombres de proyectos de compilación (consola)

- Ver una lista de nombres de proyectos de compilación (AWS CLI)
- Vea una lista de nombres de proyectos de compilación (AWS SDK)

Ver una lista de nombres de proyectos de compilación (consola)

Puedes ver una lista de los proyectos de construcción de una AWS región 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. Abre la AWS CodeBuild consola en https://console.aws.amazon.com/codesuite/codebuild/home.
2. En el panel de navegación, elija Proyectos de compilación.

> **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 utilizará para enumerar los nombres de los proyectos de construcción. 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 el que se muestran los proyectos de construcción, en función 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"
```

```
  ]
}
```

Vea una lista de nombres de proyectos de compilación (AWS SDK)

Para obtener más información sobre su uso AWS CodeBuild con los AWS SDK, consulta la. AWS Referencia de SDK y herramientas

Incorpora AWS CodeBuild

Una compilación representa un conjunto de acciones realizadas AWS CodeBuild para crear artefactos de salida (por ejemplo, un archivo JAR) a partir de un conjunto 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 Ejecute AWS CodeBuild compilaciones manualmente.

  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

- Ejecute AWS CodeBuild compilaciones manualmente

- Ejecutar se basa en AWS Lambda cálculo
- Ejecución de compilaciones en flotas de capacidad reservada
- Ejecución de compilaciones en lotes
- Ejecución de pruebas paralelas en compilaciones por lotes
- Almacenamiento de las compilaciones en caché para mejorar el rendimiento
- Depurar compilaciones AWS CodeBuild
- Eliminar compilaciones AWS CodeBuild
- Vuelve a intentar compilar manualmente en AWS CodeBuild
- Reintentar se integra automáticamente AWS CodeBuild
- Detener las incorporaciones AWS CodeBuild
- Detenga las compilaciones por lotes AWS CodeBuild
- Trigger AWS CodeBuild se crea automáticamente
- Vea los detalles de la construcción en AWS CodeBuild
- Ver una lista de identificadores de compilación en AWS CodeBuild
- Ver una lista de los identificadores de compilación de un proyecto de compilación en AWS CodeBuild

Ejecute AWS CodeBuild compilaciones manualmente

Puedes usar la AWS CodeBuild consola o AWS CLI AWS los SDK para ejecutar una compilación. CodeBuild

Temas

- Ejecute compilaciones localmente con el AWS CodeBuild agente
- Ejecutar una compilación (consola)
- Ejecute una compilación (AWS CLI)
- Ejecute una compilación por lotes (AWS CLI)
- Iniciar la ejecución de compilaciones automáticamente (AWS CLI)
- Detener la ejecución de compilaciones automáticamente (AWS CLI)
- Ejecuta una compilación (AWS SDK)

Ejecute compilaciones localmente con el AWS CodeBuild agente

Puede usar el AWS CodeBuild agente para ejecutar CodeBuild compilaciones en una máquina 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 CodeBuild público de Amazon ECR https://gallery.ecr.aws/codebuild/amazonlinux-x86_64-standardcon el siguiente comando:

   ```
   $ docker pull public.ecr.aws/codebuild/amazonlinux-x86_64-standard:4.0
   ```

   Como alternativa, si desea utilizar otra imagen de Linux, ejecute los pasos siguientes:

   a. Clona el repositorio CodeBuild de imágenes:

      ```
      $ 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. Descarga el CodeBuild agente.

   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 CodeBuild agente 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:ccb19bdd7af94e4dc761e4c58c267e9455c28ec68d938086b4dc1cf8fe6b0940
   ```

   La firma SHA de la versión ARM del agente es:

   ```
   sha256:7d7b5d35d2ac4e062ae7ba8c662ffed15229a52d09bd0d664a7816c439679192
   ```

   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
   ```

Ejecute el CodeBuild agente

Para ejecutar el CodeBuild agente

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
   ```

   *<container-image>*Sustitúyalo por el nombre de la imagen del contenedor, como `aws/codebuild/standard:5.0` o`public.ecr.aws/codebuild/amazonlinux-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.

Reciba notificaciones sobre las nuevas versiones CodeBuild del agente

Puede suscribirse a las notificaciones de Amazon SNS para que se le notifique cuando se publiquen nuevas versiones del AWS CodeBuild agente.

Para suscribirse a las notificaciones de los CodeBuild agentes

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 AWS región a EE.UU. Este (Norte de Virginia). Debe seleccionar esta AWS región porque las notificaciones de Amazon SNS a las que se está suscribiendo se crean en esta 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. Seleccione Crear subscripción.

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 de los agentes CodeBuild

1. Abra la consola de Amazon SNS en https://console.aws.amazon.com/sns/v3/home.
2. En el panel de navegación, seleccione Suscripciones.
3. Elija la suscripción y, en Actions (Acciones), elija Delete subscriptions (Eliminar suscripciones). Cuando se le pida confirmación, elija Delete.

Ejecutar una compilación (consola)

AWS CodePipeline Para ejecutar una compilación con CodeBuild, omita estos pasos y siga las instrucciones que se indican enÚselo CodeBuild con CodePipeline.

1. Abre la AWS CodeBuild consola 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
- Origen
- 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
- Origen
- 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).

Ejecute una compilación (AWS CLI)

> **Note**
>
> Para usarlo CodePipeline para ejecutar una compilación con AWS CodeBuild, omita estos pasos y siga las instrucciones que aparecen enCree una canalización que utilice CodeBuild (AWS CLI).
> Para obtener más información sobre el uso de AWS CLI with CodeBuild, consulta laReferencia 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 start-build comando con la `--project-name` opción, *`<project-name>`* sustitúyala por el nombre del proyecto de compilación y, a continuación, salte al paso 6 de este procedimiento. Para obtener una lista de proyectos de compilación, consulte Visualización de los nombres de los 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 repites la `start-build` solicitud con el mismo token, pero cambias un parámetro, se mostrará un error de CodeBuild discordancia entre los parámetros.
4. Si ejecuta el start-build comando con la `--generate-cli-skeleton` opción, JSON-formatted los datos aparecen en la salida. Copie los datos a un archivo (por ejemplo*`start-build.json`*) en una ubicación del equipo local o de la instancia en la que AWS CLI esté instalado. 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 obligatoria. 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 *sourceVersion* se especifica, se utiliza la versión más reciente.
  - Para CodeCommit, el ID de confirmación que corresponde a la versión del código fuente que deseas compilar. Si no *sourceVersion* se especifica, se usa el ID de confirmación HEAD de la rama predeterminado. (No puedes especificar un nombre de etiqueta para*sourceVersion*, pero puedes especificar el ID de confirmación de la etiqueta).
  - Por GitHub ejemplo, el ID de confirmación, el ID de la solicitud de extracción, el nombre de la sucursal o el nombre de la etiqueta que corresponde a la versión del código fuente que quieres crear. 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 *sourceVersion* se especifica, se usa el ID de confirmación HEAD de la rama por defecto.

- 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 *sourceVersion* se especifica, se usa el ID de confirmación HEAD de la rama predeterminada.

- 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 repites la solicitud de inicio de compilación con el mismo token, pero cambias un parámetro, se mostrará un error de CodeBuild discordancia entre los parámetros.
- *`insecureSslOverride`*: Booleano opcional que especifica si se debe anular la configuración de TLS insegura especificada 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 anulación solo se aplica si la fuente de la compilación es Enterprise Server. GitHub

- *privilegedModeOverride*: Booleano opcional. Si se establece en true, la compilación sustituye el modo privilegiado en el proyecto de compilación.
- *queuedTimeoutInMinutesOverride*: entero opcional que especifica el número de minutos que una compilación puede permanecer en cola antes de que se agote el tiempo de espera. Su valor mínimo es de cinco minutos y su valor máximo es 480 minutos (ocho horas).
- *reportBuildStatusOverride*: Booleano opcional que especifica si se debe enviar al proveedor de origen el estado de inicio y finalización de una compilación. Si lo configuras con un proveedor de origen que no sea GitHub GitHub Enterprise Server o Bitbucket, se generará un resultado no válidoInputException .
- *sourceAuthOverride*: cadena opcional. Un tipo de autorización para esta compilación que sustituye la definida en el proyecto de compilación. Esta anulación solo se aplica si la fuente 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 identificador de clave de AWS acceso, una clave de acceso AWS secreta o una contraseña, como parámetro en el almacén de parámetros de Amazon EC2 Systems Manager. CodeBuild puede usar un parámetro almacenado en el almacén de parámetros de Amazon EC2 Systems Manager solo si el nombre de ese parámetro comienza por `/CodeBuild/` (por ejemplo,`/CodeBuild/dockerLoginPassword`). Puede usar la CodeBuild consola 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

utiliza la CodeBuild consola para crear un parámetro, la consola lo iniciará tal y `/CodeBuild/` como se está almacenando. 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 anteriormente elegiste Crear un nuevo rol de servicio en tu cuenta, incluirás esta acción automáticamente en el rol de servicio predeterminado de tu proyecto de compilación. CodeBuild 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 Cree un proyecto de compilación (AWS CLI). Para obtener una lista de las últimas opciones

de configuración de un proyecto de compilación, consulte Visualización de 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).

Ejecute 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 start-build-batch comando con la `--project-name` opción, *<project-name>* sustitúyalo por el nombre del proyecto de compilación y, a continuación, vaya al paso 6 de este procedimiento. Para obtener una lista de proyectos de compilación, consulte Visualización de los nombres de los 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 repites la `start-build-batch` solicitud con el mismo token, pero cambias un parámetro, se mostrará un error de CodeBuild discordancia entre los parámetros.

4. Si ejecuta el start-build-batch comando con la `--generate-cli-skeleton` opción, JSON-formatted los datos se envían al *<json-file>* archivo. Este archivo es similar al esqueleto creado por el start-build comando, con la adición del siguiente objeto. Para obtener más información acerca de los objetos comunes, consulte Ejecute 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 `buildBatchConfigOverride` objeto es una ProjectBuildBatchConfig estructura que contiene las anulaciones de 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 realiza correctamente, la representación JSON de un BuildBatch objeto aparece en la salida de la consola. Consulte la sintaxis de StartBuildBatch respuesta para ver un ejemplo de estos datos.

Iniciar la ejecución de compilaciones automáticamente (AWS CLI)

Si el código fuente está almacenado en un repositorio de GitHub Enterprise Server GitHub o en uno de Enterprise Server, puedes usar GitHub webhooks para AWS CodeBuild reconstruir el código fuente cada vez que se introduzca un cambio de código en el repositorio.

Ejecute el comando create-webhook de la siguiente manera:

```
aws codebuild create-webhook --project-name <project-name>
```

*<project-name>*es el nombre del proyecto de compilación que contiene el código fuente que se va a reconstruir.

Pues GitHub, en el resultado aparece información similar a la siguiente:

```
{
  "webhook": {
    "url": "<url>"
  }
}
```

*<url>*es la URL del GitHub webhook.

En el GitHub caso de Enterprise Server, en el resultado aparece información similar a la siguiente:


```
{
    "webhook": {
        "secret": "YRV4JYAGFsekJiirp5ytx86oZpyhUdySNSDTLNuXoXX1c7aZ6XYDf37-ZFyY02rs4JSE7OmLW3w-gh-ryoVB8OSSSC1aAtBtuPkHw
YuncCCmdogCVCfniQ7ukYX2_xM--nlDma5EngIg_Bi_N465yi33zyTUNPoQlxCpLO-BwghcVa9lAurwR77-uY7i-_XCJFahwMx1f4ubOgBBSmMT2A16apqjq
QJoKSb61XVKyZy1Giuy4nliAXfv9WNn76CaCsndb3fVIE78fpygfo41xYxSQ6vpo6LRTKtPzbyeTHbVXGda1PJvnkBlnKmJDo0RTgI1m2oYr17dWziQlrrvo
CoNgy1SOO_7LKfA-nNXFc_f1SiFy0AqeMB43-dOOcdkzybHncE81QTRwEUCFfmX-AJCwmLXVOkg0G67T92Sjbpz0fRlkh5pwIFl93_bB_jOHDinK6iOiPpf2
dIDAIZgGMagqZeWb-axDeTAbopoU8J6gFI1yKo5aq9ql51zC1PERUsMgJFtJr_a-Z-L_kylr-4hSSxasSJNuJ43_XOBRWqT51xqvH-A69bVO7KbVT_Kc6wxk
SHyYCEMoa_Pfa7ZQgyfY6B00ogMNj31yFbjthORNL1cDo6-3J-McDLoyrRtSEOV9QnxvsG5zu1N5-z20rkJtg_MOfNwocfUutFXb7vrGTduHlR1dzXLRusHu
xOVVuDUWm9vhWMr-hUkeGo_1kDKyk4E2QFvZXpjYw0vFfV-dwxFRR_mifzxW1wyfmt2iFtLkp_YZj_4WeFAckGefr-ilNaYvsZpzXj78Ae1adVoLf48AmDdN
2pWsWJjatU9zt942gLisFFmKakcvJuy5yxXHaxxbhUyC8NHYiESUWPfcfnqrMsr8op3P4AUCHlpiZCYYuiwI_cac-pIUBO0Xaur_lu_fyFghgOJc7cfTnA36
rv5X5DnFDM8P3HNBeLjaF9QZ6AijegPEwTHIkJON3AUDwpkz_hwTXyUoAU8MdZfPTXbBoT6N5Z5THBHsYxR",
        "payloadUrl": "https://codebuild.us-east-2.amazonaws.com/webhooks?t=eyJlbmNyeXB0ZWREYXRhIjoiUmFqMmJERGRQbGhwLzNT
Nld3R0VGRjZZOTNwLzlZVG1NZlpIR1E0RUsxdzhGeWhnVFFqWTR0WEFwT2dJRnNmRHc3S3RNc0xYMENncXFTakg1cE1nSy9zPSIsIml2UGFyYW1ldGVyU3Bl
YyI6IndSQ1Qrc2VPQjBCZzhPeVYiLCJtYXRlcmlhbFNldFNlcmlhbCI6MX0%3D&v=1"
    }
}
```


1. Copie la URL de carga y la clave secreta de la salida. Los necesita para añadir un webhook en GitHub Enterprise Server.
2. En GitHub Enterprise Server, elige el repositorio en el que está almacenado tu CodeBuild proyecto. Elija Settings (Configuración), elija Hooks & services (Enlaces y servicios) y, a continuación, elija Add webhook (Añadir webhook).
3. Introduzca la URL de carga y la clave secreta, acepte los valores predeterminados para el resto de los campos y elija Add webhook.

Detener la ejecución de compilaciones automáticamente (AWS CLI)

Si tu código fuente está almacenado en un repositorio de GitHub Enterprise Server GitHub o en uno de Enterprise Server, puedes configurar GitHub webhooks para que AWS CodeBuild reconstruyan tu código fuente cada vez que se introduzca 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 `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 reconstruir.

Si este comando se ejecuta correctamente, no aparecerá información ni errores en el resultado.

> **Note**
>
> Esto elimina el webhook únicamente de tu CodeBuild proyecto. También debes eliminar el webhook de tu repositorio GitHub o de GitHub Enterprise Server.

Ejecuta una compilación (AWS SDK)

CodePipeline Para ejecutar una compilación con AWS CodeBuild, omita estos pasos y, en Uso AWS CodeBuild por AWS CodePipeline para probar el código y ejecutar compilaciones su lugar, siga las instrucciones que se indican.

Para obtener información sobre su uso CodeBuild con los AWS SDK, consulta laAWS Referencia de SDK y herramientas.

Ejecutar se basa en AWS Lambda cálculo

AWS Lambda la computación ofrece velocidades de inicio optimizadas para sus compilaciones. AWS Lambda admite compilaciones más rápidas debido a una menor latencia de inicio. AWS Lambda también se escala automáticamente, por lo que las compilaciones no esperan en la cola para ejecutarse. Sin embargo, hay algunos casos de uso que AWS Lambda no son compatibles y, si te afectan, utiliza el procesamiento de EC2. Para obtener más información, consulte Limitaciones de AWS Lambda cálculo.

Temas

- Qué herramientas y tiempos de ejecución se incluirán en las imágenes seleccionadas de Docker del entorno de ejecución en las que se ejecutan AWS Lambda?
- ¿Qué pasa si la imagen seleccionada no incluye las herramientas que necesito?
- Qué regiones son compatibles AWS Lambda computar en CodeBuild?
- Limitaciones de AWS Lambda cálculo
- Implemente una función de Lambda mediante AWS SAM con CodeBuild Lambda Java
- Cree una aplicación React de una sola página con CodeBuild Lambda Node.js
- Actualizar la configuración de una función de Lambda con CodeBuild Lambda Python

Qué herramientas y tiempos de ejecución se incluirán en las imágenes seleccionadas de Docker del entorno de ejecución en las que se ejecutan AWS Lambda?

AWS Lambda admite las siguientes herramientas: AWS CLI v2, AWS SAM CLI, git, go, Java, Python Node.js, 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 necesita, puede proporcionar una imagen de Docker de entorno personalizado que sí las incluya.

> **Note**
>
> Lambda no admite funciones que utilizan imágenes de contenedor de varias arquitecturas. Para obtener más información, consulte Creación de una función de Lambda con una imagen de contenedor en la Guía para desarrolladores de AWS Lambda .

Tenga en cuenta que necesita los siguientes permisos de Amazon ECR para utilizar imágenes personalizadas para la computación de Lambda:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ecr:GetAuthorizationToken"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
```

```
                "ecr:BatchCheckLayerAvailability",
                "ecr:GetDownloadUrlForLayer",
                "ecr:BatchGetImage"
            ],
            "Resource": "arn:aws:ecr:us-east-1:111122223333:repository/image-
repo"
        }
    ]
}
```

Tenga en cuenta también que debe haber instalado `curl` o `wget` para poder utilizar imágenes personalizadas.

Qué regiones son compatibles AWS Lambda computar en CodeBuild?

En CodeBuild, la AWS Lambda computación es compatible en los siguientes países 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, consulte AWS Servicios por región.

Limitaciones de AWS Lambda cálculo

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.
- 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 el tipo de entorno `LINUX_GPU_CONTAINER` y no es compatible con Windows Server Core 2019.
- AWS Lambda no admite el almacenamiento en caché, los tiempos de espera de compilación personalizados, el tiempo de espera de cola, 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, la instalación de certificados ni el acceso SSH con Session Manager.

Implemente una función de Lambda mediante AWS SAM con CodeBuild Lambda Java

The AWS Serverless Application Model (AWS SAM) es un marco de código abierto para crear aplicaciones sin servidor. Para obtener más información, consulte el AWS Serverless Application Model repositorio en. GitHub En el siguiente ejemplo de Java, se usa Gradle para crear y probar una AWS Lambda función. Después, la AWS SAM CLI se usa para implementar la 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 AWS SAM repositorio

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 para AWS Serverless Application Model desarrolladores para instalar la AWS SAM CLI en su máquina local.
2. Ejecute `sam init` y seleccione la siguiente configuración del 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 a un repositorio de código fuente compatible. Para obtener una lista de los tipos de fuentes compatibles, consulte ProjectSource.

Cree un proyecto de CodeBuild Lambda en Java

Cree un proyecto Java de AWS CodeBuild Lambda y configure los permisos de IAM necesarios para la compilación.

Para crear su proyecto CodeBuild Lambda Java

1. Abra la AWS CodeBuild consola en. https://console.aws.amazon.com/codesuite/codebuild/home
2. Si aparece una página de CodeBuild información, seleccione 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.
3. En Project name (Nombre de proyecto), escriba un nombre para este proyecto de compilación. Los nombres de los proyectos de construcción 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, selecciona el repositorio de origen en el que se encuentra tu AWS SAM proyecto.
5. En Environment (Entorno):
   - En Computación, seleccione Lambda.
   - En Tiempo(s) de ejecución, seleccione Java.
   - En Imagen, selecciona aws/codebuild /amazonlinux-x86_64-lambda-standard:corretto21.
   - En Rol de servicio, seleccione Nuevo rol de servicio. Anote el Nombre del rol. Lo necesitará 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 Roles y seleccione el rol de servicio asociado a su proyecto. Para encontrar tu rol en CodeBuild el proyecto, selecciona el proyecto de compilación, elige Editar, Entorno y, a continuación, el rol de servicio.
9. Elija la pestaña Relaciones de confianza y, a continuación, Editar política de confianza.
10. Agregue la siguiente política insertada al rol de IAM. Esto se usará para implementar tu AWS SAM infraestructura más adelante. Para obtener más información, consulta Adición y eliminación de permisos de identidad de IAM en la Guía del usuario de IAM.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Action": [
                "cloudformation:*",
                "lambda:*",
                "iam:*",
                "apigateway:*",
                "s3:*"
            ],
            "Resource": "arn:aws:iam::*:role/Service*"
        }
    ]
}
```

Configuración de la especificación de compilación del proyecto

Para crear, probar e implementar la función de Lambda, CodeBuild lee y ejecuta los comandos de compilación desde una especificación de compilación.

Para configurar la especificación de compilación del proyecto

1. En la CodeBuild consola, selecciona tu proyecto de compilación y, a continuación, elige Editar y Buildspec.
2. En Especificación de compilación, elija Insertar comandos de compilación y, a continuación, Cambiar al editor.
3. Elimine los comandos de compilación rellenados previamente 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).

Implementa tu AWS SAM Infraestructura Lambda

Use CodeBuild Lambda para implementar automáticamente su infraestructura Lambda

Para implementar su infraestructura de Lambda

1. Seleccione Iniciar la compilación. Esto compilará, probará e implementará automáticamente su AWS SAM aplicación para AWS Lambda usarla. CloudFormation
2. Una vez finalizada la compilación, navegue hasta la AWS Lambda consola y busque su nueva función de Lambda con el nombre del AWS SAM proyecto.
3. Pruebe la función de Lambda; para ello, seleccione API Gateway en la información general de Función y, a continuación, haga clic en la URL de Punto de conexión de API. Debería aparecer una página abierta con el mensaje `"message": "hello world"`.

Limpieza de la 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 la infraestructura

1. Navegue hasta la CloudFormation consola y seleccione el`aws-sam-cli-managed-default`.
2. En Recursos, vacíe el bucket de implementación `SamCliSourceBucket`.
3. Elimine la pila de `aws-sam-cli-managed-default`.
4. Elimine la CloudFormation pila asociada a su AWS SAM proyecto. Esta pila debe tener el mismo nombre que tu AWS SAM proyecto.

5. Navega hasta la CloudWatch consola y elimina los grupos de CloudWatch registros asociados a tu CodeBuild proyecto.
6. Navega hasta 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 Node.js ejemplo se utiliza Node.js para crear los artefactos de origen a partir de la aplicación Create React y devuelve los artefactos de compilación.

Configuración del repositorio de origen y el bucket de artefactos

Cree un repositorio de origen para su proyecto con yarn y Create React App.

Para configurar el repositorio de origen y el bucket de artefactos

1. Ejecute `yarn create react-app <app-name>` en su máquina local para crear una aplicación React sencilla.
2. Cargue la carpeta del proyecto de la aplicación React en un repositorio de origen admitido. Para obtener una lista de los tipos de fuentes compatibles, consulte ProjectSource.

Crear un proyecto CodeBuild Lambda Node.js

Cree un proyecto AWS CodeBuild Lambda Node.js .

Para crear su proyecto CodeBuild Lambda Node.js

1. Abra la AWS CodeBuild consola en. https://console.aws.amazon.com/codesuite/codebuild/home
2. Si se muestra 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 Fuente, selecciona el repositorio de origen en el que se encuentra tu AWS SAM proyecto.
5. En Environment (Entorno):
   - En Computación, seleccione 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 Nombre del bucket, seleccione el bucket de artefactos del proyecto que ha creado anteriormente.
   - En Empaquetado de artefactos, seleccione Zip.
7. Elija Crear el proyecto de compilación.

Configuración de la especificación de compilación del proyecto

Para crear tu aplicación React, lee y ejecuta los comandos de compilación desde un archivo buildspec. CodeBuild

Para configurar la especificación de compilación del proyecto

1. En la CodeBuild consola, selecciona tu proyecto de compilación y, a continuación, selecciona Editar y Buildspec.
2. En Especificación de compilación, elija Insertar comandos de compilación y, a continuación, Cambiar al editor.
3. Elimine los comandos de compilación rellenados previamente 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).

Compilación y ejecución de la aplicación React

Cree la aplicación React en CodeBuild Lambda, descargue los artefactos de compilación y ejecute la aplicación React localmente.

Para compilar y ejecutar su aplicación React

1. Seleccione Iniciar la compilación.
2. Una vez finalizada la compilación, navegue hasta el bucket de artefactos del proyecto de Amazon S3 y descargue el artefacto de la aplicación React.
3. Descomprima el artefacto de compilación de React y ejecute `run npm install -g serve && serve -s build` en la carpeta del proyecto.
4. El comando `serve` ofrecerá el sitio estático en un puerto local e imprimirá la salida en su terminal. Puede visitar la URL de localhost en `Local:` en la salida del terminal para ver su aplicación React.

Para obtener más información sobre cómo controlar la implementación de un servidor basado en React, consulte Create React App Deployment.

Limpieza de la infraestructura

Para evitar cargos adicionales por los recursos que utilizó durante este tutorial, elimine los recursos creados para su CodeBuild proyecto.

Para limpiar la infraestructura

1. Eliminación del bucket de Amazon S3 de artefactos del proyecto
2. Navegue 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 de Lambda con CodeBuild Lambda Python

El siguiente ejemplo de Python usa Boto3 y CodeBuild Lambda Python para actualizar la configuración de una función de Lambda. Este ejemplo se puede ampliar para administrar otros recursos mediante programación. AWS Para obtener más información, consulte la documentación de Boto3.

Requisitos previos

Cree o busque una función de Lambda en su cuenta.

En este ejemplo se presupone que ya ha creado una función de Lambda en su cuenta y que la utilizará CodeBuild para actualizar las variables de entorno de la función de Lambda. Para obtener más información sobre cómo configurar una función de Lambda CodeBuild, consulta el Implemente una función de Lambda mediante AWS SAM con CodeBuild Lambda Java ejemplo o visita. AWS Lambda

Configuración del repositorio de origen

Cree un repositorio de origen para almacenar su script de Boto3 para Python.

Para configurar el repositorio de origen

1. Copie el siguiente script de Python en un archivo nuevo denominado `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 de python a un repositorio de origen compatible. Para obtener una lista de los tipos de fuentes compatibles, consulte ProjectSource.

Cree un proyecto de CodeBuild Lambda Python

Cree un proyecto de CodeBuild Lambda Python.

Para crear su proyecto de CodeBuild Lambda en Java

1. Abra la AWS CodeBuild consola en. https://console.aws.amazon.com/codesuite/codebuild/home
2. Si aparece una página de CodeBuild información, seleccione 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.
3. En Project name (Nombre de proyecto), escriba un nombre para este proyecto de compilación. Los nombres de los proyectos de construcción 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, selecciona el repositorio de origen en el que se encuentra tu AWS SAM proyecto.

5. En Environment (Entorno):
   - En Computación, seleccione Lambda.
   - En Tiempo(s) de ejecución, elija Python.
   - Para Imagen, selecciona aws/codebuild /amazonlinux-x86_64-lambda-standard:python3.12.
   - En Rol de servicio, seleccione Nuevo rol de servicio. Anote el Nombre del rol. Lo necesitará 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 Roles y seleccione el rol de servicio asociado a su proyecto. Para encontrar tu rol en CodeBuild el proyecto, selecciona el proyecto de compilación, elige Editar, Entorno y, a continuación, el rol de servicio.
9. Elija la pestaña Relaciones de confianza y, a continuación, Editar política de confianza.
10. Agregue la siguiente política insertada al rol de IAM. Esto se usará para implementar tu AWS SAM infraestructura más adelante. Para obtener más información, consulta Adición y eliminación de permisos de identidad de IAM en la Guía del usuario de IAM.

    JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "UpdateLambdaPermissions",
            "Effect": "Allow",
            "Action": [
                "lambda:UpdateFunctionConfiguration"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

Configuración de la especificación de compilación del proyecto

Para actualizar la función Lambda, el script lee las variables de entorno de la especificación de compilación para encontrar el nombre de la función Lambda, el nombre de la variable de entorno y el valor de la variable de entorno.

Para configurar la especificación de compilación del proyecto

1. En la CodeBuild consola, selecciona tu proyecto de compilación y, a continuación, elige Editar y Buildspec.
2. En Especificación de compilación, elija Insertar comandos de compilación y, a continuación, Cambiar al editor.
3. Elimine los comandos de compilación rellenados previamente 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).

Actualización de la configuración de Lambda

Usa CodeBuild Lambda Python para actualizar automáticamente la configuración de tu función de Lambda.

Para actualizar la configuración de su función Lambda

1. Seleccione Iniciar la compilación.
2. Una vez finalizada la compilación, navegue hasta su función de Lambda.

3. Seleccione Configuración y, a continuación, elija las variables de Entorno. Debería aparecer una nueva variable de entorno con la clave `FEATURE_ENABLED` y el valor `true`.

Limpieza de la infraestructura

Para evitar cargos adicionales por los recursos que utilizó durante este tutorial, elimine los recursos creados para su CodeBuild proyecto.

Para limpiar la infraestructura

1. Navega hasta la CloudWatch consola y elimina los grupos de CloudWatch registros asociados a tu CodeBuild proyecto.
2. Navega hasta la CodeBuild consola y elimina tu CodeBuild proyecto seleccionando Eliminar proyecto de compilación.
3. Si ha creado una función de Lambda para usarla en este ejemplo, elija Acciones y Eliminar la función para limpiar la función de Lambda.

Extensiones

Si quieres ampliar este ejemplo para administrar otros AWS recursos con AWS CodeBuild Lambda Python:

- Actualice el script de Python para modificar los nuevos recursos con Boto3.
- Actualice la función de IAM asociada a su CodeBuild proyecto para tener permisos para los nuevos recursos.
- Agregue cualquier variable de entorno nueva asociada a los nuevos recursos a su especificación de compilación.

Ejecución de compilaciones en flotas de capacidad reservada

CodeBuild ofrece las siguientes flotas informáticas:

- On-demand flotas
- Flotas de capacidad reservada

Con flotas bajo demanda, CodeBuild proporciona computación para sus compilaciones. Las máquinas se destruyen cuando finaliza la construcción. On-demand las flotas están totalmente administradas e incluyen capacidades de escalado automático para hacer frente a los picos de demanda.

> **Note**
>
> On-demand las flotas no son compatibles con macOS.

CodeBuild también ofrece flotas de capacidad reservada que contienen instancias impulsadas por Amazon EC2 y 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 reducir la duración de las compilaciones. Con las flotas de capacidad reservada, sus máquinas siempre están en funcionamiento y seguirán generando costos mientras estén aprovisionadas.

> **Important**
>
> Independientemente del tiempo que se ejecute una instancia, las flotas de capacidad reservada incurren en un cargo inicial por instancia, tras el cual puede haber costos adicionales asociados. Para obtener más información, consulte https://aws.amazon.com/codebuild/pricing/.

Temas

- Creación de una flota de capacidad reservada
- Prácticas recomendadas
- ¿Puedo compartir una flota de capacidad reservada en varios CodeBuild proyectos?
- ¿Cómo funciona el trabajo de computación basado en atributos?
- ¿Puedo especificar manualmente una instancia de Amazon EC2 para mi flota?
- ¿Qué regiones admiten flotas con capacidad reservada?
- ¿Cómo se puede configurar una flota de macOS con capacidad reservada?
- ¿Cómo puedo configurar una imagen de máquina de Amazon (AMI) personalizada para una flota de capacidad reservada?
- Limitaciones de las flotas de capacidad reservada

- Propiedades de flota con capacidad reservada
- Muestras de capacidad reservada con AWS CodeBuild

Creación de una flota de capacidad reservada

Siga las instrucciones a continuación para crear una flota de capacidad reservada.

Para crear una flota de capacidad reservada

1. Inicie sesión en Consola de administración de AWS y abra la consola en AWS CodeBuild . https://console.aws.amazon.com/codesuite/codebuild/home
2. En el panel de navegación, elija Flotas de computación y, a continuación, Crear flota.
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. (Opcional) Seleccione Usar modo de ejecución de instancias: opcional para ejecutar directamente en una instancia de Amazon EC2 en lugar de en un contenedor de Docker. A continuación, elija una Versión principal y una Versión secundaria.
7. (Opcional) En Configuración adicional, haga lo siguiente:

   - Seleccione Configurar VPC: opcional para conectar la flota a una VPC y acceder a los recursos privados durante el uso.
     - En el menú desplegable de la VPC, selecciona una VPC a la que accederá tu CodeBuild flota.
     - En el menú desplegable Subredes, selecciona las subredes que CodeBuild debes usar para configurar tu VPC.
     - En el menú desplegable Grupos de seguridad, selecciona los grupos de seguridad que CodeBuild debes usar para trabajar con tu VPC.
     - En el campo Rol de servicio de la flota, elija un rol de servicio existente.

     > **Note**
     >
     > Asegúrese de que el rol de la flota tiene los permisos necesarios. Para obtener más información, consulte Cómo permitir a un usuario agregar una política de permisos para un rol de servicio de flota.

- Si ha elegido el sistema operativo Amazon Linux, seleccione Definir configuraciones de proxy (opcional) para aplicar control de acceso a la red a las instancias de capacidad reservada.
- En Comportamiento predeterminado, elija permitir o denegar el tráfico saliente a todos los destinos de forma predeterminada.
- En Reglas de proxy, elija Agregar regla de proxy para especificar las direcciones IP o los dominios de destino para los que se va a permitir o denegar el control de acceso a la red.

- Seleccione Configurar AMI personalizada: opcional para usar una imagen de máquina de Amazon (AMI) personalizada.
  - En el menú desplegable de la AMI, seleccione una imagen de máquina de Amazon (AMI) para su flota.
  - En el campo Rol de servicio de la flota, elija un rol de servicio existente.

> **Note**
>
> Asegúrese de que el rol de la flota tiene los permisos necesarios. Para obtener más información, consulte Cómo permitir a un usuario agregar una política de permisos para un rol de servicio de flota.

8. En Configuración de capacidad, en Modo de selección de computación, elija una de las siguientes opciones:

- Si elige Selección guiada, haga lo siguiente:
  - En Computación, elija el tipo de instancias incluidas en esta flota.
  - En el campo de texto Capacidad, introduzca el número mínimo de instancias de la flota.
  - (Opcional) En Configuración adicional, haga lo siguiente:
    - Seleccione Configurar escalado: opcional para escalar automáticamente la flota en función de esta configuración. En el menú desplegable Modo de escalado: opcional, elija el comportamiento cuando la demanda supere la capacidad de la flota.
- Si elige Instancia personalizada, haga lo siguiente:
  - En el menú desplegable Tipo de instancia de computación, seleccione el tipo de instancias incluidas en esta flota.
  - En el campo de texto Tamaño de volumen de EBS adicional: opcional, introduzca el volumen adicional a los 64 GB de espacio en disco proporcionados.

- En el campo de texto Capacidad, introduzca el número mínimo de instancias de la flota.
- (Opcional) En Configuración adicional, haga lo siguiente:
  - Seleccione Configurar escalado: opcional para escalar automáticamente la flota en función de esta configuración. En el menú desplegable Modo de escalado: opcional, elija el comportamiento cuando la demanda supere la capacidad de la flota.

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.

> **Important**
>
> Cuando se usa la característica de capacidad reservada, otros proyectos de la misma cuenta pueden acceder a los datos almacenados en caché en las instancias de flota (incluidos los archivos de origen, las capas de Docker y los directorios almacenados en caché que se indican en la especificación de compilación). Esto es así por diseño y permite que los proyectos de la misma cuenta compartan instancias de flotas.

¿Cómo funciona el trabajo de computación basado en atributos?

Si elige `ATTRIBUTE_BASED_COMPUTE` como `computeType` de la flota, puede especificar los atributos en un nuevo campo llamado `computeConfiguration`. Estos atributos incluyen vCPU, memoria, espacio en disco y `machineType`. Este `machineType` es `GENERAL` o `NVME`. Tras especificar uno o algunos de los atributos disponibles, CodeBuild elegiré un tipo de procesamiento de entre los tipos de instancias compatibles disponibles al finalizar`computeConfiguration`.

> **Note**
>
> CodeBuild elegirá la instancia más barata que cumpla con todos los requisitos de entrada. La memoria, las vCPU y el espacio en disco de las instancias elegidas serán superiores o iguales a los requisitos de entrada. Puede comprobar la `computeConfiguration` resuelta en la flota creada o actualizada.

Si ingresas un valor `computeConfiguration` que no puedas cumplir CodeBuild, recibirás una excepción de validación. Tenga en cuenta también que el comportamiento de desbordamiento de la flota bajo demanda se sustituirá por el comportamiento de la cola si la `computeConfiguration` no está disponible bajo demanda.

¿Puedo especificar manualmente una instancia de Amazon EC2 para mi flota?

Sí, puede introducir directamente la instancia de Amazon EC2 que desee en la consola seleccionando Instancia personalizada o configurando el parámetro de API, `InstanceType`. Este campo se usa en las siguientes API: CreateFleet UpdateFleet, CreateProject, UpdateProject y StartBuild. Para obtener más información, consulte Compute instance type.

¿Qué regiones admiten flotas con capacidad reservada?

Capacidad reservada Las flotas de Amazon Linux y Windows son compatibles con los siguientes países 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, consulte AWS Servicios por región.

Las flotas de macOS Medium con capacidad reservada son compatibles con los siguientes países Regiones de AWS: EE.UU. Este (Norte de Virginia), EE.UU. Este (Ohio), EE.UU. Oeste (Oregón), Asia Pacífico (Sídney) y Europa (Fráncfort). Capacidad reservada Las flotas de macOS Large son compatibles con los siguientes países Regiones de AWS: EE.UU. Este (Norte de Virginia), EE.UU. Este (Ohio), EE.UU. Oeste (Oregón) y Asia Pacífico (Sídney).

¿Cómo se puede configurar una flota de macOS con capacidad reservada?

Para configurar una flota de macOS con capacidad reservada

1. Inicia sesión en Consola de administración de AWS y abre la AWS CodeBuild consola en https://console.aws.amazon.com/codesuite/codebuild/home.
2. En el panel de navegación, elija Flotas de computación y, a continuación, Crear flota.
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, seleccione macOS.
5. En el campo Computación, seleccione uno de los siguientes tipos de máquinas de computación: Apple M2, 24 GB de memoria, 8 vCPU o Apple M2, 32 GB de memoria, 12 vCPU.
6. En el campo de texto Capacidad, introduzca el número mínimo de instancias de la flota.
7. (Opcional) Para usar una imagen personalizada para la flota, consulte ¿Cómo puedo configurar una imagen de máquina de Amazon (AMI) personalizada para una flota de capacidad reservada? para garantizar que la Imagen de máquina de Amazon (AMI) cumpla los requisitos previos.
8. (Opcional) Para configurar una VPC con su flota, en Configuración adicional, haga lo siguiente:
   - En el menú desplegable opcional de VPC, selecciona una VPC a la que accederá tu CodeBuild flota.
   - En el menú desplegable Subredes, selecciona las subredes que CodeBuild debes usar para configurar tu VPC.
   - En el menú desplegable Grupos de seguridad, selecciona los grupos de seguridad que CodeBuild debes usar para trabajar con tu VPC.
   - En el campo Rol de servicio de la flota, elija un rol de servicio existente.

> **Note**
>
> Asegúrese de que el rol de la flota tiene los permisos necesarios. Para obtener más información, consulte Cómo permitir a un usuario agregar una política de permisos para un rol de servicio de flota.

9. Elija Crear flota de computación y espere a que se lance la instancia de flota. Una vez lanzada*n*/*n*, la capacidad estará donde *n* esté disponible.
10. Una vez que se haya lanzado la flota informática, 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.

¿Cómo puedo configurar una imagen de máquina de Amazon (AMI) personalizada para una flota de capacidad reservada?

Cómo configurar una imagen de máquina de Amazon (AMI) personalizada para una flota de capacidad reservada

1. Inicia sesión en Consola de administración de AWS y abre la AWS CodeBuild consola en https://console.aws.amazon.com/codesuite/codebuild/home.
2. En el panel de navegación, elija Flotas de computación y, a continuación, Crear flota.
3. En el campo de texto Nombre de la flota de computación, introduzca el nombre de su flota.
4. Elija Imagen personalizada para su flota y asegúrese de que su imagen de máquina de Amazon (AMI) cumpla los siguientes requisitos previos:

   - Si el tipo de entorno es `MAC_ARM`, asegúrese de que la Arquitectura de la AMI sea `Mac-Arm` de 64 bits.
   - Si el tipo de entorno es `LINUX_EC2`, asegúrese de que la Arquitectura de la AMI sea `x86` de 64 bits.
   - Si el tipo de entorno es `ARM_EC2`, asegúrese de que la Arquitectura de la AMI sea `Arm` de 64 bits.
   - Si el tipo de entorno es `WINDOWS_EC2`, asegúrese de que la Arquitectura de la AMI sea `x86` de 64 bits.

- La AMI permite el ARN de la organización del CodeBuild servicio. Para obtener una lista de los ARN de organización, consulte Amazon Machine Images (AMI).
- Si la AMI está cifrada con una AWS KMS clave, la AWS KMS clave también debe incluir el ID de la organización del CodeBuild servicio. Para obtener una lista de los ID de organización, consulte Amazon Machine Images (AMI). Para obtener más información sobre AWS KMS las claves, consulte Permitir que las organizaciones y las unidades organizativas usen una clave de KMS en la guía del usuario de Amazon EC2. Para conceder a CodeBuild la organización permiso para usar una clave de KMS, añada la siguiente declaración a la política de claves:

```
{
    "Sid": "Allow access for organization root",
    "Effect": "Allow",
    "Principal": "*",
    "Action": [
        "kms:Describe*",
        "kms:List*",
        "kms:Get*",
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:CreateGrant"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "aws:PrincipalOrgID": "o-123example"
        }
    }
}
```

- En el campo Rol de servicio de la flota, conceda los siguientes permisos de Amazon EC2:

  JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
               "ec2:DescribeImages",
```

```
                "ec2:DescribeSnapshots"
            ],
            "Resource": "*"
        }
    ]
}
```

Limitaciones de las flotas de capacidad reservada

Hay algunos casos de uso que las flotas de capacidad reservada no admiten. En casi de que le afectan, utilice flotas bajo demanda:

- Las flotas de capacidad reservada no admiten métricas de utilización de compilaciones.
- Las flotas de macOS con capacidad reservada no admiten la sesión de depuración.

Para obtener más información sobre las límites y las cuotas, consulte Flotas de computación.

Propiedades de flota con capacidad reservada

Una flota de capacidad reservada contiene las siguientes propiedades. Para obtener más información sobre las flotas de capacidad reservada, consulte Ejecución de compilaciones en flotas de capacidad reservada.

Sistema operativo

El sistema operativo. A continuación, se muestran los sistemas operativos disponibles:

- Amazon Linux
- macOS
- Windows Server 2019
- Windows Server 2022

Arquitectura

Arquitectura del procesador. Están disponibles las siguientes arquitecturas:

- x86_64
- Arm64

Tipo de entorno

Los tipos de entorno disponibles cuando se seleccione Amazon Linux. A continuación, se muestran los tipos de entorno disponibles:

- Linux EC2
- Linux GPU

Tipo de instancia de computación

Las configuraciones de computación de las instancias de flota.

Selección guiada

Especifique diferentes tipos de computación seleccionando la configuración de vCPU, memoria y espacio en disco. Para obtener más información acerca de la disponibilidad de tipo de computación por región, consulte Acerca de los tipos de entorno de capacidad reservada.

Instancia personalizada

Especifique manualmente el tipo de instancia deseado.

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.

On-demand

Las compilaciones de Overflow se ejecutan bajo CodeBuild demanda.

> **Note**
>
> Si eliges configurar tu comportamiento de desbordamiento según demanda al crear una VPC-connected flota, asegúrate de añadir los permisos de VPC necesarios a tu función de servicio de proyecto. Para obtener más información, consulta un ejemplo de declaración de política para permitir el CodeBuild acceso a AWS los servicios necesarios para crear una interfaz de red de VPC.

> **Important**
>
> Si decide configurar su comportamiento de desbordamiento como bajo demanda, tenga en cuenta que las compilaciones de desbordamiento se facturarán por separado, de manera 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.

Imágenes de máquina de Amazon (AMI)

Las propiedades de Imagen de máquina de Amazon (AMI) de su flota. Las siguientes propiedades son compatibles CodeBuild con:

Regiones de AWS	ARN de la organización	ID de organización
us-east-1	arn:aws:organizations::851725618577:organization/o-c6wcu152r1	o-c6wcu152r1
us-east-2	arn:aws:organizations::992382780434:organization/o-seufr2suvq	o-seufr2suvq
us-west-2	arn:aws:organizations::381491982620:organization/o-0412o99a4r	o-0412o99a4r
ap-northeast-1	arn:aws:organizations::891376993293:organization/o-b6k3sjqavm	o-b6k3sjqavm

Regiones de AWS	ARN de la organización	ID de organización
ap-south-1	arn:aws:organizations::891376924779:organization/o-krtah1lkeg	o-krtah1lkeg
ap-southeast-1	arn:aws:organizations::654654522137:organization/o-mcn8uvc3tp	o-mcn8uvc3tp
ap-southeast-2	arn:aws:organizations::767398067170:organization/o-6crt0f6bu4	o-6crt0f6bu4
eu-central-1	arn:aws:organizations::590183817084:organization/o-lb2lne3te6	o-lb2lne3te6
eu-west-1	arn:aws:organizations::891376938588:organization/o-ullrrg5qf0	o-ullrrg5qf0
sa-east-1	arn:aws:organizations::533267309133:organization/o-db63c45ozw	o-db63c45ozw

Configuración adicional

VPC (opcional)

La VPC a la que accederá su CodeBuild flota. Para obtener más información, consulte Úselo AWS CodeBuild con Amazon Virtual Private Cloud.

> **Note**
>
> Si se especifica una anulación de flota al llamar a la StartBuild API, CodeBuild ignorará la configuración de la VPC del proyecto.

Subredes

Las subredes de VPC que se CodeBuild utilizan para configurar tu configuración de VPC. Tenga en cuenta que las flotas de capacidad reservada solo admiten una subred en una única zona de disponibilidad. Además, asegúrese de que las subredes incluyan una puerta de enlace de NAT.

Grupos de seguridad

Los grupos de seguridad de VPC que se CodeBuild utilizan con su VPC. Asegúrese de que sus grupos de seguridad permitan conexiones salientes.

Rol de servicio de la flota

Define el rol de servicio de su flota a partir de un rol de servicio existente en su cuenta.

Definir configuraciones de proxy (opcional)

Configuraciones de proxy que aplican el control de acceso a la red a las instancias de capacidad reservada. Para obtener más información, consulte Uso AWS CodeBuild con un servidor proxy gestionado.

> **Note**
>
> Las configuraciones de proxy no son compatibles con VPC, Windows o macOS.

Comportamiento predeterminado

Define el comportamiento del tráfico saliente.

Permitir

Permite el tráfico saliente a todos los destinos de forma predeterminada.

Denegar

Deniega el tráfico saliente a todos los destinos de forma predeterminada.

Reglas de proxy

Especifica las direcciones IP o los dominios de destino para permitir o denegar el control de acceso a la red.

Muestras de capacidad reservada con AWS CodeBuild

Estos ejemplos se pueden usar para experimentar con flotas de capacidad reservada en. 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 de las compilaciones en caché para mejorar el rendimiento.

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'
```

Ejecución de compilaciones en lotes

Se puede usar 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, ya que CodeBuild debes poder invocar las `RetryBuild` acciones `StartBuildStopBuild`, y en tu nombre para ejecutar las 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
- Distribución ramificada 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
    - identifier: build4
      env:
        compute-type: ARM_LAMBDA_1GB
    - identifier: build5
      env:
        fleet: fleet_name
```

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-gráfico.

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
    - identifier: build3
      env:
        compute-type: ARM_LAMBDA_1GB
    - identifier: build4
      env:
        fleet: fleet_name
    - identifier: build5
      env:
        compute-type: GENERAL_LINUX_XLAGRE
```

Para obtener más información acerca de la sintaxis de la lista de especificación de compilación, consulte batch/build-lista.

Matriz de compilación

Una matriz de compilación define las tareas con diferentes 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/amazonlinux-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-matriz`.

Distribución ramificada de compilación

Una distribución ramificada de compilación define una tarea que se dividirá en varias compilaciones en el lote. Esto se puede usar para ejecutar pruebas en paralelo. CodeBuild crea una compilación independiente para cada fragmento de casos de prueba en función del valor establecido en el `parallelism` campo.

En el siguiente ejemplo, se define una distribución ramificada de compilación que crea cinco compilaciones que se ejecutan en paralelo.

```
version: 0.2

batch:
   fast-fail: false
   build-fanout:
     parallelism: 5
     ignore-failure: false

phases:
  install:
    commands:
      - npm install
   build:
    commands:
      - mkdir -p test-results
      - cd test-results
      - |
        codebuild-tests-run \
         --test-command 'npx jest --runInBand --coverage' \
         --files-search "codebuild-glob-search '**/test/**/*.test.js'" \
         --sharding-strategy 'equal-distribution'
```

En este ejemplo, suponiendo que hay 100 pruebas que deben ejecutarse, CodeBuild crea cinco compilaciones, cada una de las cuales ejecuta 20 pruebas en paralelo.

Para obtener más información acerca de la sintaxis del grafo de especificación de compilación, consulte `batch/build-abanico`.

Modo de informe por lotes

Si el proveedor de origen de tu proyecto es Bitbucket o GitHub Enterprise GitHub, 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. 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
- Ejecute una compilación por lotes (AWS CLI)
- Detenga las compilaciones por lotes AWS CodeBuild

Ejecución de pruebas paralelas en compilaciones por lotes

Puede usarlo AWS CodeBuild para ejecutar pruebas paralelas en compilaciones por lotes. La ejecución de pruebas paralelas es un método de prueba en el que varios casos de prueba se ejecutan simultáneamente en diferentes entornos, equipos o navegadores, en lugar de ejecutarse secuencialmente. Este método puede reducir significativamente el tiempo total de ejecución de las pruebas y mejorar su eficiencia. En CodeBuild, puedes dividir tus pruebas en varios entornos y ejecutarlas simultáneamente.

Las principales ventajas de la ejecución de pruebas paralelas son:

1. Tiempo de ejecución reducido: pruebas que tardarían horas de forma secuencial pueden completarse en minutos.

2. Mejor utilización de los recursos: hace un uso eficiente de los recursos informáticos disponibles.
3. Comentarios más rápidos: la realización más rápida de las pruebas permite obtener comentarios más rápidamente a los desarrolladores.
4. Cost-effective- Ahorra tiempo y costes informáticos a largo plazo.

Al implementar la ejecución de pruebas paralelas se suelen considerar dos enfoques principales: entornos separados y multiproceso. Si bien ambos métodos buscan conseguir la ejecución de pruebas simultáneas, difieren significativamente en su implementación y efectividad. Los entornos separados crean instancias aisladas en las que cada conjunto de pruebas se ejecuta de forma independiente, mientras que el multiproceso ejecuta varias pruebas simultáneamente dentro del mismo espacio de proceso utilizando diferentes subprocesos.

Las principales ventajas de los entornos separados en comparación con el multiproceso son:

1. Aislamiento: cada prueba se ejecuta en un entorno completamente aislado, lo que evita la interferencia entre pruebas.
2. Conflictos de recursos: no hay competencia por recursos compartidos, algo que suele producirse en el multiproceso.
3. Estabilidad: es menos propenso a condiciones de carrera y a problemas de sincronización.
4. Depuración más sencilla: cuando fallan las pruebas, es más fácil identificar la causa, ya que cada entorno es independiente.
5. Administración del estado: administre fácilmente problemas de estado compartidos que afectan habitualmente a las pruebas multiproceso.
6. Mejor escalabilidad: puede agregar fácilmente más entornos sin complejidad.

Temas

- Soporte en AWS CodeBuild
- Habilitación de la ejecución de pruebas paralelas en compilaciones por lotes
- Utilice el comando codebuild-tests-run de la CLI
- Utilice el comando codebuild-glob-search de la CLI
- Acerca de la división de pruebas
- Combinación automática de informes de compilación individuales
- Ejemplo de ejecución de pruebas paralelas para varios marcos de prueba

Soporte en AWS CodeBuild

AWS CodeBuild proporciona un sólido soporte para la ejecución de pruebas en paralelo a través de su función de creación por lotes, diseñada específicamente para aprovechar la ejecución en entornos independientes. Esta implementación se alinea perfectamente con los beneficios de los entornos de pruebas aislados.

Compilación por lotes con distribución de pruebas

CodeBuildSu funcionalidad de compilación por lotes permite la creación de múltiples entornos de compilación que se ejecutan simultáneamente. Cada entorno funciona como una unidad completamente aislada, con sus propios recursos de computación, entorno de tiempo de ejecución y dependencias. A través de la configuración de compilación por lotes puede especificar cuántos entornos paralelos necesitan y cómo deben distribuirse las pruebas entre ellos.

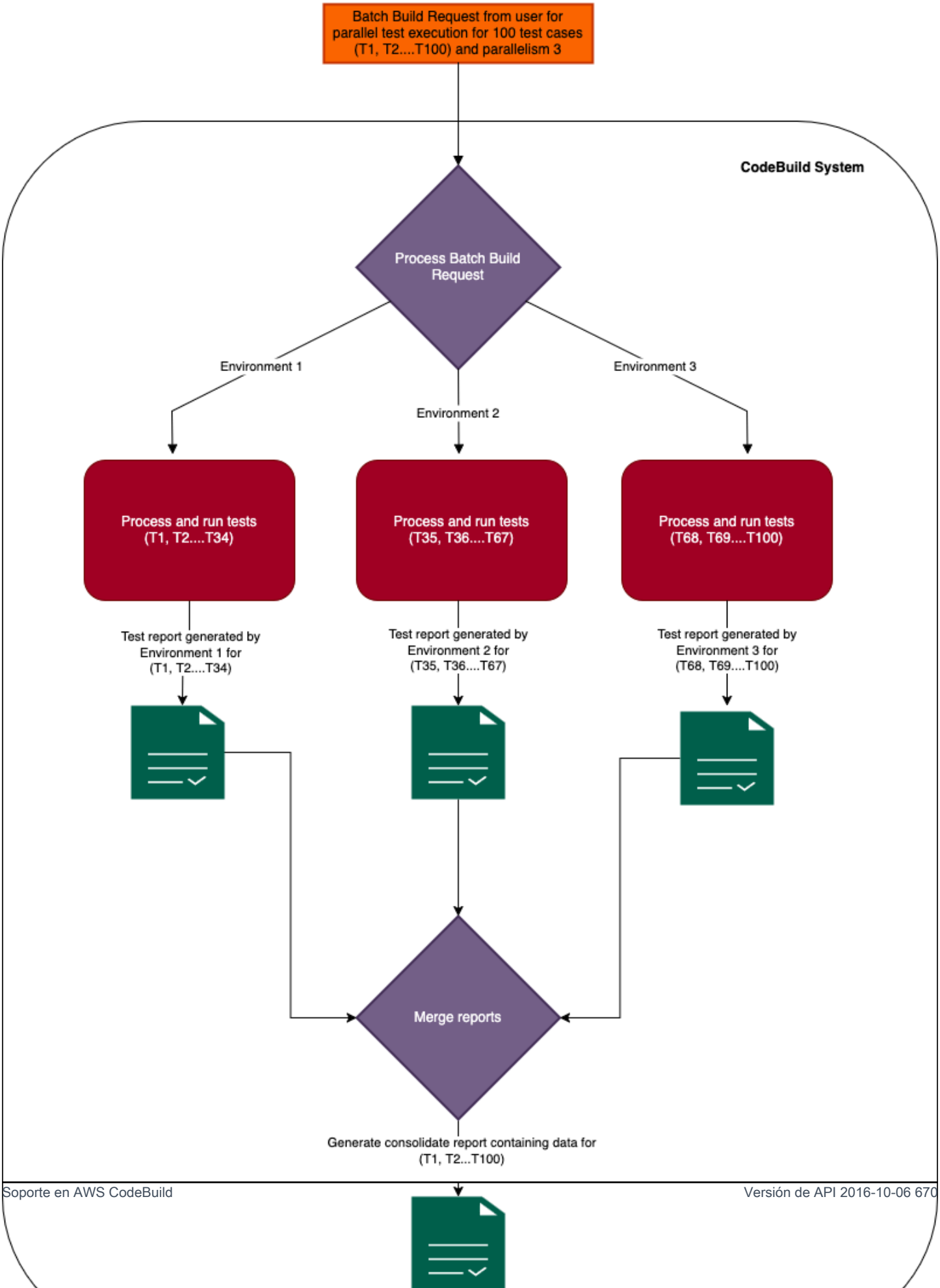
Prueba de CLI de partición

CodeBuild incluye un mecanismo de distribución de pruebas incorporado a través de su herramienta CLI`codebuild-tests-run`, que divide automáticamente las pruebas en diferentes entornos.

Agregación de informes

Uno de los puntos fuertes clave de la implementación CodeBuild es su capacidad para gestionar la agregación de los resultados de las pruebas sin problemas. Si bien las pruebas se ejecutan en entornos independientes, recopila y combina CodeBuild automáticamente los informes de prueba de cada entorno en un informe de prueba unificado a nivel de creación por lotes. Esta consolidación proporciona una vista integral de los resultados de las pruebas y, al mismo tiempo, mantiene los beneficios de eficiencia de la ejecución paralela.

El siguiente diagrama explica el concepto completo de ejecución de pruebas en paralelo en AWS CodeBuild.


Batch Build Request from user for parallel test execution for 100 test cases (T1, T2....T100) and parallelism 3
CodeBuild System
Process Batch Build Request
Environment 1
Environment 2
Environment 3
Process and run tests (T1, T2....T34)
Process and run tests (T35, T36....T67)
Process and run tests (T68, T69....T100)
Test report generated by Environment 1 for (T1, T2....T34)
Test report generated by Environment 2 for (T35, T36....T67)
Test report generated by Environment 3 for (T68, T69....T100)
Merge reports
Generate consolidate report containing data for (T1, T2...T100)

Habilitación de la ejecución de pruebas paralelas en compilaciones por lotes

Para ejecutar pruebas en paralelo, actualice el archivo buildspec de compilación por lotes para incluir el campo build-fanout y el número de compilaciones paralelas para dividir el conjunto de pruebas en el campo `parallelism`, como se muestra a continuación. El `parallelism` campo especifica cuántos ejecutores independientes están configurados para ejecutar el conjunto de pruebas.

Para ejecutar las pruebas en varios entornos de ejecución paralela, establezca el campo `parallelism` en un valor mayor que cero. En el ejemplo siguiente, `parallelism` se establece en cinco, lo que significa que CodeBuild inicia cinco compilaciones idénticas que ejecutan una parte del conjunto de pruebas en paralelo.

Puede usar el comando codebuild-tests-run de la CLI para dividir y ejecutar las pruebas. Los archivos de prueba se dividirán y una parte de las pruebas se ejecutará en cada compilación. Esto reduce el tiempo total necesario para ejecutar todo el conjunto de pruebas. En el siguiente ejemplo, las pruebas se dividirán en cinco y los puntos de división se calcularán en función del nombre de las pruebas.

```
version: 0.2

batch:
  fast-fail: false
  build-fanout:
    parallelism: 5
    ignore-failure: false

phases:
  install:
    commands:
      - npm install jest-junit --save-dev
  pre_build:
    commands:
      - echo 'prebuild'
  build:
    commands:
      - |
        codebuild-tests-run \
         --test-command 'npx jest --runInBand --coverage' \
         --files-search "codebuild-glob-search '**/_tests_/**/*.test.js'" \
         --sharding-strategy 'equal-distribution'
```

```
  post_build:
    commands:
      - codebuild-glob-search '**/*.xml'
      - echo "Running post-build steps..."
      - echo "Build completed on `date`"

reports:
  test-reports:
    files:
      - '**/junit.xml'
    base-directory: .
    discard-paths: yes
    file-format: JUNITXML
```

Si los informes están configurados para la compilación en forma de distribución, los informes de prueba se generan para cada compilación por separado, y se pueden ver en la pestaña Informes de las compilaciones correspondientes de la consola. AWS CodeBuild

Para obtener más información sobre cómo ejecutar pruebas paralelas por lotes, consulte Ejemplo de ejecución de pruebas paralelas para varios marcos de prueba.

Utilice el comando codebuild-tests-run de la CLI

AWS CodeBuild proporciona una CLI que tomará como entrada un comando de prueba y la ubicación del archivo de prueba. La CLI con esta entrada dividirá las pruebas en el número de fragmentos especificado en el `parallelism` campo en función de los nombres de los archivos de prueba. La asignación de archivos de prueba a la partición se decide mediante la estrategia de partición.

```
codebuild-tests-run \
    --files-search "codebuild-glob-search '**/__tests__/*.js'" \
    --test-command 'npx jest --runInBand --coverage' \
    --sharding-strategy 'equal-distribution'
```

En la siguiente tabla se describen los capos del comando `codebuild-tests-run` de la CLI.

Nombre del campo	Tipo	Obligatorio u opcional	Definición
`test-command`	Cadena	Obligatorio	Este comando se usa para ejecutar las pruebas.
`files-search`	Cadena	Obligatorio	Este comando proporciona una lista de archivos de prueba. Puedes usar el comando de la CLI codebuild-glob-search AWS CodeBuild proporcionado o cualquier otra herramienta de búsqueda de archivos que prefieras. **Note** Asegúrese de que el comando `files-search` muestre los nombres de archivo, cada uno de ellos separado por una línea.
`sharding-strategy`	Enum	Opcional	Valores válidos: `equal-distribution` ,

Nombre del campo	Tipo	Obligatorio u opcional	Definición
			(predeterminado), `stability` • `equal-distribution` : particione los archivos de prueba de manera uniforme en función de los nombres de los archivos de prueba. • `stability` : particione los archivos de prueba utilizando un hash coherente de los nombres de archivo. Para obtener más información, consulte Acerca de la división de pruebas.

La CLI `codebuild-tests-run` trabaja primero para identificar la lista de archivos de prueba mediante el comando proporcionado en el parámetro `files-search`. A continuación, determina un subconjunto de archivos de prueba designados para la partición (entorno) actual mediante la estrategia de partición especificada. Por último, este subconjunto de archivos de prueba se formatea en una lista separada por espacios y se añade al final del comando proporcionado en el parámetro `test-command` antes de que se ejecute.

Para los marcos de prueba que no aceptan listas separadas por espacios, la CLI `codebuild-tests-run` proporciona una alternativa flexible a través de la variable de entorno `CODEBUILD_CURRENT_SHARD_FILES`. Esta variable contiene una lista separada por

líneas de rutas de archivos de prueba designadas para la partición de compilación actual. Aprovechando esta variable de entorno, puede adaptarse fácilmente a varios requisitos del marco de pruebas y adaptarse a aquellos que requieren formatos de entrada diferentes de listas separadas por espacios. Además, también puede formatear los nombres de los archivos de prueba según las necesidades del marco de prueba. A continuación se ofrece un ejemplo de uso de `CODEBUILD_CURRENT_SHARD_FILES` en Linux con el marco Django. Aquí se usa `CODEBUILD_CURRENT_SHARD_FILES` para obtener rutas de archivo de notación de puntos compatibles con Django:

```
codebuild-tests-run \
    –files-search "codebuild-glob-search '/tests/test_.py'" \
    –test-command 'python3 manage.py test $(echo "$CODEBUILD_CURRENT_SHARD_FILES" | sed
 -E "s/\//__/g; s/\.py$//; s/__/./g")' \
    –sharding-strategy 'equal-distribution'
```

> **Note**
>
> Tenga en cuenta que la variable de entorno `CODEBUILD_CURRENT_SHARD_FILES` solo se puede utilizar dentro del ámbito de la `codebuild-tests-run` CLI.
> Además, si utiliza `CODEBUILD_CURRENT_SHARD_FILES` dentro de test-command, escriba `CODEBUILD_CURRENT_SHARD_FILES` entre comillas dobles como se muestra en el ejemplo anterior.

Utilice el comando codebuild-glob-search de la CLI

AWS CodeBuild proporciona una herramienta CLI integrada denominada `codebuild-glob-search` que permite buscar archivos en el directorio de trabajo en función de uno o más patrones globales. Esta herramienta puede resultar especialmente útil cuando desee ejecutar pruebas en archivos o directorios específicos del proyecto.

De uso

La CLI `codebuild-glob-search` tiene la siguiente sintaxis de uso:

```
codebuild-glob-search <glob_pattern1> [<glob_pattern2> ...]
```

- *<glob_pattern1>*, *<glob_pattern2>*, etc.: uno o más patrones globales para compararlos con los archivos del directorio de trabajo.
- *: coincide con cualquier secuencia de caracteres (excepto los separadores de ruta).
- **: coincide con cualquier secuencia de caracteres (incluidos los separadores de ruta).

> **Note**
>
> Asegúrese de que la cadena global vaya entre comillas. Para comprobar los resultados de la coincidencia de patrones, utilice el comando `echo`.
>
> ```
> version: 0.2
>
> phases:
>   build:
>     commands:
>       - echo $(codebuild-glob-search '**/__tests__/*.js')
>       - codebuild-glob-search '**/__tests__/*.js' | xargs -n 1 echo
> ```

Output

La CLI generará una nueva lista de rutas de archivo separadas por líneas que coincidan con los patrones globales proporcionados. Las rutas de archivo devueltas serán relativas al directorio de trabajo.

Si no se encuentra ningún archivo que coincida con los patrones proporcionados, la CLI generará un mensaje que indica que no se ha encontrado ningún archivo.

Tenga en cuenta que los directorios que se encuentren debido a un patrón determinado se excluirán de los resultados de la búsqueda.

Ejemplo

Si desea buscar solo archivos dentro del directorio tests y sus subdirectorios con extensión `.js`, puede usar el siguiente comando con la `codebuild-glob-search` CLI:

```
codebuild-glob-search '**/__tests__/*.js'
```

Este comando buscará todos los archivos con una `.js` extensión dentro del `__tests__` directorio y sus subdirectorios, tal y como indica el patrón.

Acerca de la división de pruebas

La característica de división de pruebas de AWS CodeBuild le permite paralelizar la ejecución del conjunto de pruebas en varias instancias de computación, lo que reduce el tiempo total de ejecución de las pruebas. Esta función se habilita mediante la configuración por lotes en los ajustes de tu CodeBuild proyecto y la `codebuild-tests-run` utilidad en tu archivo buildspec.

Las pruebas se dividen en función de la estrategia de fragmentación especificada. CodeBuild proporciona dos estrategias de fragmentación, tal como se especifica a continuación:

Equal-distribution

La estrategia de partición `equal-distribution` divide las pruebas en compilaciones paralelas por orden alfabético de los nombres de los archivos de prueba. Este método ordena primero los archivos de prueba y, a continuación, emplea un método basado en fragmentos para distribuirlos, lo que garantiza que archivos similares se agrupen para las pruebas. Se recomienda cuando se trata de un conjunto relativamente pequeño de archivos de prueba. Si bien este método tiene como objetivo asignar aproximadamente el mismo número de archivos a cada partición, con una diferencia máxima de uno, no garantiza la estabilidad. Cuando se agregan o quitan archivos de prueba en compilaciones posteriores, la distribución de los archivos existentes puede cambiar, lo que podría provocar una reasignación entre particiones.

Stability

La estrategia de partición `stability` emplea un algoritmo de hash coherente para dividir las pruebas entre particiones, lo que garantiza que la distribución de los archivos se mantenga estable. Cuando se agregan o quitan archivos, este método garantiza que las asignaciones existentes de archivos a particiones permanezcan prácticamente inalteradas. En el caso de conjuntos de pruebas de gran tamaño, se recomienda utilizar la opción de estabilidad para distribuir uniformemente las pruebas entre particiones. Este mecanismo tiene como objetivo proporcionar una distribución casi equitativa, lo que garantiza que cada partición reciba una cantidad similar de archivos, con una variación mínima. Si bien la estrategia de estabilidad no garantiza una distribución equitativa ideal, ofrece una distribución casi equitativa que mantiene la coherencia en las asignaciones de archivos entre las compilaciones, incluso cuando se agregan o eliminan archivos.

Para habilitar la división de pruebas, debe configurar la sección de lotes en los ajustes de su CodeBuild proyecto, especificando el `parallelism` nivel deseado y otros parámetros relevantes. Además, deberá incluir la utilidad `codebuild-tests-run` en el archivo buildspec, junto con los comandos de prueba y el método de división adecuados.

Combinación automática de informes de compilación individuales

En las compilaciones por lotes de Fanout, AWS CodeBuild admite la fusión automática de los informes de compilación individuales en un informe consolidado a nivel de lote. Esta característica proporciona una vista completa de los resultados de las pruebas y cobertura de código en todas las compilaciones de un lote.

Funcionamiento

Al ejecutar compilaciones `fanout` por lotes, cada compilación individual genera informes de prueba. ??? CodeBuild luego, consolida automáticamente informes idénticos de diferentes compilaciones en un informe unificado, que se adjunta a la compilación por lotes. Se puede acceder fácilmente a estos informes consolidados a través del `reportArns` campo de la BatchGetBuildBatches API y también se pueden ver en la pestaña Informes de la consola. Esta capacidad de combinación se extiende también a los informes de detección automática.

Los informes consolidados se crean en grupos de informes que se especifican en la especificación de creación o son descubiertos automáticamente por. CodeBuild Puede analizar tendencias de los informes combinados directamente en estos grupos de informes, lo que proporciona información valiosa sobre el rendimiento general de la compilación y las métricas de calidad en compilaciones históricas del mismo proyecto de compilación por lotes.

Para cada compilación individual del lote, se crean CodeBuild automáticamente grupos de informes independientes. Estos siguen una convención de nomenclatura específica que combina el nombre del grupo de informes de creación por lotes con un sufijo de `BuildFanoutShard<shard_number>`, donde el `shard_number` representa el número de partición en el que se crea el grupo de informes. Esta organización le permite realizar un seguimiento y analizar tendencias tanto en la compilación consolidada como individual, lo que proporciona flexibilidad a la hora de supervisar y evaluar sus procesos de compilación.

El informe de creación por lotes sigue la misma estructura que los informes de compilación individuales. Los siguientes campos clave de la pestaña Informe son específicos de los informes de compilación por lotes:

Estado de informe de compilación por lotes

El estado de los informes de creación por lotes sigue reglas específicas según el tipo de informe:

- Informes de pruebas:
  - Correcto: el estado se establece como correcto cuando todos los informes de compilación individuales se han realizado correctamente.
  - Error: el estado se establece como fallido si ha fallado algún informe de compilación individual.
  - Incompleto: el estado se marca como incompleto si falta algún informe de compilación individual o tiene un estado incompleto.
- Informes de cobertura de código:
  - Completo: el estado se establece como completo cuando se han completado todos los informes de compilación individuales.
  - Error: el estado se establece como fallido si ha fallado algún informe de compilación individual.
  - Incompleto: el estado se marca como incompleto si falta algún informe de compilación individual o tiene un estado incompleto.

Resumen de pruebas

El informe de prueba combinado consolida los siguientes campos de todos los informes de compilación individuales:

- duration-in-nano-seconds: tiempo máximo de duración de las pruebas en nanosegundos entre todos los informes de compilación individuales.
- total: recuento combinado de todos los casos de prueba, sumando el número total de pruebas de cada compilación.
- status-counts: proporciona una vista consolidada de estados de las pruebas, como superadas, fallidas u omitidas, que se calcula sumando el recuento de cada tipo de estado en todas las compilaciones individuales.

Resumen de cobertura de código

El informe combinado de cobertura de código combina campos de todas las compilaciones individuales utilizando los siguientes cálculos:

- branches-covered: suma de todas las ramificaciones cubiertas de informes individuales.
- branches-missed: suma de todas las ramificaciones omitidas de informes individuales.

- branch-coverage-percentage: `(Total covered branches / Total branches) * 100`
- lines-covered: suma de todas las líneas cubiertas de informes individuales.
- lines-missed: suma de todas las líneas omitidas de informes individuales.
- lines-coverage-percentage: `(Total covered lines / Total lines) * 100`

ID de ejecución

ARN de compilación por lotes.

Casos de prueba

El informe combinado contiene una lista consolidada de todos los casos de prueba de versiones individuales, a la que se puede acceder mediante la DescribeTestCases API y el informe de creación por lotes de la consola.

Coberturas de código

El informe de cobertura de código combinado proporciona información consolidada sobre la cobertura de líneas y sucursales de cada archivo en todas las compilaciones individuales, a la que se puede acceder a través de la DescribeCodeCoverages API y el informe de creación por lotes de la consola. Nota: En el caso de los archivos cubiertos por varios archivos de prueba distribuidos en diferentes particiones, el informe combinado sigue estos criterios de selección:

1. La selección principal se basa en la cobertura de línea más alta entre las particiones.
2. Si la cobertura de línea es igual en varias particiones, se seleccione la partición con mayor cobertura de ramificación.

Ejemplo de ejecución de pruebas paralelas para varios marcos de prueba

Puede utilizar el comando `codebuild-tests-run` de la CLI para dividir y ejecutar pruebas en entornos de ejecución paralelos. En la siguiente sección, se proporcionan ejemplos de `buildspec.yml` para varios marcos que ilustran el uso del comando `codebuild-tests-run`.

- Cada uno de los siguientes ejemplos incluye un nivel de `parallelism` de cinco, lo que significa que se crearán cinco entornos de ejecución idénticos para dividir las pruebas. Puede elegir un nivel de `parallelism` que se ajuste a su proyecto modificando el valor de `parallelism` en la sección `build-fanout`.
- En cada uno de los ejemplos siguientes se muestra la configuración de las pruebas para que se dividan por el nombre del archivo de prueba, que es el predeterminado. Esto distribuye las pruebas de manera uniforme entre los entornos de ejecución paralelos.

Antes de empezar, consulte Ejecución de pruebas paralelas en compilaciones por lotes para obtener más información.

Para obtener una lista completa de opciones al utilizar el comando `codebuild-tests-run` de la CLI, consulte Utilice el comando codebuild-tests-run de la CLI .

Temas

- Configuración de pruebas paralelas con Django
- Configuración de pruebas paralelas con Elixir
- Configuración de pruebas paralelas con Go
- Configuración de pruebas paralelas con Java (Maven)
- Configuración de pruebas paralelas con Javascript (Jest)
- Configuración de pruebas paralelas con Kotlin
- Configuración de pruebas paralelas con PHPUnit
- Configuración de pruebas paralelas con Pytest
- Configuración de pruebas paralelas con Ruby (Cucumber)
- Configuración de pruebas paralelas con Ruby (Cucumber)

Configuración de pruebas paralelas con Django

A continuación se ofrece un ejemplo de `buildspec.yml` que muestra la ejecución de pruebas paralelas con Django en una plataforma Ubuntu:

```
version: 0.2

batch:
  fast-fail: false
  build-fanout:
    parallelism: 5

phases:
  install:
    commands:
      - echo 'Installing Python dependencies'
      - sudo yum install -y python3 python3-pip
      - python3 -m ensurepip --upgrade
      - python3 -m pip install django
```

```
  pre_build:
    commands:
      - echo 'Prebuild'
  build:
    commands:
      - echo 'Running Django Tests'
      - |
        codebuild-tests-run \
         --test-command 'python3 manage.py test $(echo "$CODEBUILD_CURRENT_SHARD_FILES"
 | sed -E "s/\//__/g; s/\.py$//; s/__/./g")' \
         --files-search "codebuild-glob-search '**/tests/*test_*.py'" \
         --sharding-strategy 'equal-distribution'
  post_build:
    commands:
      - echo 'Test execution completed'
```

El ejemplo anterior muestra el uso de la variable de entorno `CODEBUILD_CURRENT_SHARD_FILES`. Aquí `CODEBUILD_CURRENT_SHARD_FILES` se usa para obtener rutas de archivo de notación de puntos de recuperación compatibles con Django. Use `CODEBUILD_CURRENT_SHARD_FILES` entre comillas dobles como se muestra arriba.

Configuración de pruebas paralelas con Elixir

A continuación se ofrece un ejemplo de `buildspec.yml` que muestra la ejecución de pruebas paralelas con Elixir en una plataforma Ubuntu:

```
version: 0.2

batch:
  fast-fail: false
  build-fanout:
    parallelism: 5

phases:
  install:
    commands:
      - echo 'Installing Elixir dependencies'
      - sudo apt update
      - sudo DEBIAN_FRONTEND=noninteractive apt install -y elixir
      - elixir --version
      - mix --version
  pre_build:
    commands:
```

```
      - echo 'Prebuild'
  build:
    commands:
      - echo 'Running Elixir Tests'
      - |
        codebuild-tests-run \
         --test-command 'mix test' \
         --files-search "codebuild-glob-search '**/test/**/*_test.exs'" \
         --sharding-strategy 'equal-distribution'
  post_build:
    commands:
      - echo "Test execution completed"
```

Configuración de pruebas paralelas con Go

El siguiente es un ejemplo de un ejemplo `buildspec.yml` que muestra la ejecución de pruebas en paralelo con Go en una plataforma Linux:

```
version: 0.2

batch:
  fast-fail: false
  build-fanout:
    parallelism: 5
    ignore-failure: false

phases:
  install:
    commands:
      - echo 'Fetching Go version'
      - go version
  pre_build:
    commands:
      - echo 'prebuild'
  build:
    commands:
      - echo 'Running go Tests'
      - go mod init calculator
      - cd calc
      - |
        codebuild-tests-run \
         --test-command "go test -v calculator.go" \
         --files-search "codebuild-glob-search '**/*test.go'"
```

```
  post_build:
    commands:
      - echo "Test execution completed"
```

En el ejemplo anterior, la `calculator.go` función contiene funciones matemáticas sencillas para probar y todos los archivos de prueba y el `calculator.go` archivo están dentro de `calc` la carpeta.

Configuración de pruebas paralelas con Java (Maven)

El siguiente es un ejemplo de una `buildspec.yml` que muestra la ejecución de pruebas en paralelo con Java en una plataforma Linux:

```
version: 0.2

batch:
  fast-fail: false
  build-fanout:
    parallelism: 5
    ignore-failure: false

phases:
  pre_build:
    commands:
      - echo 'prebuild'
  build:
    commands:
      - echo "Running mvn test"
      - |
        codebuild-tests-run \
          --test-command 'mvn test -Dtest=$(echo "$CODEBUILD_CURRENT_SHARD_FILES" | sed
 "s|src/test/java/||g; s/\.java//g; s|/|.|g; s/ /,/g" | tr "\n" "," | sed "s/,$//")' \
          --files-search "codebuild-glob-search '**/test/**/*.java'"

  post_build:
    commands:
      - echo "Running post-build steps..."
      - echo "Test execution completed"
```

En el ejemplo dado, la variable de entorno `CODEBUILD_CURRENT_SHARD_FILES` contiene archivos de prueba en la partición actual, separados por líneas. Estos archivos se convierten en una lista de nombres de clases separados por comas en el formato aceptado por el parámetro `-Dtest` para Maven.

Configuración de pruebas paralelas con Javascript (Jest)

A continuación se ofrece un ejemplo de `buildspec.yml` que muestra la ejecución de pruebas paralelas con Javascript en una plataforma Ubuntu:

```
version: 0.2

batch:
  fast-fail: true
  build-fanout:
    parallelism: 5
    ignore-failure: false

phases:
  install:
    commands:
      - echo 'Installing Node.js dependencies'
      - apt-get update
      - apt-get install -y nodejs
      - npm install
      - npm install --save-dev jest-junit
  pre_build:
    commands:
      - echo 'prebuild'
  build:
    commands:
      - echo 'Running JavaScript Tests'
      - |
         codebuild-tests-run \
          --test-command "npx jest" \
          --files-search "codebuild-glob-search '**/test/**/*.test.js'" \
          --sharding-strategy 'stability'
    post_build:
      commands:
        - echo 'Test execution completed'
```

Configuración de pruebas paralelas con Kotlin

El siguiente es un ejemplo de una `buildspec.yml` que muestra la ejecución de pruebas en paralelo con Kotlin en una plataforma Linux:

```
version: 0.2
```

```
batch:
  fast-fail: false
  build-fanout:
    parallelism: 2
    ignore-failure: false

phases:
  install:
    runtime-versions:
      java: corretto11
    commands:
      - echo 'Installing dependencies'
      - KOTLIN_VERSION="1.8.20" # Replace with your desired version
      - curl -o kotlin-compiler.zip -L "https://github.com/JetBrains/kotlin/releases/
download/v${KOTLIN_VERSION}/kotlin-compiler-${KOTLIN_VERSION}.zip"
      - unzip kotlin-compiler.zip -d /usr/local
      - export PATH=$PATH:/usr/local/kotlinc/bin
      - kotlin -version
      - curl -O https://repo1.maven.org/maven2/org/junit/platform/junit-platform-
console-standalone/1.8.2/junit-platform-console-standalone-1.8.2.jar
  pre_build:
    commands:
      - echo 'prebuild'
  build:
    commands:
      - echo 'Running Kotlin Tests'
      - |
        codebuild-tests-run \
          --test-command 'kotlinc src/main/kotlin/*.kt $(echo
 "$CODEBUILD_CURRENT_SHARD_FILES" | tr "\n" " ") -d classes -cp junit-platform-console-
standalone-1.8.2.jar' \
          --files-search "codebuild-glob-search 'src/test/kotlin/*.kt'"
      - |
        codebuild-tests-run \
          --test-command '
            java -jar junit-platform-console-standalone-1.8.2.jar --class-path classes
 \
              $(for file in $CODEBUILD_CURRENT_SHARD_FILES; do
                 class_name=$(basename "$file" .kt)
                 echo "--select-class $class_name"
               done)
          ' \
          --files-search "codebuild-glob-search 'src/test/kotlin/*.kt'"
  post_build:
```

```
    commands:
      - echo "Test execution completed"
```

En el ejemplo anterior, la CLI `codebuild-tests-run` se usa dos veces. Durante la primera ejecución, kotlinc compila los archivos. La variable `CODEBUILD_CURRENT_SHARD_FILES` recupera los archivos de prueba asignados a la partición actual, que luego se convierten en una lista separada por espacios. En la segunda ejecución, JUnit ejecuta las pruebas. Una vez más, `CODEBUILD_CURRENT_SHARD_FILES` recupera los archivos de prueba asignados a la partición actual, pero esta vez se convierten en nombres de clases.

Configuración de pruebas paralelas con PHPUnit

El siguiente es un ejemplo de una `buildspec.yml` que muestra la ejecución de pruebas en paralelo con PHPUnit en una plataforma Linux:

```
version: 0.2

batch:
   fast-fail: false
   build-fanout:
     parallelism: 5
     ignore-failure: false

phases:
   install:
     commands:
       - echo 'Install dependencies'
       - composer require --dev phpunit/phpunit
   pre_build:
     commands:
       - echo 'prebuild'
   build:
     commands:
       - echo 'Running phpunit Tests'
       - composer dump-autoload
       - |
         codebuild-tests-run \
          --test-command "./vendor/bin/phpunit --debug" \
          --files-search "codebuild-glob-search '**/tests/*Test.php'"
   post_build:
       commands:
         - echo 'Test execution completed'
```

Configuración de pruebas paralelas con Pytest

A continuación se ofrece un ejemplo de `buildspec.yml` que muestra la ejecución de pruebas paralelas con Pytest en una plataforma Ubuntu:

```
version: 0.2

batch:
  fast-fail: false
  build-fanout:
    parallelism: 5
    ignore-failure: false

phases:
  install:
    commands:
      - echo 'Installing Python dependencies'
      - apt-get update
      - apt-get install -y python3 python3-pip
      - pip3 install --upgrade pip
      - pip3 install pytest
  build:
    commands:
      - echo 'Running Python Tests'
      - |
         codebuild-tests-run \
          --test-command 'python -m pytest' \
          --files-search "codebuild-glob-search 'tests/test_*.py'" \
          --sharding-strategy 'equal-distribution'
  post_build:
    commands:
      - echo "Test execution completed"
```

A continuación se ofrece un ejemplo de `buildspec.yml` que muestra la ejecución de pruebas paralelas con Pytest en una plataforma Windows:

```
version: 0.2

batch:
  fast-fail: false
  build-fanout:
    parallelism: 5
```

```
    ignore-failure: false

phases:
  install:
    commands:
      - echo 'Installing Python dependencies'
      - pip install pytest
  pre_build:
    commands:
      - echo 'prebuild'
  build:
    commands:
      - echo 'Running pytest'
      - |
        & codebuild-tests-run `
         --test-command 'pytest @("$env:CODEBUILD_CURRENT_SHARD_FILES" -split \"`r?`n
\")'  `
         --files-search "codebuild-glob-search '**/test_*.py' '**/*_test.py'" `
         --sharding-strategy 'equal-distribution'
  post_build:
    commands:
      - echo "Test execution completed"
```

En el ejemplo anterior, la variable de `CODEBUILD_CURRENT_SHARD_FILES` entorno se usa para recuperar los archivos de prueba asignados al fragmento actual y pasarlos como matriz al comando pytest.

Configuración de pruebas paralelas con Ruby (Cucumber)

El siguiente es un ejemplo de una `buildspec.yml` que muestra la ejecución de pruebas paralelas con Cucumber en una plataforma Linux:

```
version: 0.2

batch:
  fast-fail: false
  build-fanout:
    parallelism: 5
    ignore-failure: false

phases:
  install:
    commands:
```

```
      - echo 'Installing Ruby dependencies'
      - gem install bundler
      - bundle install
  pre_build:
    commands:
      - echo 'prebuild'
  build:
    commands:
      - echo 'Running Cucumber Tests'
      - cucumber --init
      - |
        codebuild-tests-run \
         --test-command "cucumber" \
         --files-search "codebuild-glob-search '**/*.feature'"
  post_build:
    commands:
      - echo "Test execution completed"
```

Configuración de pruebas paralelas con Ruby (Cucumber)

A continuación se ofrece un ejemplo de `buildspec.yml` que muestra la ejecución de pruebas paralelas con RSpec en una plataforma Ubuntu:

```
version: 0.2

batch:
  fast-fail: false
  build-fanout:
    parallelism: 5
    ignore-failure: false

phases:
  install:
    commands:
      - echo 'Installing Ruby dependencies'
      - apt-get update
      - apt-get install -y ruby ruby-dev build-essential
      - gem install bundler
      - bundle install
  build:
    commands:
      - echo 'Running Ruby Tests'
      - |
```

```
        codebuild-tests-run \
         --test-command 'bundle exec rspec' \
         --files-search "codebuild-glob-search 'spec/**/*_spec.rb'" \
         --sharding-strategy 'equal-distribution'
  post_build:
    commands:
      - echo "Test execution completed"
```

Almacenamiento de las compilaciones en caché para mejorar el rendimiento

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 otorga 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
- Especificación de una 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.

Para usar Amazon S3 en una compilación, puede especificar las rutas de los archivos en los que desea almacenar en caché. `buildspec.yml` CodeBuild almacenará y actualizará automáticamente la caché en la ubicación de Amazon S3 configurada en el proyecto. Si no especifica las rutas de los archivos, CodeBuild haremos todo lo posible por almacenar en caché las dependencias lingüísticas comunes para acelerar las compilaciones. Puede ver los detalles de la memoria caché en los registros de compilación.

Además, si quieres tener varias versiones de la caché, puedes definir una clave de caché en. `buildspec.yml` CodeBuild almacena la caché en el contexto de esta clave de caché y crea una copia de caché única que no se actualizará una vez creada. Las claves de caché también se pueden compartir entre proyectos. Las características como claves dinámicas, control de versiones de caché y uso compartido de memoria caché entre compilaciones solo están disponibles cuando se especifica una clave.

Para obtener más información sobre la sintaxis de la caché en el archivo buildspec, consulte cache en la referencia de buildspec.

Temas

- Generación de claves dinámicas
- codebuild-hash-files
- Versión de caché
- Uso compartido de caché entre proyectos
- Ejemplos de buildspec

Generación de claves dinámicas

Una clave de caché puede incluir comandos de shell y variables de entorno para hacerla única, lo que permite actualizar automáticamente la caché cuando la clave cambia. Por ejemplo, puede definir una clave utilizando el hash del archivo `package-lock.json`. Cuando las dependencias de ese archivo cambian, cambia el hash y, por tanto, la clave de caché, lo que desencadena la creación automática de una nueva caché.

```
cache:
    key: npm-key-$(codebuild-hash-files package-lock.json)
```

CodeBuild evaluará la expresión `$(codebuild-hash-files package-lock.json)` para obtener la clave final:

```
npm-key-abc123
```

También puede definir una clave de caché mediante variables de entorno, como `CODEBUILD_RESOLVED_SOURCE_VERSION`. Esto garantiza que cada vez que cambie la fuente, se genere una nueva clave, lo que provocará que se guarde automáticamente una nueva caché:

```
cache:
   key: npm-key-$CODEBUILD_RESOLVED_SOURCE_VERSION
```

CodeBuild evaluará la expresión y obtendrá la clave final:

```
npm-key-046e8b67481d53bdc86c3f6affdd5d1afae6d369
```

codebuild-hash-files

`codebuild-hash-files`es una herramienta de CLI que calcula un SHA-256 hash para un conjunto de archivos del directorio de CodeBuild origen mediante patrones globales:

```
codebuild-hash-files <glob-pattern-1> <glob-pattern-2> ...
```

Algunos ejemplos de uso de `codebuild-hash-files`:

```
codebuild-hash-files package-lock.json
codebuild-hash-files '**/*.md'
```

Versión de caché

La versión de caché es un hash generado a partir de las rutas de los directorios que se almacenan en caché. Si dos cachés tienen versiones diferentes, se tratan como cachés distintas durante el proceso de coincidencia. Por ejemplo, las dos cachés siguientes se consideran diferentes porque hacen referencia a rutas diferentes:

```
version: 0.2

phases:
  build:
    commands:
```

```
      - pip install pandas==2.2.3 --target pip-dependencies
cache:
  key: pip-dependencies
  paths:
    - "pip-dependencies/**/*"
```

```
version: 0.2

phases:
  build:
    commands:
      - pip install pandas==2.2.3 --target tmp/pip-dependencies
cache:
  key: pip-dependencies
  paths:
    - "tmp/pip-dependencies/**/*"
```

Uso compartido de caché entre proyectos

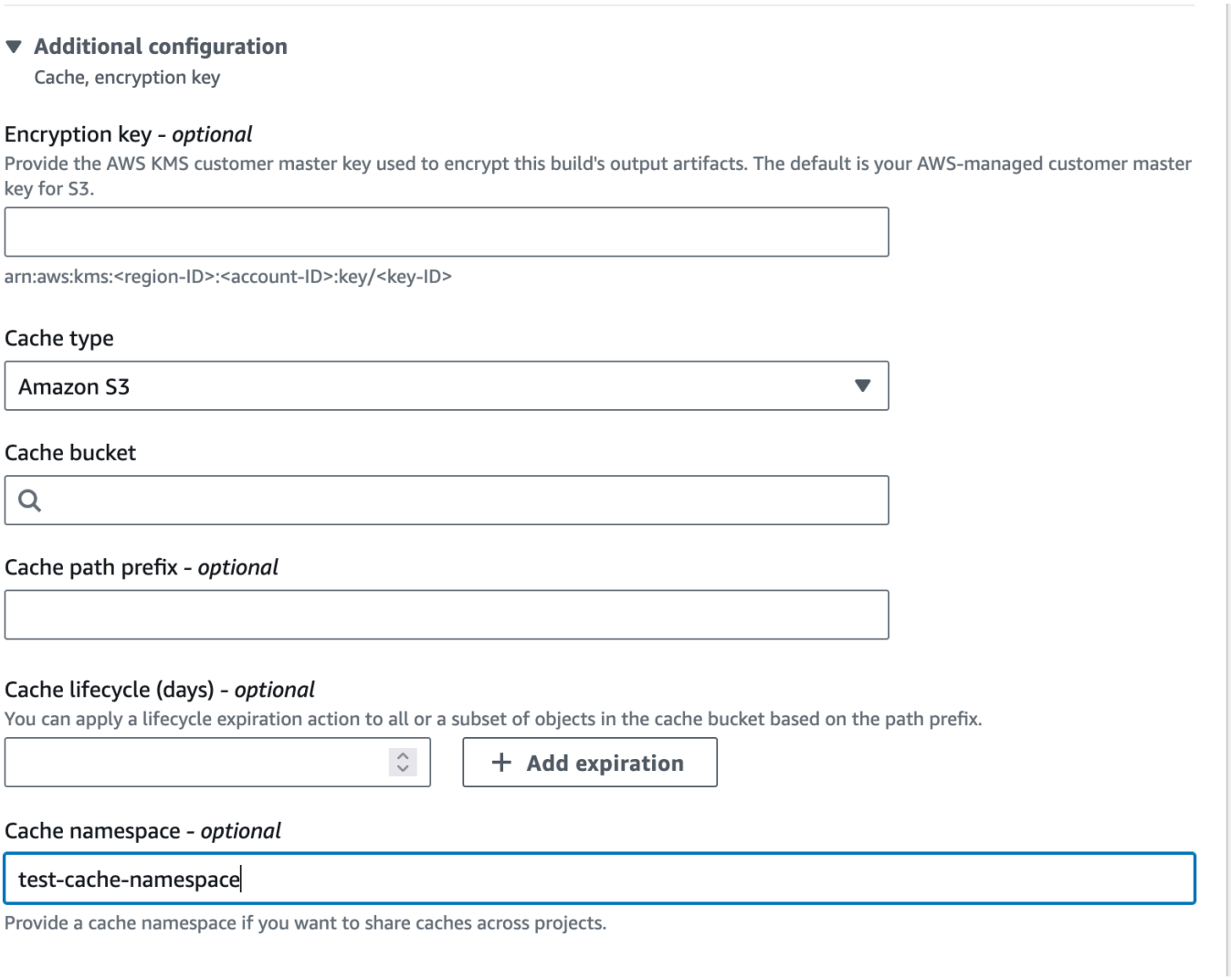
Puede usar el campo `cacheNamespace` de la API en la sección `cache` para compartir una caché entre varios proyectos. Este campo define el ámbito de la caché. Para compartir una caché, debe hacer lo siguiente:

- Usa lo mismo`cacheNamespace`.
- Especifique la misma `key` de caché.
- Defina rutas de caché idénticas.
- Utilice los mismos buckets de Amazon S3 y `pathPrefix` si están configurados.

Esto garantiza la coherencia y permite compartir la caché entre los proyectos.

Especificación de un espacio de nombres de caché (consola)

1. Abra la AWS CodeBuild consola en https://console.aws.amazon.com/codesuite/codebuild/home.
2. Elija Crear proyecto. Para obtener más información, consulte Creación de un proyecto de compilación (consola) y Ejecutar una compilación (consola).
3. En Artefactos, elija Configuración adicional.
4. En Tipo de caché, elija Amazon S3.
5. En Espacio de nombres de caché: opcional, introduzca un valor de espacio de nombres.



6. Continúe con los valores predeterminados y, a continuación, elija Crear el proyecto de compilación.

Especifique un espacio de nombres de caché (AWS CLI)

Puede usar el `--cache` parámetro de AWS CLI para especificar un espacio de nombres de caché.

```
--cache '{"type": "S3", "location": "your-s3-bucket", "cacheNamespace": "test-cache-namespace"}'
```

Ejemplos de buildspec

A continuación, se ofrecen varios ejemplos de buildspec para idiomas comunes:

Temas

- Dependencias de la caché Node.js
- Dependencias de Python de caché
- Dependencias de Ruby de caché
- Dependencias de Go de caché

Dependencias de la caché Node.js

Si tu proyecto incluye un `package-lock.json` archivo y lo usa `npm` para administrar Node.js dependencias, el siguiente ejemplo muestra cómo configurar el almacenamiento en caché. De forma predeterminada, `npm` instala dependencias en el directorio `node_modules`.

```
version: 0.2

phases:
  build:
    commands:
      - npm install
cache:
  key: npm-$(codebuild-hash-files package-lock.json)
  paths:
    - "node_modules/**/*"
```

Dependencias de Python de caché

Si el proyecto incluye un archivo `requirements.txt` y utiliza pip para administrar dependencias de Python, en el siguiente ejemplo se muestra cómo configurar almacenamiento en caché. De forma predeterminada, pip instala paquetes en el directorio `site-packages` del sistema.

```
version: 0.2

phases:
  build:
    commands:
      - pip install -r requirements.txt
cache:
  key: python-$(codebuild-hash-files requirements.txt)
  paths:
    - "/root/.pyenv/versions/${python_version}/lib/python${python_major_version}/site-
packages/**/*"
```

Además, puede instalar dependencias en un directorio específico y configurar el almacenamiento en caché para ese directorio.

```
version: 0.2

phases:
  build:
    commands:
      - pip install -r requirements.txt --target python-dependencies
cache:
  key: python-$(codebuild-hash-files requirements.txt)
  paths:
    - "python-dependencies/**/*"
```

Dependencias de Ruby de caché

Si el proyecto incluye un archivo `Gemfile.lock` y utiliza `Bundler` para administrar dependencias de gemas, en el siguiente ejemplo se muestra cómo configurar correctamente almacenamiento en caché.

```
version: 0.2

phases:
  build:
    commands:
      - bundle install --path vendor/bundle
cache:
  key: ruby-$(codebuild-hash-files Gemfile.lock)
  paths:
    - "vendor/bundle/**/*"
```

Dependencias de Go de caché

Si el proyecto incluye un archivo `go.sum` y utiliza módulos de Go para administrar dependencias, en el siguiente ejemplo se muestra cómo configurar almacenamiento en caché. De forma predeterminada, los módulos de Go se descargan y almacenan en el directorio `${GOPATH}/pkg/mod`.

```
version: 0.2

phases:
  build:
```

```
    commands:
      - go mod download
cache:
  key: go-$(codebuild-hash-files go.sum)
  paths:
    - "/go/pkg/mod/**/*"
```

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 Enterprise Server o Bitbucket), la opción se ignora. GitHub
- 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 sin VPC. Si quiere usar contenedores de Docker para las compilaciones de VPC, consulte

Runtime Privilege and Linux Capabilities en el sitio web de Docker Docs y habilite 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 en caché anulan los elementos de origen si tienen el mismo nombre. Los directorios se especifican utilizando rutas de caché en la archivo buildspec. Para obtener más información, consulte Sintaxis de buildspec.
  - Evite que los nombres de directorio sean los mismos en la fuente y en la caché. Locally-cached los directorios pueden anular o eliminar el contenido de los directorios del repositorio de origen que tengan 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 que funcione con una VPC. Para obtener más información sobre el uso de VPC con CodeBuild, consulte. Úselo AWS CodeBuild con Amazon Virtual Private Cloud

Especificación de una caché local

Puede usar la consola AWS CLI, el SDK o CloudFormation especificar una caché local. Para obtener más información acerca del almacenamiento en caché local, consulte Almacenamiento en la caché local.

Temas

- Especificar el almacenamiento en la caché local (CLI)
- Especificar el almacenamiento en la caché local (consola)
- Especifique el almacenamiento en caché local (CloudFormation)

Especificar el almacenamiento en la caché local (CLI)

Puede usar el `--cache` parámetro en 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]
```

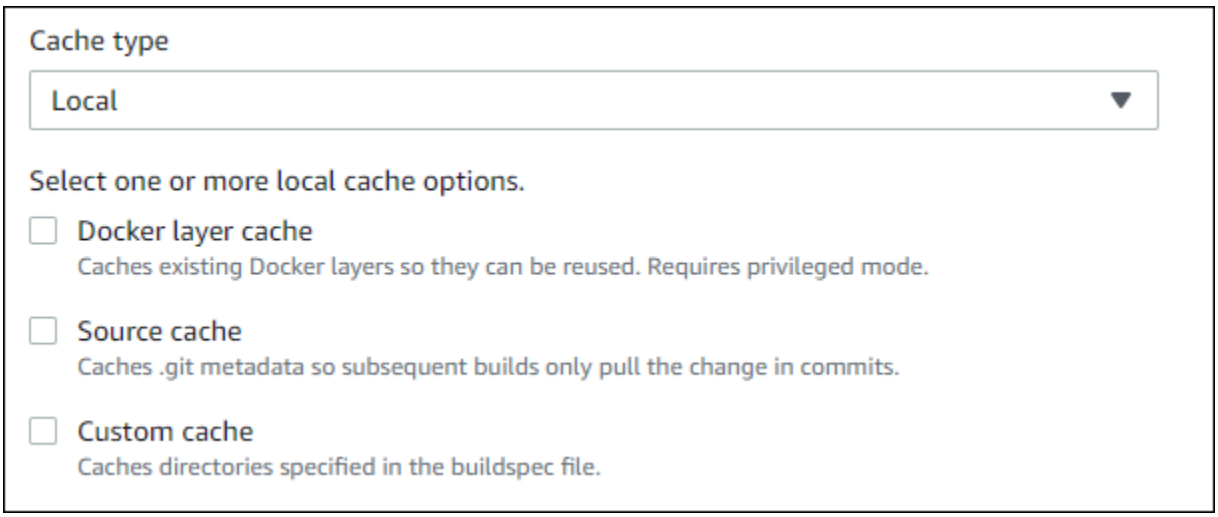
- Para especificar una caché personalizada:

```
--cache type=LOCAL,mode=[LOCAL_CUSTOM_CACHE]
```

Para obtener más información, consulte Cree 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 los tres opciones de caché local.



Para obtener más información, consulte Creación de un proyecto de compilación (consola).

Especifique el almacenamiento en caché local (CloudFormation)

Si utiliza CloudFormation para especificar una caché local, en la `Cache` propiedad, para`Type`, especifique`LOCAL`. El siguiente YAML-formatted CloudFormation código de ejemplo especifica 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 sin VPC. Si quiere usar contenedores de Docker para las compilaciones de VPC, consulte Runtime Privilege and Linux Capabilities en el sitio web de Docker Docs y habilite el modo privilegiado. Además, Windows no admite el modo privilegiado.

Para obtener más información, consulte Cree un proyecto de compilación (CloudFormation).

Depurar compilaciones AWS CodeBuild

AWS CodeBuild proporciona dos métodos para depurar las compilaciones durante el desarrollo y la solución de problemas. Puede usar el entorno CodeBuild Sandbox para investigar problemas y validar las correcciones en tiempo real, o puede usar el administrador de sesiones de AWS Systems Manager para conectarse al contenedor de compilación y ver su estado.

Depure compilaciones con sandbox CodeBuild

El entorno CodeBuild sandbox proporciona una sesión de depuración interactiva en un entorno seguro y aislado. Puede interactuar con el entorno directamente a través del Consola de administración de AWS o AWS CLI, ejecutar comandos y validar el proceso de compilación paso a paso. Utiliza un modelo de facturación por segundo rentable y admite la misma integración nativa con los proveedores y AWS servicios de origen que su entorno de compilación. También puede conectarse a un entorno de pruebas mediante clientes de SSH o desde entornos de desarrollo integrados (IDE).

Para obtener más información sobre los precios de CodeBuild sandbox, consulta la documentación de CodeBuild precios. Para obtener instrucciones detalladas, consulte la documentación de Depure compilaciones con sandbox CodeBuild.

Depuración de compilaciones con Administrador de sesiones

AWS El administrador de sesiones de Systems Manager permite el acceso directo a las compilaciones en ejecución en su entorno de ejecución real. Este método le permite conectarse a contenedores de compilación activos e inspeccionar el proceso de compilación en tiempo real. Puede examinar el sistema de archivos, supervisar los procesos en ejecución y solucionar problemas a medida que se produzcan.

Para obtener instrucciones detalladas, consulte la documentación de Depuración de compilaciones con Administrador de sesiones.

Depure compilaciones con sandbox CodeBuild

En AWS CodeBuild, puedes depurar una compilación mediante el uso de CodeBuild sandbox para ejecutar comandos personalizados y solucionar problemas de tu compilación.

Temas

- Requisitos previos
- Depure compilaciones con CodeBuild sandbox (consola)
- Depure compilaciones con sandbox (CodeBuild AWS CLI)
- Tutorial: Conexión a un entorno de pruebas mediante SSH
- Resolución de problemas AWS CodeBuild Problemas de conexión SSH en Sandbox

Requisitos previos

Antes de usar un CodeBuild entorno aislado, asegúrate de que tu rol de CodeBuild servicio tenga la siguiente política de SSM:

JSON

```
{
    "Version":"2012-10-17",
```

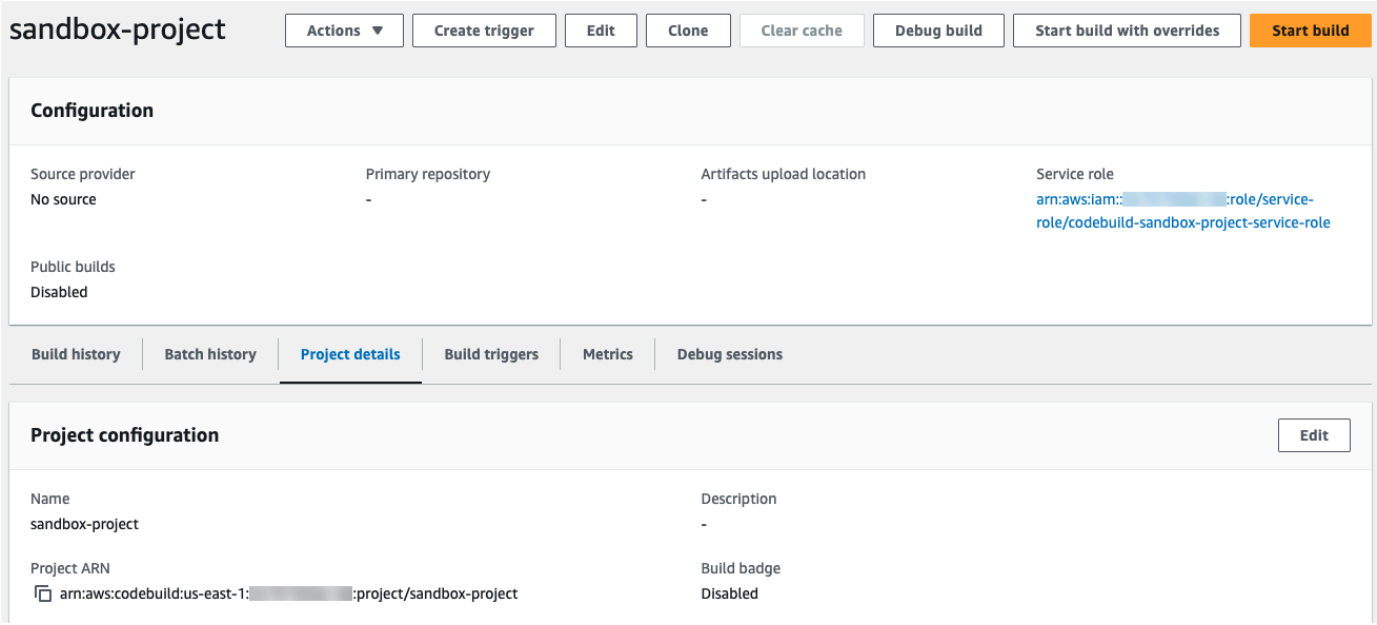
```
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssmmessages:CreateControlChannel",
                "ssmmessages:CreateDataChannel",
                "ssmmessages:OpenControlChannel",
                "ssmmessages:OpenDataChannel"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:StartSession"
            ],
            "Resource": [
                "arn:aws:codebuild:us-east-1:111122223333:build/*",
                "arn:aws:ssm:us-east-1::document/AWS-StartSSHSession"
            ]
        }
    ]
}
```

Depure compilaciones con CodeBuild sandbox (consola)

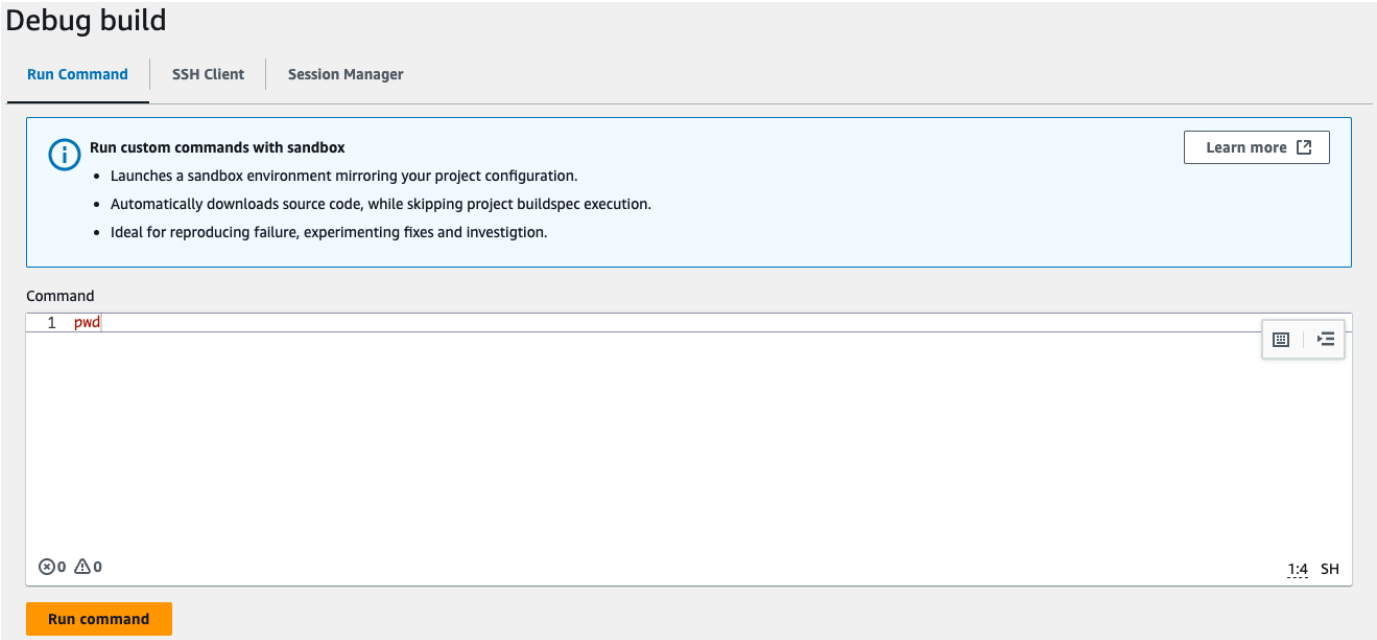
Usa las siguientes instrucciones para ejecutar comandos y conectar tu cliente SSH con el entorno limitado de la CodeBuild consola.

Ejecuta comandos con CodeBuild sandbox (consola)

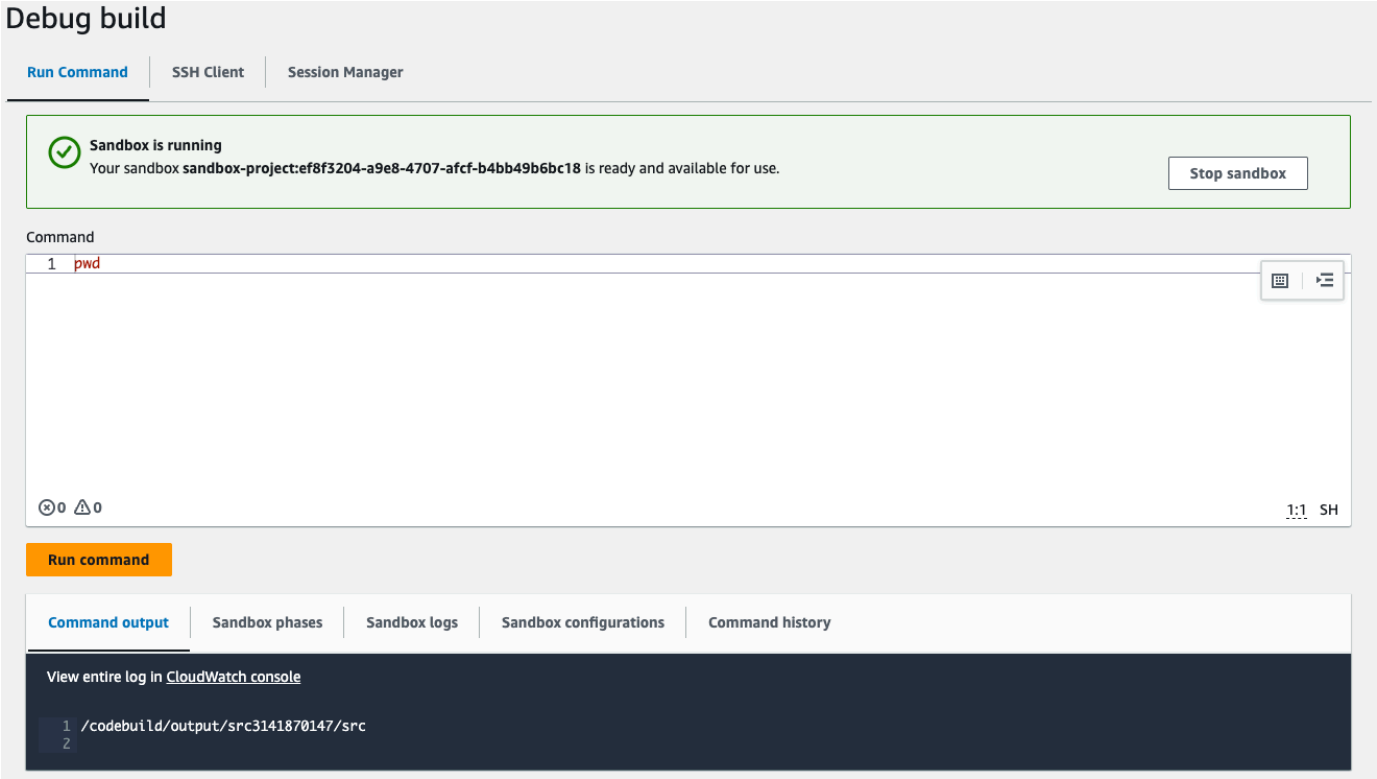
1. Abre la AWS CodeBuild consola en. https://console.aws.amazon.com/codesuite/codebuild/home
2. En el panel de navegación, elija Proyectos de compilación. Elija el proyecto de compilación y, a continuación, elija Iniciar depuración.



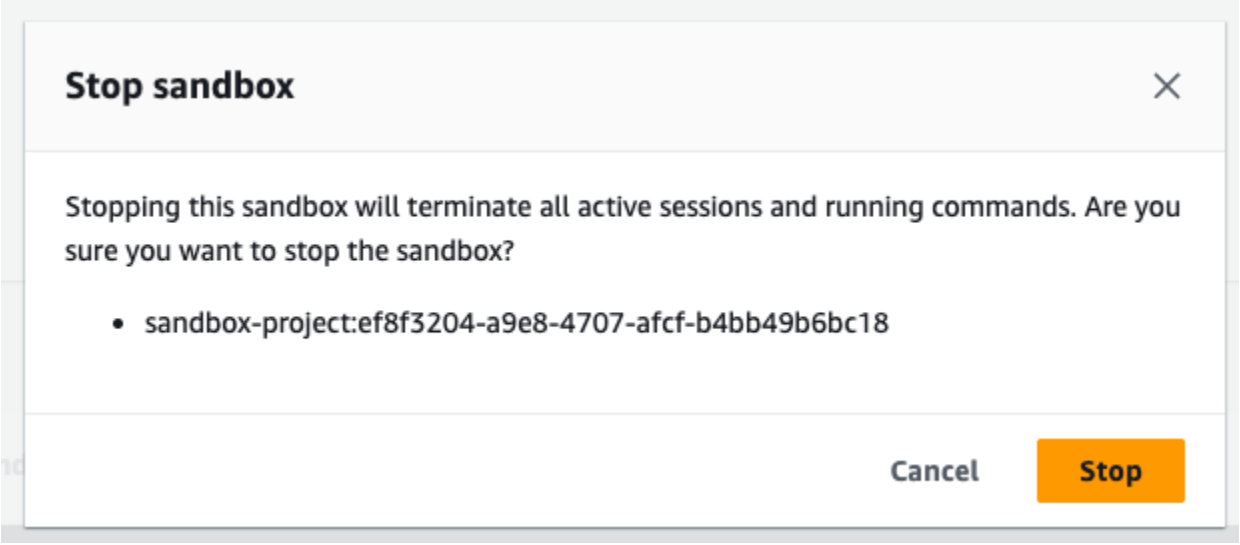
3. En la pestaña Ejecutar comando, introduzca comandos personalizados y, a continuación, elija Ejecutar comando.

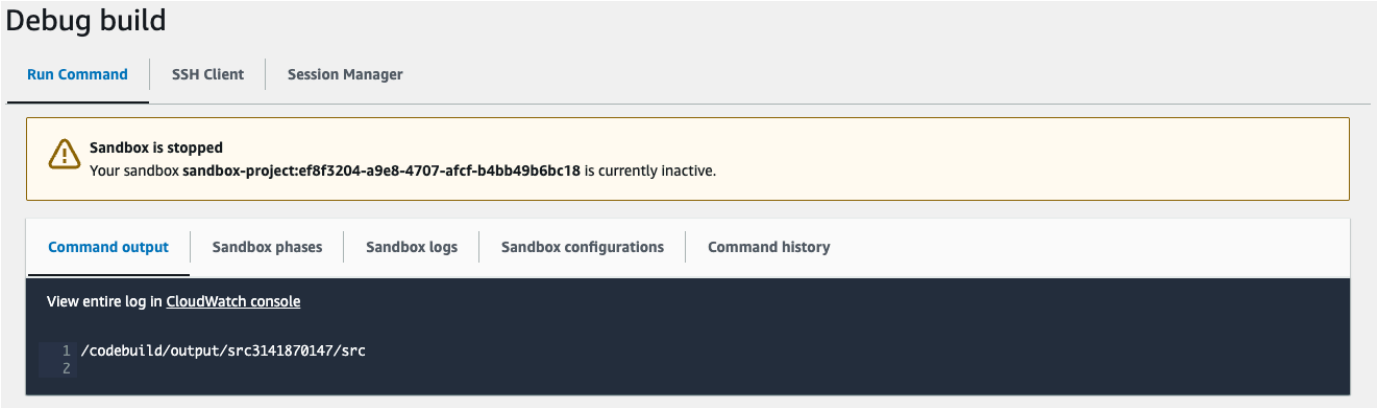


4. A continuación, su CodeBuild sandbox se inicializará y comenzará a ejecutar sus comandos personalizados. El resultado se mostrará en la pestaña Salida cuando se haya completado.



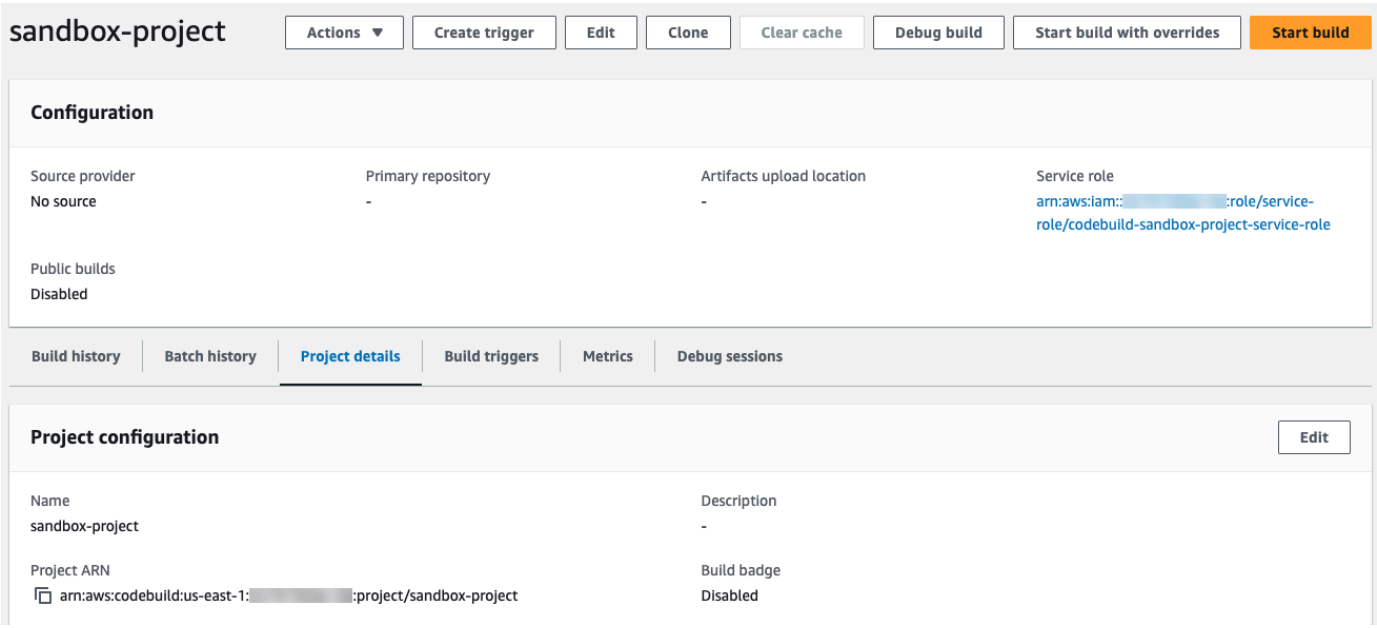
5. Cuando se haya completado la solución de problemas, puede detener el entorno de pruebas seleccionando Detener entorno de pruebas. A continuación, seleccione Detener para confirmar que el entorno de pruebas se detendrá.



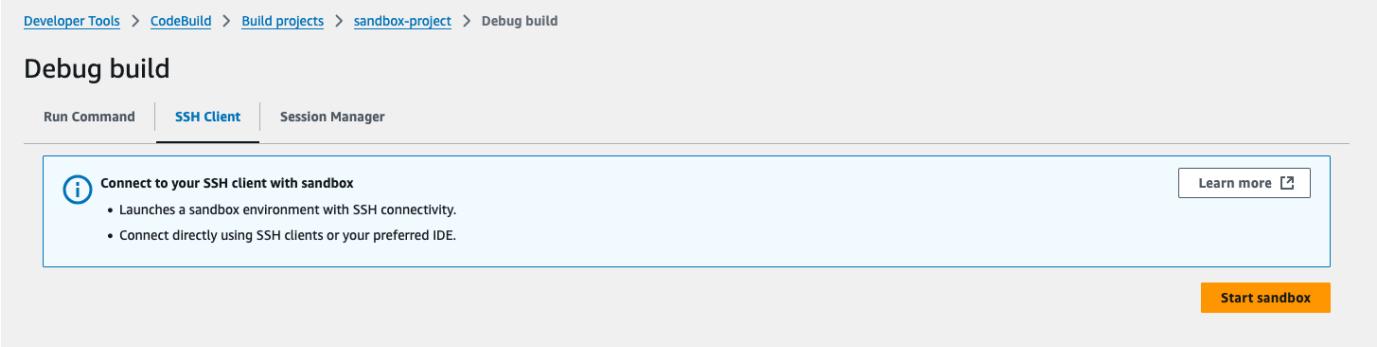


Conéctese a su cliente SSH con CodeBuild sandbox (consola)

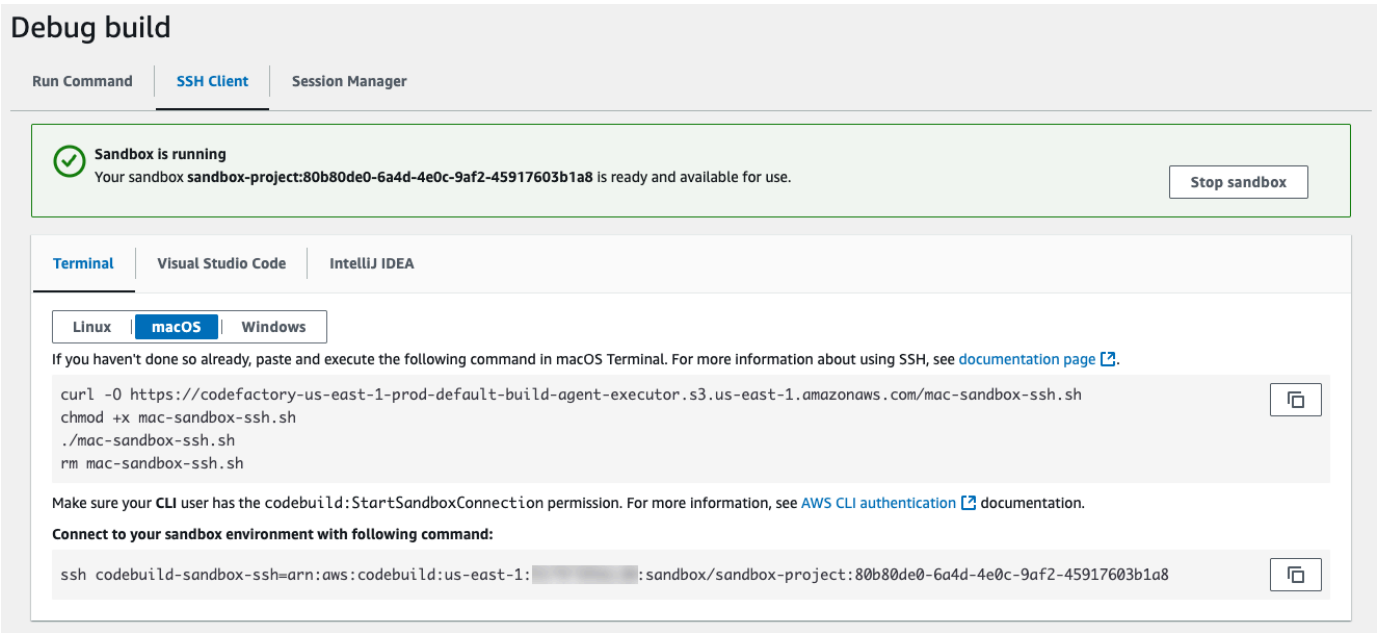
1. Abre la AWS CodeBuild consola en. https://console.aws.amazon.com/codesuite/codebuild/home
2. En el panel de navegación, elija Proyectos de compilación. Elija el proyecto de compilación y, a continuación, elija Iniciar depuración.



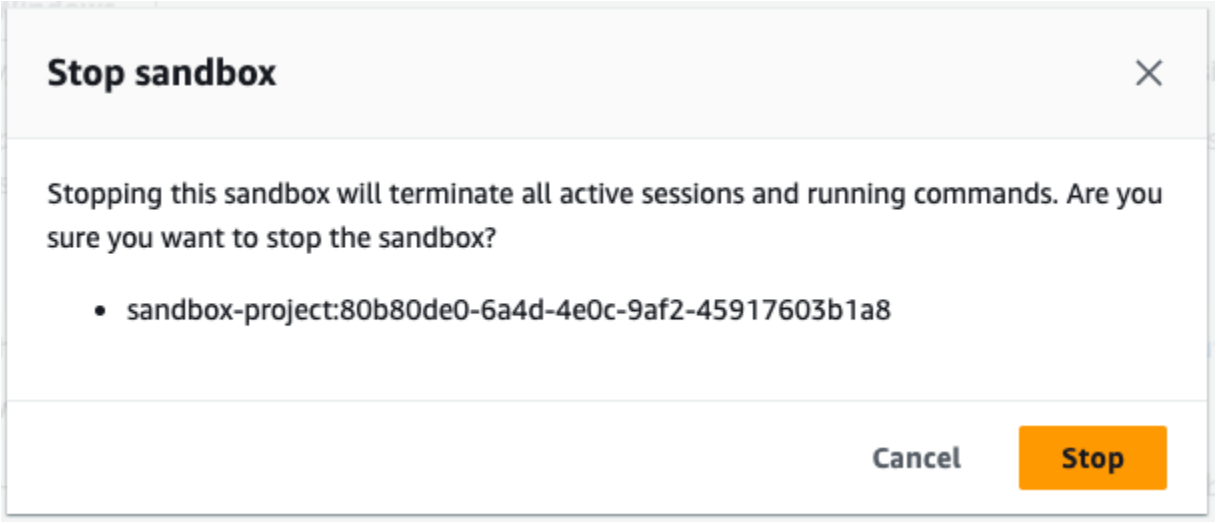
3. En la pestaña Cliente de SSH, seleccione Iniciar entorno de pruebas.

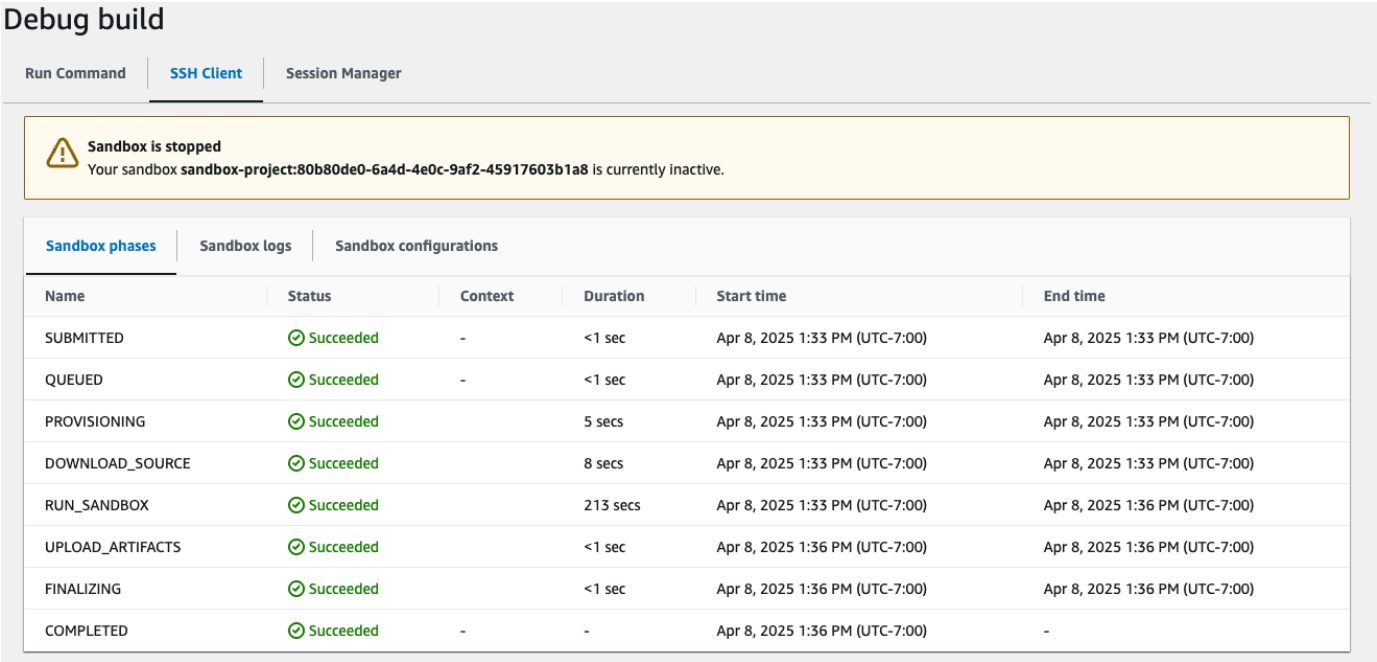


4. Cuando comience a funcionar el CodeBuild sandbox, sigue las instrucciones de la consola para conectar tu cliente SSH al sandbox.



5. Cuando se haya completado la solución de problemas, puede detener el entorno de pruebas seleccionando Detener entorno de pruebas. A continuación, seleccione Detener para confirmar que el entorno de pruebas se detendrá.





Depure compilaciones con sandbox (CodeBuild AWS CLI)

Usa las siguientes instrucciones para ejecutar comandos y conectar tu cliente SSH con sandbox. CodeBuild

Inicie un sandbox CodeBuild (AWS CLI)

CLI command

```
aws codebuild start-sandbox --project-name $PROJECT_NAME
```

- `--project-name`: nombre CodeBuild del proyecto

Sample request

```
aws codebuild start-sandbox --project-name "project-name"
```

Sample response

```
{
    "id": "project-name",
    "arn": "arn:aws:codebuild:us-west-2:962803963624:sandbox/project-name",
    "projectName": "project-name",
    "requestTime": "2025-02-06T11:24:15.560000-08:00",
    "status": "QUEUED",
    "source": {
        "type": "S3",
        "location": "arn:aws:s3:::cofa-e2e-test-1-us-west-2-beta-default-build-
sources/eb-sample-jetty-v4.zip",
        "insecureSsl": false
    },
    "environment": {
        "type": "LINUX_CONTAINER",
        "image": "aws/codebuild/standard:6.0",
        "computeType": "BUILD_GENERAL1_SMALL",
        "environmentVariables": [{
                "name": "foo",
                "value": "bar",
                "type": "PLAINTEXT"
            },
            {
                "name": "bar",
                "value": "baz",
                "type": "PLAINTEXT"
            }
        ],
        "privilegedMode": false,
        "imagePullCredentialsType": "CODEBUILD"
    },
    "timeoutInMinutes": 10,
    "queuedTimeoutInMinutes": 480,
    "logConfig": {
        "cloudWatchLogs": {
```

```
            "status": "ENABLED",
            "groupName": "group",
            "streamName": "stream"
        },
        "s3Logs": {
            "status": "ENABLED",
            "location": "codefactory-test-pool-1-us-west-2-beta-default-build-logs",
            "encryptionDisabled": false
        }
    },
    "encryptionKey": "arn:aws:kms:us-west-2:962803963624:alias/SampleEncryptionKey",
    "serviceRole": "arn:aws:iam::962803963624:role/BuildExecutionServiceRole",
    "currentSession": {
        "id": "0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
        "currentPhase": "QUEUED",
        "status": "QUEUED",
        "startTime": "2025-02-06T11:24:15.626000-08:00",
        "logs": {
            "groupName": "group",
            "streamName": "stream/0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
            "deepLink": "https://console.aws.amazon.com/cloudwatch/home?
region=us-west-2#logsV2:log-groups/log-group/group/log-events/stream
$252F0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
            "s3DeepLink": "https://s3.console.aws.amazon.com/s3/object/
codefactory-test-pool-1-us-west-2-beta-default-build-logs/0103e0e7-52aa-4a3d-81dd-
bfc27226fa54.gz?region=us-west-2",
            "cloudWatchLogsArn": "arn:aws:logs:us-west-2:962803963624:log-
group:group:log-stream:stream/0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
            "s3LogsArn": "arn:aws:s3:::codefactory-test-pool-1-us-west-2-beta-
default-build-logs/0103e0e7-52aa-4a3d-81dd-bfc27226fa54.gz",
            "cloudWatchLogs": {
                "status": "ENABLED",
                "groupName": "group",
                "streamName": "stream"
            },
            "s3Logs": {
                "status": "ENABLED",
                "location": "codefactory-test-pool-1-us-west-2-beta-default-build-
logs",
                "encryptionDisabled": false
            }
        }
    }
```

```
}
```

Obtenga información sobre el estado del sandbox (AWS CLI)

CLI command

```
aws codebuild batch-get-sandboxes --ids $SANDBOX_IDs
```

Sample request

```
aws codebuild stop-sandbox --id "arn:aws:codebuild:us-west-2:962803963624:sandbox/
project-name"
```

- `--ids`: lista separada por comas de `sandboxIds` o `sandboxArns`.

  Puede proporcionar un ID de entorno de pruebas o un ARN de entorno de pruebas:

- ID de entorno de pruebas: *<codebuild-project-name>*:*<UUID>*

  Por ejemplo, `project-name:d25be134-05cb-404a-85da-ac5f85d2d72c`.

- Sandbox ARN: arn:aws:codebuild: ::sandbox/: *<region>* *<account-id>* *<codebuild-project-name>* *<UUID>*

  Por ejemplo, `arn:aws:codebuild:us-west-2:962803963624:sandbox/project-name:d25be134-05cb-404a-85da-ac5f85d2d72c`.

Sample response

```
{
    "sandboxes": [{
        "id": "project-name",
        "arn": "arn:aws:codebuild:us-west-2:962803963624:sandbox/project-name",
        "projectName": "project-name",
        "requestTime": "2025-02-06T11:24:15.560000-08:00",
        "endTime": "2025-02-06T11:39:21.587000-08:00",
        "status": "STOPPED",
        "source": {
            "type": "S3",
```

```
            "location": "arn:aws:s3:::cofa-e2e-test-1-us-west-2-beta-default-build-
sources/eb-sample-jetty-v4.zip",
            "insecureSsl": false
        },
        "environment": {
            "type": "LINUX_CONTAINER",
            "image": "aws/codebuild/standard:6.0",
            "computeType": "BUILD_GENERAL1_SMALL",
            "environmentVariables": [{
                    "name": "foo",
                    "value": "bar",
                    "type": "PLAINTEXT"
                },
                {
                    "name": "bar",
                    "value": "baz",
                    "type": "PLAINTEXT"
                }
            ],
            "privilegedMode": false,
            "imagePullCredentialsType": "CODEBUILD"
        },
        "timeoutInMinutes": 10,
        "queuedTimeoutInMinutes": 480,
        "logConfig": {
            "cloudWatchLogs": {
                "status": "ENABLED",
                "groupName": "group",
                "streamName": "stream"
            },
            "s3Logs": {
                "status": "ENABLED",
                "location": "codefactory-test-pool-1-us-west-2-beta-default-build-
logs",
                "encryptionDisabled": false
            }
        },
        "encryptionKey": "arn:aws:kms:us-west-2:962803963624:alias/
SampleEncryptionKey",
        "serviceRole": "arn:aws:iam::962803963624:role/BuildExecutionServiceRole",
        "currentSession": {
            "id": "0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
            "currentPhase": "COMPLETED",
            "status": "STOPPED",
```

```
            "startTime": "2025-02-06T11:24:15.626000-08:00",
            "endTime": "2025-02-06T11:39:21.600000-08:00",
            "phases": [{
                    "phaseType": "SUBMITTED",
                    "phaseStatus": "SUCCEEDED",
                    "startTime": "2025-02-06T11:24:15.577000-08:00",
                    "endTime": "2025-02-06T11:24:15.606000-08:00",
                    "durationInSeconds": 0
                },
                {
                    "phaseType": "QUEUED",
                    "phaseStatus": "SUCCEEDED",
                    "startTime": "2025-02-06T11:24:15.606000-08:00",
                    "endTime": "2025-02-06T11:24:16.067000-08:00",
                    "durationInSeconds": 0
                },
                {
                    "phaseType": "PROVISIONING",
                    "phaseStatus": "SUCCEEDED",
                    "startTime": "2025-02-06T11:24:16.067000-08:00",
                    "endTime": "2025-02-06T11:24:20.519000-08:00",
                    "durationInSeconds": 4,
                    "contexts": [{
                        "statusCode": "",
                        "message": ""
                    }]
                },
                {
                    "phaseType": "DOWNLOAD_SOURCE",
                    "phaseStatus": "SUCCEEDED",
                    "startTime": "2025-02-06T11:24:20.519000-08:00",
                    "endTime": "2025-02-06T11:24:22.238000-08:00",
                    "durationInSeconds": 1,
                    "contexts": [{
                        "statusCode": "",
                        "message": ""
                    }]
                },
                {
                    "phaseType": "RUNNING_SANDBOX",
                    "phaseStatus": "TIMED_OUT",
                    "startTime": "2025-02-06T11:24:22.238000-08:00",
                    "endTime": "2025-02-06T11:39:21.560000-08:00",
                    "durationInSeconds": 899,
```

```
                    "contexts": [{
                        "statusCode": "BUILD_TIMED_OUT",
                        "message": "Build has timed out. "
                    }]
                },
                {
                    "phaseType": "COMPLETED",
                    "startTime": "2025-02-06T11:39:21.560000-08:00"
                }
            ],
            "logs": {
                "groupName": "group",
                "streamName": "stream/0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
                "deepLink": "https://console.aws.amazon.com/cloudwatch/
home?region=us-west-2#logsV2:log-groups/log-group/group/log-events/stream
$252F0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
                "s3DeepLink": "https://s3.console.aws.amazon.com/s3/object/
codefactory-test-pool-1-us-west-2-beta-default-build-logs/0103e0e7-52aa-4a3d-81dd-
bfc27226fa54.gz?region=us-west-2",
                "cloudWatchLogsArn": "arn:aws:logs:us-west-2:962803963624:log-
group:group:log-stream:stream/0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
                "s3LogsArn": "arn:aws:s3:::codefactory-test-pool-1-us-west-2-beta-
default-build-logs/0103e0e7-52aa-4a3d-81dd-bfc27226fa54.gz",
                "cloudWatchLogs": {
                    "status": "ENABLED",
                    "groupName": "group",
                    "streamName": "stream"
                },
                "s3Logs": {
                    "status": "ENABLED",
                    "location": "codefactory-test-pool-1-us-west-2-beta-default-
build-logs",
                    "encryptionDisabled": false
                }
            }
        }
    }],
    "sandboxesNotFound": []
}
```

Detener un sandbox (AWS CLI)

CLI command

```
aws codebuild stop-sandbox --id $SANDBOX-ID
```

- `--id`: `sandboxId` o `sandboxArn`.

Sample request

```
aws codebuild stop-sandbox --id "arn:aws:codebuild:us-west-2:962803963624:sandbox/
project-name"
```

Sample response

```
{
    "id": "project-name",
    "arn": "arn:aws:codebuild:us-west-2:962803963624:sandbox/project-name",
    "projectName": "project-name",
    "requestTime": "2025-02-06T11:24:15.560000-08:00",
    "status": "STOPPING",
    "source": {
        "type": "S3",
        "location": "arn:aws:s3:::cofa-e2e-test-1-us-west-2-beta-default-build-
sources/eb-sample-jetty-v4.zip",
        "insecureSsl": false
    },
    "environment": {
        "type": "LINUX_CONTAINER",
        "image": "aws/codebuild/standard:6.0",
        "computeType": "BUILD_GENERAL1_SMALL",
        "environmentVariables": [{
                "name": "foo",
                "value": "bar",
                "type": "PLAINTEXT"
            },
            {
                "name": "bar",
                "value": "baz",
                "type": "PLAINTEXT"
            }
        ],
```

```
        "privilegedMode": false,
        "imagePullCredentialsType": "CODEBUILD"
    },
    "timeoutInMinutes": 10,
    "queuedTimeoutInMinutes": 480,
    "logConfig": {
        "cloudWatchLogs": {
            "status": "ENABLED",
            "groupName": "group",
            "streamName": "stream"
        },
        "s3Logs": {
            "status": "ENABLED",
            "location": "codefactory-test-pool-1-us-west-2-beta-default-build-logs",
            "encryptionDisabled": false
        }
    },
    "encryptionKey": "arn:aws:kms:us-west-2:962803963624:alias/SampleEncryptionKey",
    "serviceRole": "arn:aws:iam::962803963624:role/BuildExecutionServiceRole",
    "currentSession": {
        "id": "0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
        "currentPhase": "RUN_SANDBOX",
        "status": "STOPPING",
        "startTime": "2025-02-06T11:24:15.626000-08:00",
        "phases": [{
                "phaseType": "SUBMITTED",
                "phaseStatus": "SUCCEEDED",
                "startTime": "2025-02-08T14:33:26.144000-08:00",
                "endTime": "2025-02-08T14:33:26.173000-08:00",
                "durationInSeconds": 0
            },
            {
                "phaseType": "QUEUED",
                "phaseStatus": "SUCCEEDED",
                "startTime": "2025-02-08T14:33:26.173000-08:00",
                "endTime": "2025-02-08T14:33:26.702000-08:00",
                "durationInSeconds": 0
            },
            {
                "phaseType": "PROVISIONING",
                "phaseStatus": "SUCCEEDED",
                "startTime": "2025-02-08T14:33:26.702000-08:00",
                "endTime": "2025-02-08T14:33:30.530000-08:00",
                "durationInSeconds": 3,
```

```
                "contexts": [{
                    "statusCode": "",
                    "message": ""
                }]
            },
            {
                "phaseType": "DOWNLOAD_SOURCE",
                "phaseStatus": "SUCCEEDED",
                "startTime": "2025-02-08T14:33:30.530000-08:00",
                "endTime": "2025-02-08T14:33:33.478000-08:00",
                "durationInSeconds": 2,
                "contexts": [{
                    "statusCode": "",
                    "message": ""
                }]
            },
            {
                "phaseType": "RUN_SANDBOX",
                "startTime": "2025-02-08T14:33:33.478000-08:00"
            }
        ],
        "logs": {
            "groupName": "group",
            "streamName": "stream/0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
            "deepLink": "https://console.aws.amazon.com/cloudwatch/home?
region=us-west-2#logsV2:log-groups/log-group/group/log-events/stream
$252F0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
            "s3DeepLink": "https://s3.console.aws.amazon.com/s3/object/
codefactory-test-pool-1-us-west-2-beta-default-build-logs/0103e0e7-52aa-4a3d-81dd-
bfc27226fa54.gz?region=us-west-2",
            "cloudWatchLogsArn": "arn:aws:logs:us-west-2:962803963624:log-
group:group:log-stream:stream/0103e0e7-52aa-4a3d-81dd-bfc27226fa54",
            "s3LogsArn": "arn:aws:s3:::codefactory-test-pool-1-us-west-2-beta-
default-build-logs/0103e0e7-52aa-4a3d-81dd-bfc27226fa54.gz",
            "cloudWatchLogs": {
                "status": "ENABLED",
                "groupName": "group",
                "streamName": "stream"
            },
            "s3Logs": {
                "status": "ENABLED",
                "location": "codefactory-test-pool-1-us-west-2-beta-default-build-
logs",
                "encryptionDisabled": false
```

```
            }
        }
    }
}
```

Iniciar la ejecución de un comando (AWS CLI)

CLI command

```
aws codebuild start-command-execution --command $COMMAND --type $TYPE --sandbox-id
 $SANDBOX-ID
```

- `--command`: comando que debe ejecutarse.
- `--sandbox-id`: `sandboxId` o `sandboxArn`.
- `--type`: tipo de comando, `SHELL`.

Sample request

```
aws codebuild start-command-execution --command "echo "Hello World"" --type SHELL --
sandbox-id "arn:aws:codebuild:us-west-2:962803963624:sandbox/project-name
```

Sample response

```
{
    "id": "e1c658c2-02bb-42a8-9abb-94835241fcd6",
    "sandboxId": "f7126a4a-b0d5-452f-814c-fea73718f805",
    "submitTime": "2025-02-06T20:12:02.683000-08:00",
    "status": "SUBMITTED",
    "command": "echo \"Hello World\"",
    "type": "SHELL",
    "logs": {
        "groupName": "group",
        "streamName": "stream",
        "deepLink": "https://console.aws.amazon.com/cloudwatch/home?region=us-
west-2#logsV2:log-groups/log-group/group/log-events/stream",
        "s3DeepLink": "https://s3.console.aws.amazon.com/s3/object/codefactory-test-
pool-1-us-west-2-beta-default-build-logs/f7126a4a-b0d5-452f-814c-fea73718f805.gz?
region=us-west-2",
        "cloudWatchLogsArn": "arn:aws:logs:us-west-2:962803963624:log-
group:group:log-stream:stream",
```

```
        "s3LogsArn": "arn:aws:s3:::codefactory-test-pool-1-us-west-2-beta-default-
build-logs/f7126a4a-b0d5-452f-814c-fea73718f805.gz",
        "cloudWatchLogs": {
            "status": "ENABLED",
            "groupName": "group",
            "streamName": "stream"
        },
        "s3Logs": {
            "status": "ENABLED",
            "location": "codefactory-test-pool-1-us-west-2-beta-default-build-logs",
            "encryptionDisabled": false
        }
    }
}
```

Obtenga información sobre la ejecución de los comandos (AWS CLI)

CLI command

```
aws codebuild batch-get-command-executions --command-execution-ids $COMMAND-IDs --
sandbox-id $SANDBOX-IDs
```

- `--command-execution-ids`: lista separada por comas de `commandExecutionIds`.
- `--sandbox-id`: `sandboxId` o `sandboxArn`.

Sample request

```
aws codebuild batch-get-command-executions --command-execution-
ids"c3c085ed-5a8f-4531-8e95-87d547f27ffd" --sandbox-id "arn:aws:codebuild:us-
west-2:962803963624:sandbox/project-name"
```

Sample response

```
{
    "commandExecutions": [{
        "id": "c3c085ed-5a8f-4531-8e95-87d547f27ffd",
        "sandboxId": "cd71e456-2a4c-4db4-ada5-da892b0bba05",
        "submitTime": "2025-02-10T20:18:17.118000-08:00",
        "startTime": "2025-02-10T20:18:17.939000-08:00",
        "endTime": "2025-02-10T20:18:17.976000-08:00",
```

```
        "status": "SUCCEEDED",
        "command": "echo \"Hello World\"",
        "type": "SHELL",
        "exitCode": "0",
        "standardOutputContent": "Hello World\n",
        "logs": {
            "groupName": "group",
            "streamName": "stream",
            "deepLink": "https://console.aws.amazon.com/cloudwatch/home?region=us-
west-2#logsV2:log-groups/log-group/group/log-events/stream",
            "s3DeepLink": "https://s3.console.aws.amazon.com/s3/object/
codefactory-test-pool-1-us-west-2-beta-default-build-logs/cd71e456-2a4c-4db4-ada5-
da892b0bba05.gz?region=us-west-2",
            "cloudWatchLogsArn": "arn:aws:logs:us-west-2:962803963624:log-
group:group:log-stream:stream",
            "s3LogsArn": "arn:aws:s3:::codefactory-test-pool-1-us-west-2-beta-
default-build-logs/cd71e456-2a4c-4db4-ada5-da892b0bba05.gz",
            "cloudWatchLogs": {
                "status": "ENABLED",
                "groupName": "group",
                "streamName": "stream"
            },
            "s3Logs": {
                "status": "ENABLED",
                "location": "codefactory-test-pool-1-us-west-2-beta-default-build-
logs",
                "encryptionDisabled": false
            }
        }
    }],
    "commandExecutionsNotFound": []
}
```

Enumera las ejecuciones de comandos en un entorno limitado (AWS CLI)

CLI command

```
aws codebuild list-command-executions-for-sandbox --sandbox-id $SANDBOX-ID --next-
token $NEXT_TOKEN --max-results $MAX_RESULTS --sort-order $SORT_ORDER
```

- `--next-token`: siguiente token, si lo hay, para obtener resultados paginados. Obtendrá este valor de la ejecución anterior de entornos de pruebas de la lista.

- `--max-results`: (opcional) número máximo de registros del entorno de pruebas que se recuperará.
- `--sort-order`: orden en el que se deben recuperar los registros del entorno de pruebas.

Sample request

```
aws codebuild list-command-executions-for-sandbox --sandbox-id
 "arn:aws:codebuild:us-west-2:962803963624:sandbox/project-name"
```

Sample response

```
{
    "commandExecutions": [{
            "id": "aad6687e-07bc-45ab-a1fd-f5440229b528",
            "sandboxId": "cd71e456-2a4c-4db4-ada5-da892b0bba05",
            "submitTime": "2025-02-10T20:18:35.304000-08:00",
            "startTime": "2025-02-10T20:18:35.615000-08:00",
            "endTime": "2025-02-10T20:18:35.651000-08:00",
            "status": "FAILED",
            "command": "fail command",
            "type": "SHELL",
            "exitCode": "127",
            "standardErrContent": "/codebuild/output/tmp/script.sh: 4: fail: not
 found\n",
            "logs": {
                "groupName": "group",
                "streamName": "stream",
                "deepLink": "https://console.aws.amazon.com/cloudwatch/home?
region=us-west-2#logsV2:log-groups/log-group/group/log-events/stream",
                "s3DeepLink": "https://s3.console.aws.amazon.com/s3/object/
codefactory-test-pool-1-us-west-2-beta-default-build-logs/cd71e456-2a4c-4db4-ada5-
da892b0bba05.gz?region=us-west-2",
                "cloudWatchLogsArn": "arn:aws:logs:us-west-2:962803963624:log-
group:group:log-stream:stream",
                "s3LogsArn": "arn:aws:s3:::codefactory-test-pool-1-us-west-2-beta-
default-build-logs/cd71e456-2a4c-4db4-ada5-da892b0bba05.gz",
                "cloudWatchLogs": {
                    "status": "ENABLED",
                    "groupName": "group",
                    "streamName": "stream"
                },
                "s3Logs": {
```

```
                    "status": "ENABLED",
                    "location": "codefactory-test-pool-1-us-west-2-beta-default-
build-logs",
                    "encryptionDisabled": false
                }
            }
        },
        {
            "id": "c3c085ed-5a8f-4531-8e95-87d547f27ffd",
            "sandboxId": "cd71e456-2a4c-4db4-ada5-da892b0bba05",
            "submitTime": "2025-02-10T20:18:17.118000-08:00",
            "startTime": "2025-02-10T20:18:17.939000-08:00",
            "endTime": "2025-02-10T20:18:17.976000-08:00",
            "status": "SUCCEEDED",
            "command": "echo \"Hello World\"",
            "type": "SHELL",
            "exitCode": "0",
            "standardOutputContent": "Hello World\n",
            "logs": {
                "groupName": "group",
                "streamName": "stream",
                "deepLink": "https://console.aws.amazon.com/cloudwatch/home?
region=us-west-2#logsV2:log-groups/log-group/group/log-events/stream",
                "s3DeepLink": "https://s3.console.aws.amazon.com/s3/object/
codefactory-test-pool-1-us-west-2-beta-default-build-logs/cd71e456-2a4c-4db4-ada5-
da892b0bba05.gz?region=us-west-2",
                "cloudWatchLogsArn": "arn:aws:logs:us-west-2:962803963624:log-
group:group:log-stream:stream",
                "s3LogsArn": "arn:aws:s3:::codefactory-test-pool-1-us-west-2-beta-
default-build-logs/cd71e456-2a4c-4db4-ada5-da892b0bba05.gz",
                "cloudWatchLogs": {
                    "status": "ENABLED",
                    "groupName": "group",
                    "streamName": "stream"
                },
                "s3Logs": {
                    "status": "ENABLED",
                    "location": "codefactory-test-pool-1-us-west-2-beta-default-
build-logs",
                    "encryptionDisabled": false
                }
            }
        }
    ]
```

```
}
```

Enumera los entornos limitados (AWS CLI)

CLI command

```
aws codebuild list-sandboxes --next-token $NEXT_TOKEN --max-results $MAX_RESULTS --
sort-order $SORT_ORDER
```

Sample request

```
aws codebuild list-sandboxes
```

Sample response

```
{
    "ids": [
        "s3-log-project-integ-test-temp1739250628l4985d64e0f-7880-41df-9a3c-
fb6597a266d2:827a5243-0841-4b69-a720-4438796f6967",
        "s3-log-project-integ-test-temp1739249999716bbd438dd-8bb8-47bd-
ba6b-0133ac65b3d3:e2fa4eab-73af-42e3-8903-92fddaf9f378",
        "s3-log-project-integ-test-
temp17392474779450fbdacc2-2d6e-4190-9ad5-28f891bb7415:cd71e456-2a4c-4db4-ada5-
da892b0bba05",
        "s3-log-project-integ-test-temp1739224628416430142lc-5030-4fa1-b4d3-
ca15e44771c5:9e26ab3f-65e4-4896-a19c-56b1a95e630a",
        "s3-log-project-integ-test-temp17392136731949705 6d8d-6d8e-4f5a-a37c-
a62f5686731f:22d91b06-df1e-4e9c-a664-c0abb8d5920b",
        "s3-log-project-integ-test-temp1739213439503f6283f19-390c-4dc8-95a9-
c8480113384a:82cc413e-fc46-47ab-898f-ae23c83a613f",
        "s3-log-project-integ-test-temp1739054385570b1f1ddc2-0a23-4062-
bd0c-24e9e4a99b99:c02562f3-2396-42ec-98da-38e3fe5da13a",
        "s3-log-project-integ-test-temp173905400540237dab1ac-1fde-4dfb-a8f5-
c0114333dc89:d2f30493-f65e-4fa0-a7b6-08a5e77497b9",
        "s3-log-project-integ-test-
temp17390534055719c534090-7bc4-48f1-92c5-34acaec5bf1e:df5f1c8a-f017-43b7-91ba-
ad2619e2c059",
        "s3-log-project-integ-test-temp1739052719086a61813cc-
ebb9-4db4-9391-7f43cc984ee4:d61917ec-8037-4647-8d52-060349272c4a",
        "s3-log-project-integ-test-temp173898670094078b67edb-
c42f-42ed-9db2-4b5c1a5fc66a:ce33dfbc-beeb-4466-8c99-a3734a0392c7",
```

```
        "s3-log-project-integ-test-
temp17389863425584d21b7cd-32e2-4f11-9175-72c89ecaffef:046dadf0-1f3a-4d51-a2c0-
e88361924acf",
        "s3-log-project-integ-test-
temp1738985884273977ccd23-394b-46cc-90d3-7ab94cf764dc:0370dc41-9339-4b0a-91ed-51929761b244",
        "s3-log-project-integ-test-temp1738985365972241b614f-8e41-4387-
bd25-2b8351fbc9e0:076c392a-9630-47d8-85a9-116aa34edfff",
        "s3-log-project-integ-test-
temp1738985043988a51a9e2b-09d6-4d24-9c3c-1e6e21ac9fa8:6ea3949c-435b-4177-
aa4d-614d5956244c",
        "s3-log-project-integ-test-temp1738984123354c68b31ad-49d1-4f4b-981d-
b66c00565ff6:6c3fff6c-815b-48b5-ada3-737400a6dee8",
        "s3-log-project-integ-test-
temp1738977263715d4d5bf6c-370a-48bf-8ea6-905358a6cf92:968a0f54-724a-42d1-9207-6ed854b2fae8",
        "s3-log-project-integ-test-
temp173897358796816ce8d7d-2a5e-41ef-855b-4a94a8d2795d:80f9a7ce-930a-402e-934e-
d8b511d68b04",
        "s3-log-project-integ-test-temp17389730633301af5e452-0966-467c-
b684-4e36d47f568c:cabbe989-2e8a-473c-af25-32edc8c28646",
        "s3-log-project-integ-test-temp1738901503813173fd468-
b723-4d7b-9f9f-82e88d17f264:f7126a4a-b0d5-452f-814c-fea73718f805",
        "s3-log-project-integ-test-temp1738890502472c13616fb-
bd0f-4253-86cc-28b74c97a0ba:c6f197e5-3a53-45b6-863e-0e6353375437",
        "s3-log-project-integ-test-
temp1738890304468361 0daf3-8da7-43c6-8580-9978432432ce:d20aa317-8838-4966-
bbfc-85b908213df1",
        "s3-log-project-integ-test-temp173888857196780b5ab8b-e54b-44fd-a222-
c5a374fffe96:ab4b9970-ffae-47a0-b3a8-7b6790008cad",
        "s3-log-project-integ-test-temp1738888336931c11d378d-e74d-49a4-
a723-3b92e6f7daac:4922f0e8-9b7d-4119-9c9f-115cd85e703e",
        "s3-log-project-integ-test-temp17388881717651612a397-c23f-4d88-
ba87-2773cd3fc0c9:be91c3fc-418e-4feb-8a3a-ba58ff8f4e8a",
        "s3-log-project-integ-test-
temp17388879727174c3c62ed-6195-4afb-8a03-59674d0e1187:a48826a8-3c0d-43c5-
a1b5-1c98a0f978e9",
        "s3-log-project-integ-test-temp1738885948597cef305e4-b8b4-46b0-a65b-
e2d0a7b83294:c050e77d-e3f8-4829-9a60-46149628fe96",
        "s3-log-project-integ-test-temp173888561463001a7d2a8-
e4e4-4434-94db-09d3da9a9e17:8c3ac3f5-7111-4297-aec9-2470d3ead873",
        "s3-log-project-integ-test-
temp1738869855076eb19cafd-04fe-41bd-8aa0-40826d0c0d27:d25be134-05cb-404a-85da-
ac5f85d2d72c",
        "s3-project-integ-test-temp1738868157467148eacfc-d39b-49fc-a137-
e55381cd2978:4909557b-c221-4814-b4b6-7d9e93d37c35",
```

```
        "s3-project-integ-test-temp1738820926895abec0af2-
e33d-473c-9cf4-2122dd9d6876:8f5cf218-71d6-40a4-a4be-6cacebd7765f",
        "s3-project-integ-test-temp1738819988775 74f969a6-1c2e-4441-b463-
ab175b45ce32:04396851-c901-4986-9117-585528e3877f",
        "s3-project-integ-test-temp17388189812309abd2604-29ba-4cf6-
b6bf-073207b7db9c:540075c7-f5ec-41e8-9341-2233c09247eb",
        "s3-project-integ-test-temp1738818843474d3ea9ac1-b609-461b-
bbdb-2da245c9bc96:865d4c3c-fbfe-4ece-9c92-d0c928341404",
        "s3-project-integ-test-temp1738818542236006e9169-e6d9-4344-9b59-
f557e7aec619:1f9ffa87-da15-4290-83e2-eebdd877497b",
        "s3-project-integ-test-
temp173881809557486ad11fd-7931-48d7-81d5-499cea52a6bc:c4c2efc4-685f-4e13-8b0f-1ef85ec300b1",
        "s3-project-integ-test-temp1738817941033229410 20-3f0b-49c3-b836-
fcd818ec9484:0344cfba-de48-456d-b2a8-6566bd4a5d6e",
        "s3-project-integ-test-temp1738817680747b93d0d0b-ea16-497f-9559-
af25ee6dcfdf:654a3a55-d92a-4dc6-8da8-56fd4d40d7e1",
        "s3-project-integ-test-temp1738817402719125 5c3da-086c-4270-b047-
acac0b7bee0d:b7e82740-2c69-42fc-ab5a-dbf15bc016a1",
        "s3-project-integ-test-temp1738817099799016e7fa3-b9b5-46a2-
bcd5-0888c646743f:8705a6a4-79ff-427a-a1c3-85c4e8fe462e",
        "s3-project-integ-test-temp1738816479281bb0c3606-5ebf-4623-
bed5-12b60e9d3512:f23fc74b-a981-4835-8e28-375fcd4c99e4",
        "s3-project-integ-test-
temp1738816263585c939a133-4d37-482c-9238-1dbff34b7674:ca28e234-0045-4ae6-8732-938b17597f50",
        "s3-project-integ-test-
temp173881580873072d18733-8fe4-43b1-83f7-95f25bb27ccf:c6f0f55b-5736-47c7-
a3aa-1b8461a6d5ed"
    ]
}
```

Tutorial: Conexión a un entorno de pruebas mediante SSH

En este tutorial se muestra cómo conectarse a un CodeBuild sandbox mediante un cliente SSH.

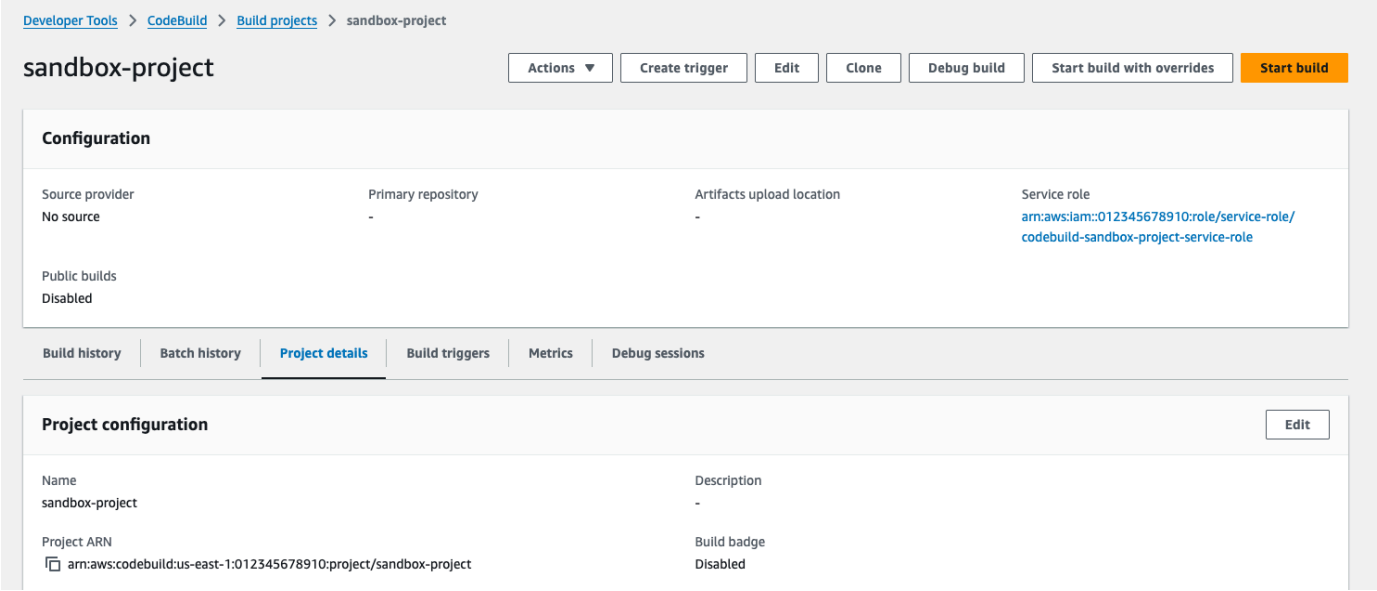
Para completar este tutorial, primero debe:

- Asegúrese de tener un proyecto existente AWS CodeBuild .
- Configure los permisos de IAM apropiados configurados para su rol en CodeBuild el proyecto.
- Instala y configura AWS CLI en tu máquina local.

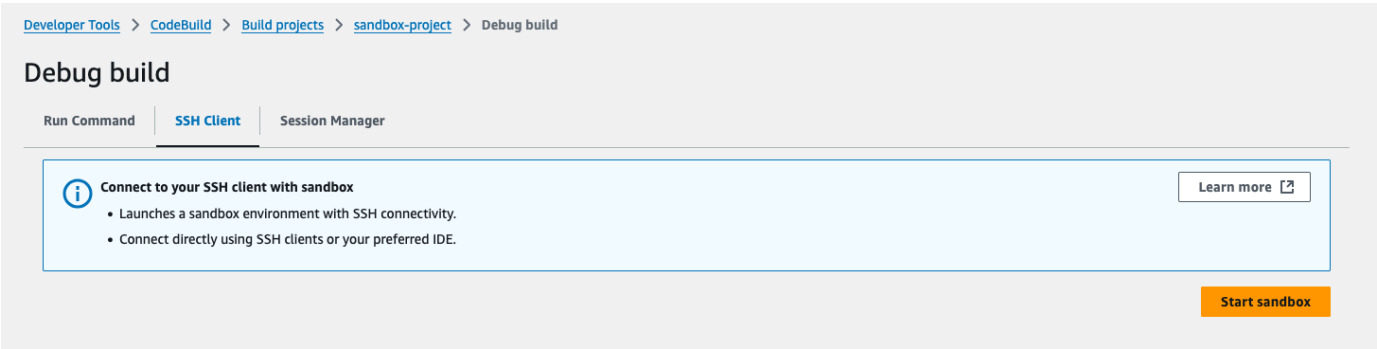
Paso 1: inicio de un entorno de pruebas

Para iniciar un CodeBuild sandbox en la consola

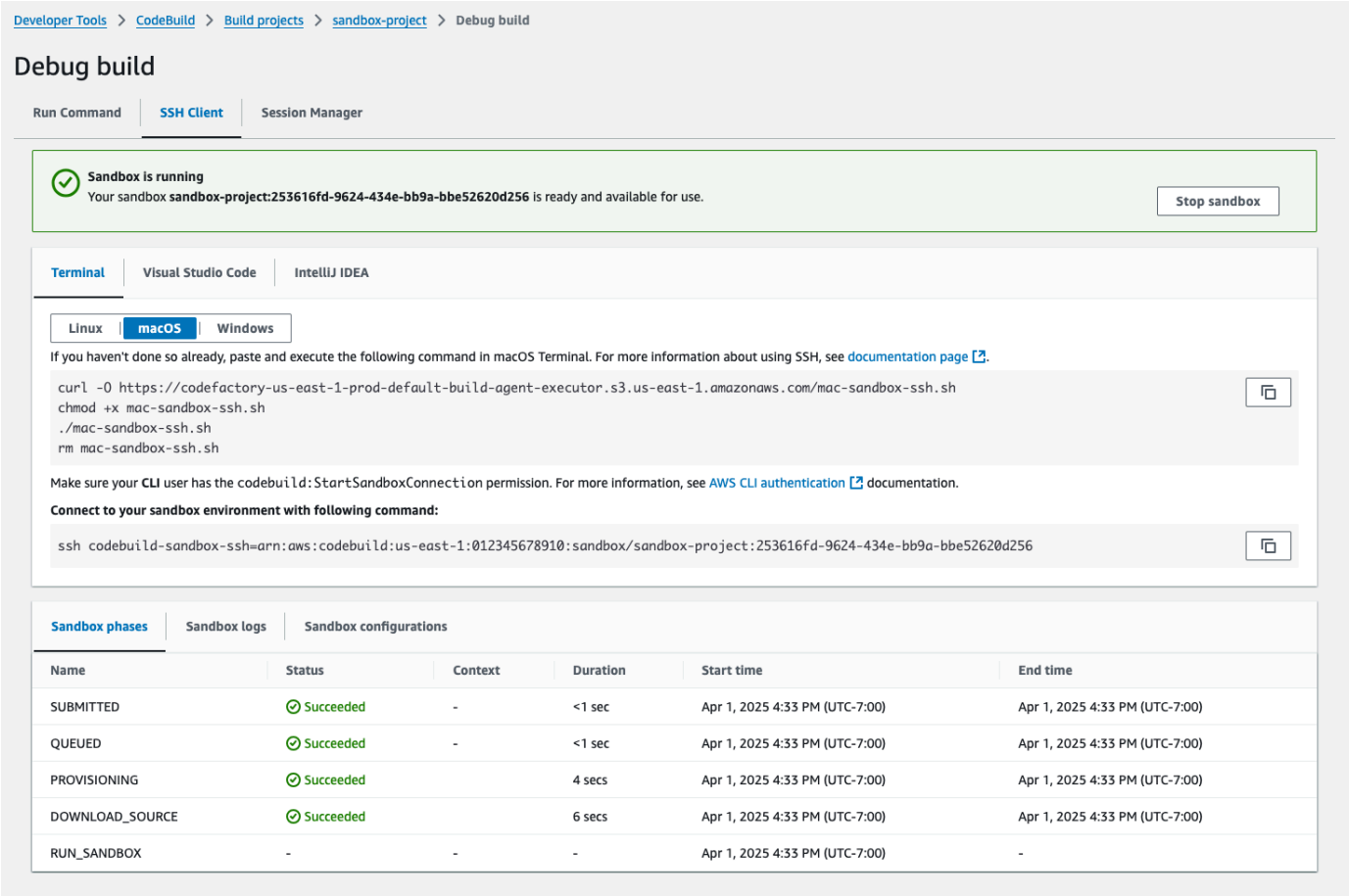
1. Abre la AWS CodeBuild consola en. https://console.aws.amazon.com/codesuite/codebuild/home
2. En el panel de navegación, elija Proyectos de compilación. Elija el proyecto de compilación y, a continuación, elija Iniciar depuración.



3. En la pestaña Cliente de SSH, seleccione Iniciar entorno de pruebas.



4. El proceso de inicialización del entorno de pruebas puede tardar un poco. Puede conectarse al entorno de pruebas cuando su estado cambie a `RUN_SANDDBOX`.

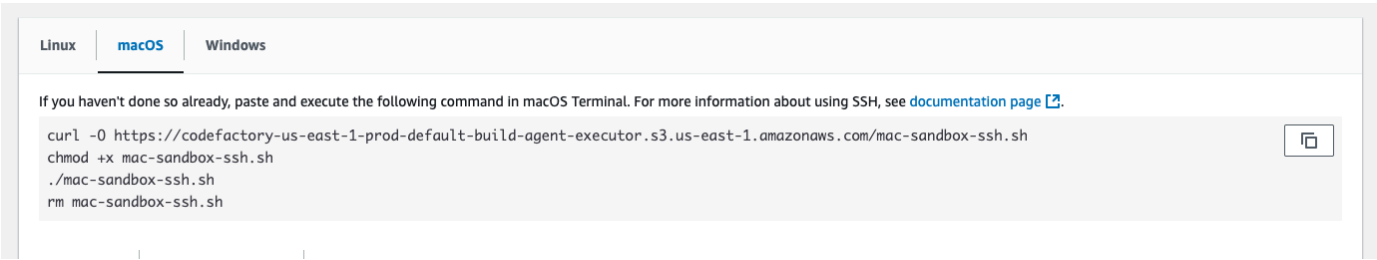


Paso 2: modificación de la configuración de SSH local

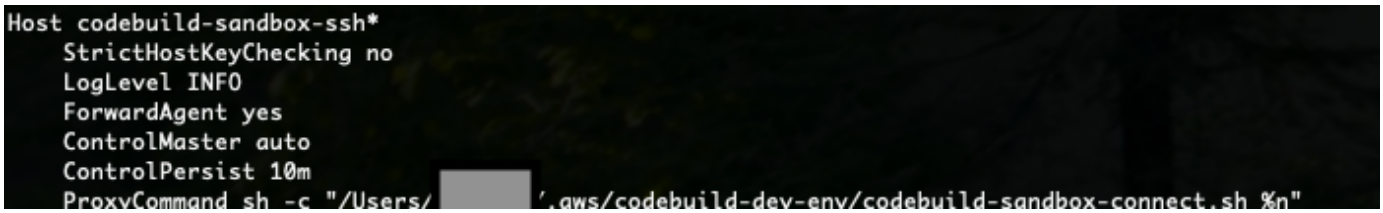
Si es la primera vez que se conecta a un entorno de pruebas, debe realizar un proceso de configuración único siguiendo estos pasos:

Cómo modificar la configuración de SSH local en la consola

1. Busque los comandos de configuración correspondientes a su sistema operativo.
2. Abra el terminal local y, a continuación, copie y ejecute los comandos proporcionados para descargar y ejecutar el script para configurar la configuración de SSH local. Por ejemplo, si el sistema operativo es macOS, utilice el siguiente comando:



3. El script de configuración agregará las configuraciones necesarias para conectarse a los entornos de pruebas. Se le pedirá que acepte estos cambios.
4. Tras la configuración correcta, se creará una nueva entrada de configuración SSH para el CodeBuild sandbox.



Paso 3: conexión con el entorno de pruebas

Cómo modificar la configuración de SSH local en la consola

1. Configure AWS CLI la autenticación y asegúrese de que su AWS CLI usuario tenga el `codebuild:StartSandboxConnection` permiso. Para obtener más información, consulte Autenticación con credenciales de usuario de IAM para la AWS CLI en la Guía del usuario de Interfaz de línea de comandos de AWS de la versión 1.
2. Conéctese al entorno de pruebas con el siguiente comando:

```
ssh codebuild-sandbox-ssh=arn:aws:codebuild:us-east-1:<account-id>:sandbox/<sandbox-id>
```

> **Note**
>
> Para solucionar problemas de conexión, use la marca `-v` para habilitar la salida detallada. Por ejemplo, `ssh -v codebuild-sandbox-ssh=arn:aws:codebuild:us-east-1:<account-id>:sandbox/<sandbox-id>`. Para obtener sugerencias adicionales para la solución de problemas, consulte Resolución de problemas AWS CodeBuild Problemas de conexión SSH en Sandbox.

Paso 4: Revisa tus resultados

Una vez conectado, puede depurar errores de compilación, probar comandos de compilación, experimentar con cambios de configuración y verificar variables de entorno y dependencias con el entorno de pruebas.

Resolución de problemas AWS CodeBuild Problemas de conexión SSH en Sandbox

Usa la información de este tema para ayudarte a identificar, diagnosticar y solucionar los problemas de conexión SSH en CodeBuild modo sandbox.

Temas

- StartSandboxConnectionInvalidInputExceptionerror al usar SSH en un entorno sandbox CodeBuild
- Error: «No se pueden encontrar las credenciales» cuando se conecta mediante SSH a CodeBuild un entorno sandbox
- StartSandboxConnectionAccessDeniedExceptionerror al usar SSH en un CodeBuild entorno sandbox
- Error: «ssh: no se pudo resolver el nombre de host» cuando se conectó mediante SSH a un entorno sandbox CodeBuild

StartSandboxConnectionInvalidInputExceptionerror al usar SSH en un entorno sandbox CodeBuild

Problema: al intentar conectarse a un entorno CodeBuild sandbox mediante el comando`ssh codebuild-sandbox-ssh=<sandbox-arn>`, es posible que aparezca un `InvalidInputException` error como el siguiente:

```
An error occurred (InvalidInputException) when calling the StartSandboxConnection
operation: Failed to start SSM session for {sandbox-arn}
User: arn:aws:sts::<account-ID>:assumed-role/<service-role-name>/AWSCodeBuild-<UUID>
is not authorized to perform: ssm:StartSession on resource.
```

```
An error occurred (InvalidInputException) when calling the StartSandboxConnection
operation: Failed to start SSM session for
sandbox <sandbox-arn>: codebuild:<UUID> is not connected.
```

Causa posible:

- Falta el agente de Systems Manager de Amazon EC2: la imagen de compilación no tiene el agente de SSM instalado o configurado correctamente.
- Permisos insuficientes: la función de servicio CodeBuild del proyecto carece de los permisos SSM necesarios.

Solución recomendada: si utiliza una imagen personalizada para la compilación, haga lo siguiente.

1. Instalar el SSM Agent de . Para obtener más información, consulte Instalación y desinstalación manual de SSM Agent en instancias de Amazon EC2 para Linux en . La versión de SSM Agent debe ser `3.0.1295.0` o posterior.
2. Copia el archivo en el `/etc/amazon/ssm/` directorio de tu imagen. https://github.com/aws/aws-codebuild-docker-images/blob/master/ubuntu/standard/7.0/amazon-ssm-agent.json De este modo se habilita Modo de contenedor en SSM Agent.
3. Asegúrese de que la función de servicio de su CodeBuild proyecto tenga los siguientes permisos y, a continuación, reinicie el entorno de pruebas:

```
{
   "Effect": "Allow",
      "Action": [
         "ssmmessages:CreateControlChannel",
         "ssmmessages:CreateDataChannel",
         "ssmmessages:OpenControlChannel",
         "ssmmessages:OpenDataChannel"
      ],
      "Resource": "*"
 },
 {
    "Effect": "Allow",
    "Action": [
       "ssm:StartSession"
     ],
     "Resource": [
        "arn:aws:codebuild:region:account-id:build/*",
        "arn:aws:ssm:region::document/AWS-StartSSHSession"
     ]
 }
```

Error: «No se pueden encontrar las credenciales» cuando se conecta mediante SSH a CodeBuild un entorno sandbox

Problema: al intentar conectarse a un entorno CodeBuild aislado mediante el comando`ssh codebuild-sandbox-ssh=<sandbox-arn>`, es posible que aparezca el siguiente error de credenciales:

```
Unable to locate credentials. You can configure credentials by running
```

```
"aws configure".
```

Posible causa: AWS las credenciales no se han configurado correctamente en su entorno local.

Solución recomendada: Configure sus AWS CLI credenciales siguiendo la documentación oficial: Configuración de los ajustes de la AWS CLI guía del usuario de la interfaz de línea de AWS comandos de la versión 2.

`StartSandboxConnectionAccessDeniedException`error al usar SSH en un CodeBuild entorno sandbox

Problema: al intentar conectarse a un entorno CodeBuild sandbox mediante el comando`ssh codebuild-sandbox-ssh=<sandbox-arn>`, es posible que aparezca el siguiente error de permiso:

```
An error occurred (AccessDeniedException) when calling the StartSandboxConnection
operation:
User: arn:aws:sts::account-id:assumed-role/role-name
is not authorized to perform: codebuild:StartSandboxConnection on resource:
sandbox-arn
because no identity-based policy allows the codebuild:StartSandboxConnection action
```

Posible causa: sus AWS credenciales carecen de los CodeBuild permisos necesarios para realizar esta operación.

Solución recomendada: asegúrese de que el usuario o rol de IAM asociado a sus AWS CLI credenciales tenga los siguientes permisos:

```
{
    "Effect": "Allow",
    "Action": [
       "codebuild:StartSandboxConnection"
     ],
     "Resource": [
        "arn:aws:codebuild:region:account-id:sandbox/*"
     ]
}
```

Error: «ssh: no se pudo resolver el nombre de host» cuando se conectó mediante SSH a un entorno sandbox CodeBuild

Problema: al intentar conectarse a un entorno CodeBuild aislado mediante el comando`ssh codebuild-sandbox-ssh=<sandbox-arn>`, aparece el siguiente error de resolución del nombre de host:

```
ssh: Could not resolve hostname
```

Posible causa: este error se produce normalmente cuando el script de conexión a un CodeBuild entorno aislado requerido no se ha ejecutado correctamente en el entorno local.

Solución recomendada:

1. Descargue el script de conexión a CodeBuild Sandbox.
2. Ejecute el script en el terminal para establecer la configuración de SSH necesaria.
3. Vuelva a intentar la conexión de SSH con el entorno de pruebas.

Depuración de compilaciones con Administrador de sesiones

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 prerrequisitos:

- 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 . La versión de SSM Agent debe ser 3.0.1295.0 o posterior.
  2. Copie el archivo en https://github.com/aws/aws-codebuild-docker-images/blob/master/ubuntu/standard/5.0/amazon-ssm-agent.jsonel `/etc/amazon/ssm/` directorio de la imagen. De este modo se habilita Modo de contenedor en SSM Agent.

  > **Note**
  >
  > Las imágenes personalizadas requieren el agente SSM más actualizado para que esta función funcione según lo esperado.

- El rol CodeBuild de servicio debe tener la siguiente política de SSM:

  JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
     {
       "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:

  JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "logs:DescribeLogGroups",
            "Resource": "arn:aws:logs:us-east-1:111122223333:log-group:*:*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": "arn:aws:logs:us-east-1:111122223333:log-
group:MyLogGroup:*"
        }
    ]
}
```

  Amazon S3

  Si utiliza Amazon S3 para almacenar sus registros, añada el siguiente permiso al rol CodeBuild de servicio:

JSON

```
{
  "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
```

Eliminar compilaciones AWS CodeBuild

Puede usar los SDK AWS CLI o los AWS SDK para eliminar las compilaciones incorporadas AWS CodeBuild.

Temas

- Eliminación de compilaciones (AWS CLI)
- Eliminar compilaciones (AWS SDK)

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)
  - Consulta una lista de los 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"
  ]
}
```

Eliminar compilaciones (AWS SDK)

Para obtener información sobre su uso AWS CodeBuild con los AWS SDK, consulta la. AWS Referencia de SDK y herramientas

Vuelve a intentar compilar manualmente en AWS CodeBuild

Puedes usar la AWS CodeBuild consola o los AWS CLI AWS SDK para volver a intentar manualmente una compilación única o una compilación por lotes. AWS CodeBuild

Temas

- Cómo reintentar hacer una compilación de forma manual (consola)
- Vuelva a intentar una compilación manualmente (AWS CLI)
- Vuelva a intentar una compilación manualmente (AWS SDKs)

Cómo reintentar hacer una compilación de forma manual (consola)

1. Abre la AWS CodeBuild consola en. https://console.aws.amazon.com/codesuite/codebuild/home
2. Realice una de las siguientes acciones:
   - Si aparece la *build-ID* página *build-project-name*:, selecciona Reintentar compilar.
   - En el panel de navegación, elija Historial de compilaciones. En la lista de compilaciones, seleccione la casilla correspondiente a la compilación y después elija Reintentar compilación.
   - En el panel de navegación, elija Proyectos de compilación. 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.

Vuelva a intentar una compilación manualmente (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 requerida. 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)
  - Consulta una lista de los identificadores de compilación de un proyecto de compilación (AWS CLI)
  - Consulta una lista de los 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 repites la `retry-build` solicitud con el mismo token, pero cambias un parámetro, se CodeBuild devolverá un error de discordancia entre los parámetros.

Vuelva a intentar una compilación manualmente (AWS SDKs)

Para obtener más información sobre el uso AWS CodeBuild con los AWS SDK, consulte. AWS Referencia de SDK y herramientas

Reintentar se integra automáticamente AWS CodeBuild

Puedes usar la AWS CodeBuild consola o los AWS SDK para volver a intentar tus compilaciones automáticamente. AWS CLI AWS CodeBuild Con el reintento automático activado, CodeBuild llamará automáticamente `RetryBuild` utilizando el rol de servicio del proyecto después de una acumulación fallida hasta un límite especificado. Por ejemplo, si el límite de reintentos automáticos está establecido en dos, CodeBuild llamará a la `RetryBuild` API para volver a intentar automáticamente la compilación hasta dos veces más.

> **Note**
>
> CodeBuild no admite el reintento automático para. CodePipeline

Temas

- Cómo reintentar hacer una compilación de forma automática (consola)
- Cómo reintentar hacer una compilación de forma automática (AWS CLI)
- Vuelva a intentar compilar automáticamente (AWS SDK)

Cómo reintentar hacer una compilación de forma automática (consola)

1. Abra la AWS CodeBuild consola en. https://console.aws.amazon.com/codesuite/codebuild/home
2. Elija Crear proyecto. Para obtener más información, consulte Creación de un proyecto de compilación (consola) y Ejecutar una compilación (consola).
   - En Environment (Entorno):
     - Como Auto-retry límite, introduce el número máximo de reintentos automáticos que desees después de una compilación fallida.
3. En Entorno, elija Configuración adicional.
4. Continúe con los valores predeterminados y, a continuación, elija Crear el proyecto de compilación.

Cómo reintentar hacer una compilación de forma automática (AWS CLI)

- Ejecute el comando create-project:

```
aws codebuild create-project \
    --name "<project-name>" \
    --auto-retry-limit <auto-retry-limit> \
    --source "<source>" \
    --artifacts {<artifacts>} \
    --environment "{\"type\": \"environment-type>\",\"image\": \"image-type>\",
\"computeType\": \"compute-type>\"}" \
    --service-role "service-role>"
```

En el comando anterior, sustituya los siguientes marcadores de posición:

- *<auto-retry-limit>*: establezca el límite de reintentos automáticos en el número máximo de reintentos automáticos deseado después de una compilación fallida.
- *<project-name>*,,*<source>*,, *<artifacts> environment-type> image-type>compute-type>*, y*service-role>*: Defina los ajustes de configuración del proyecto que desee.

Vuelva a intentar compilar automáticamente (AWS SDK)

Para obtener más información sobre su uso AWS CodeBuild con los AWS SDK, consulta la. AWS Referencia de SDK y herramientas

Detener las incorporaciones AWS CodeBuild

Puedes usar la AWS CodeBuild consola o AWS CLI AWS los SDK para detener una integración. AWS CodeBuild

Temas

- Detener una compilación (consola)
- Detener una compilación (AWS CLI)
- Detener una compilación (AWS SDKs)

Detener una compilación (consola)

1. Abre la AWS CodeBuild consola en. https://console.aws.amazon.com/codesuite/codebuild/home
2. Realice una de las siguientes acciones:
   - Si aparece la *build-ID* página *build-project-name*:, selecciona Detener la compilación.
   - En el panel de navegación, elija Historial de compilaciones. 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 Proyectos de compilación. 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 se puede detener correctamente una compilación (por ejemplo, si el proceso de creación ya se ha completado), el botón Detener está desactivado o puede que no aparezca.

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 requerida. 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)
  - Consulta una lista de los identificadores de compilación de un proyecto de compilación (AWS CLI)

Si se detiene AWS CodeBuild correctamente la compilación, el `buildStatus` valor del `build` objeto de la salida es`STOPPED`.

Si CodeBuild no se puede detener la compilación correctamente (por ejemplo, si la compilación ya está completa), el `buildStatus` valor del `build` objeto de la salida es el estado final de la compilación (por ejemplo,`SUCCEEDED`).

Detener una compilación (AWS SDKs)

Para obtener más información sobre el uso AWS CodeBuild con los AWS SDK, consulte. AWS Referencia de SDK y herramientas

Detenga las compilaciones por lotes AWS CodeBuild

Puedes usar la AWS CodeBuild consola o AWS CLI AWS los SDK para detener la creación por lotes. AWS CodeBuild

> **Note**
>
> Si utiliza computación Lambda en la compilación por lotes, la compilación Lambda en curso no se puede detener.

Temas

- Detener una compilación por lotes (consola)
- Detener una compilación por lotes (AWS CLI)
- Detener una compilación por lotes (AWS SDKs)

Detener una compilación por lotes (consola)

1. Abre la AWS CodeBuild consola en. https://console.aws.amazon.com/codesuite/codebuild/home
2. Realice una de las siguientes acciones:
   - Si aparece la *build-ID* página *build-project-name*:, selecciona Detener la compilación.
   - En el panel de navegación, elija Historial de compilaciones. 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 Proyectos de compilación. 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.

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 requerida. 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)
    - Consulta una lista de los identificadores de compilación por lotes de un proyecto de compilación (AWS CLI)

Detener una compilación por lotes (AWS SDKs)

Para obtener más información sobre el uso AWS CodeBuild con los AWS SDK, consulte. AWS Referencia de SDK y herramientas

Trigger AWS CodeBuild se crea automáticamente

Puede crear un desencadenador 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 sobre cómo crear y editar desencadenadores, consulte Crea activadores AWS CodeBuild y Editar AWS CodeBuild activadores.

Temas

- Crea activadores AWS CodeBuild
- Editar AWS CodeBuild activadores

Crea activadores AWS CodeBuild

Puede crear un desencadenador 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.

Temas

- Crear AWS CodeBuild activadores (consola)
- Cree AWS CodeBuild activadores mediante programación

Crear AWS CodeBuild activadores (consola)

Utilice el siguiente procedimiento para crear desencadenadores mediante la Consola de administración de AWS.

Para crear un disparador

1. Abre la AWS CodeBuild consola 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 activador diario que comienza todos los días a las P.M 12:15. tiene una hora de inicio de 12 y un minuto de inicio de 15. Un desencadenador diario que comienza cada día a medianoche tiene una hora de inicio 0 y un minuto de inicio 0. Un desencadenante diario que comienza todos los días a las P.M 11:59. tiene una hora de inicio de 23 y un minuto de inicio de 59.

Frecuencia	Parámetros requeridos	Details
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).
Weekly	Minuto de inicio Hora de inicio	Utilice el menú desplegable Start minute (Minuto de inicio).

Frecuencia	Parámetros requeridos	Details
	Día de inicio	Utilice el menú desplegable Start hour (Hora de inicio). Utilice el menú desplegable Start day (Día de inicio).
Personalizada	Expresión Cron	Introduzca una expresión Cron en Cron expressio 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.

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.
- Para AWS CodeCommit, escriba 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.
- Para GitHub 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 y 2160 minutos (36 horas). Este valor especifica cuánto tiempo AWS CodeBuild intenta una compilación antes de que se detenga. 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.

Cree AWS CodeBuild activadores 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 mediante programación. Consulta la referencia EventBridge de API de Amazon para obtener más información.

Editar AWS CodeBuild activadores

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 Crea activadores AWS CodeBuild.

Temas

- Editar AWS CodeBuild activadores (consola)
- Edite los AWS CodeBuild activadores mediante programación

Editar AWS CodeBuild activadores (consola)

Utilice el siguiente procedimiento para editar desencadenadores mediante la Consola de administración de AWS.

Para editar un desencadenador

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. 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

activador diario que comienza todos los días a las P.M 12:15. tiene una hora de inicio de 12 y un minuto de inicio de 15. Un desencadenador diario que comienza cada día a medianoche tiene una hora de inicio 0 y un minuto de inicio 0. Un desencadenante diario que comienza todos los días a las P.M 11:59. tiene una hora de inicio de 23 y un minuto de inicio de 59.

Frecuencia	Parámetros requeridos	Details
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).
Weekly	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	Details
Personalizada	Expresión Cron	Introduzca una expresión Cron en Cron expressio 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.

7. Seleccione Enable this trigger (Activar este disparador).

> **Note**
>
> Puedes usar la CloudWatch consola de Amazon https://console.aws.amazon.com/cloudwatch/para editar la versión de origen, el tiempo de espera y otras opciones que no están disponibles en AWS CodeBuild.

Edite los AWS CodeBuild activadores 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.

Vea los detalles de la construcción en AWS CodeBuild

Puedes usar la AWS CodeBuild consola o los AWS SDK para ver los detalles de las compilaciones administradas por CodeBuild. AWS CLI

Temas

- Ver detalles de las compilaciones (consola)
- Ver detalles de las compilaciones (AWS CLI)
- Ver los detalles de la compilación (AWS SDK)
- Transiciones de fases de compilación

Ver detalles de las compilaciones (consola)

1. Abre la AWS CodeBuild consola en. https://console.aws.amazon.com/codesuite/codebuild/home
2. Realice una de las siguientes acciones:

   - En el panel de navegación, elija Historial de compilaciones. 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 Proyectos de compilación. 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 el uso del AWS CLI with AWS CodeBuild, consulte laReferencia 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)
  - Consulta una lista de los 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
```

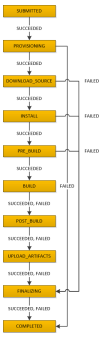
Si el comando tiene éxito, aparecen datos similares a los descritos en Para ver la información resumida de la compilación en la salida.

Ver los detalles de la compilación (AWS SDK)

Para obtener más información sobre su uso AWS CodeBuild con los AWS SDK, consulta la. AWS Referencia de SDK y herramientas

Transiciones de fases de compilación

Las compilaciones se llevan AWS CodeBuild a cabo 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

Puedes usar la AWS CodeBuild consola o los AWS SDK para ver una lista de los ID de compilación de las compilaciones administradas por CodeBuild. AWS CLI

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 (AWS SDK)

Ver una lista de identificadores de compilación (consola)

1. Abre la AWS CodeBuild consola en. https://console.aws.amazon.com/codesuite/codebuild/home
2. En el panel de navegación, elija Historial de compilaciones.

> **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 el uso del AWS CLI with CodeBuild, consulte laReferencia 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 enumerar los ID 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 el uso de AWS CLI with CodeBuild, consulte laReferencia 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 utilizada para indicar cómo enumerar los ID de compilación del lote. 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...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-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 (AWS SDK)

Para obtener más información sobre su uso CodeBuild con los AWS SDK, consulta la. AWS Referencia de SDK y herramientas

Ver una lista de los identificadores de compilación de un proyecto de compilación en AWS CodeBuild

Puedes usar la AWS CodeBuild consola o los AWS CLI AWS SDK para ver una lista de los ID de compilación de un proyecto de compilación. CodeBuild

Temas

- Ver una lista de identificadores de compilación de un proyecto de compilación (consola)
- Consulta una lista de los identificadores de compilación de un proyecto de compilación (AWS CLI)
- Consulta una lista de los identificadores de compilación por lotes de un proyecto de compilación (AWS CLI)
- Consulta una lista de los identificadores de compilación de un proyecto de compilación (AWS SDKs)

Ver una lista de identificadores de compilación de un proyecto de compilación (consola)

1. Abre la CodeBuild consola en. https://console.aws.amazon.com/codebuild/
2. En el panel de navegación, elija Proyectos de compilación. 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.

Consulta una lista de los identificadores de compilación de un proyecto de compilación (AWS CLI)

Para obtener más información sobre el uso de AWS CLI with AWS CodeBuild, consulta laReferencia 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*: La cadena obligatoria se usa para indicar el nombre del proyecto de compilación para el que se muestran los ID 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 usa para indicar cómo enumerar los ID 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 ejecutas un comando similar al siguiente:

```
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"
  ]
}
```

Consulta una lista de los identificadores de compilación por lotes de un proyecto de compilación (AWS CLI)

Para obtener más información sobre el uso de AWS CLI with AWS CodeBuild, consulta laReferencia 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*: La cadena obligatoria se usa para indicar el nombre del proyecto de compilación para el que se muestran los ID 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 usa para indicar cómo enumerar los ID 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 ejecutas un comando similar al siguiente:

```
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...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-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"
  ]
}
```

Consulta una lista de los identificadores de compilación de un proyecto de compilación (AWS SDKs)

Para obtener más información sobre el uso AWS CodeBuild con los AWS SDK, consulte. AWS Referencia de SDK y herramientas

Informes de prueba en AWS CodeBuild

Puede crear informes CodeBuild que contengan detalles sobre las pruebas que se realizan 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)
- Visual Studio TRX XML (.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. Se crea un nuevo informe de prueba en el grupo de informes cada vez que se ejecutan los casos de prueba. No es necesario crear un grupo de informes antes de ejecutar las pruebas. Si especificas un nombre de grupo de informes, CodeBuild crea un grupo de informes automáticamente cuando ejecutas tus 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 los permisos de carga en el bucket de S3.

Temas

- Creación de informes de pruebas
- Creación de informes de cobertura de código
- Auto-discover informes en CodeBuild
- Grupos de informes
- Marcos de pruebas
- Consultar los informes de pruebas
- Permisos de informes de prueba
- Estados de un informe de pruebas

Creación de informes 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 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 especificas un ARN, CodeBuild usa 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, con el formato *<project-name>*-*<report-group-name>*. Si el grupo de informes con nombre ya existe, CodeBuild usa 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 Ejecute AWS CodeBuild compilaciones manualmente.

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.

Creación de 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.

```
line coverage = (total lines covered)/(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 informes de cobertura de código:

- JaCoCo XML
- SimpleCov JSON[1]
- Clover XML
- Cobertura XML
- INFORMACIÓN DE 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 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 Ejecute AWS CodeBuild compilaciones manualmente.

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.

Auto-discover informes en CodeBuild

Con la detección automática, CodeBuild busca en todos los archivos de compilación una vez finalizada la fase de creación, busca todos los tipos de archivos de informes compatibles y crea automáticamente nuevos grupos e informes de análisis 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 informes descubiertos tienen el mismo tipo de formato, se colocarán en el mismo grupo de informes o informe.

La detección automática de informes se configura mediante 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 característica, establezca `CODEBUILD_CONFIG_AUTO_DISCOVER` en `false`.

`CODEBUILD_CONFIG_AUTO_DISCOVER_DIR`

(Opcional) Esta variable determina dónde se CodeBuild buscan los posibles archivos de informes. Tenga en cuenta que CodeBuild las búsquedas se realizan `**/*` de forma predeterminada.

Estas variables de entorno pueden modificarse durante la fase de compilación. Por ejemplo, si solo quiere habilitar la detección automática de informes para las compilaciones en la ramificación de git `main`, puede comprobar dicha ramificación durante el proceso de compilación y establecer `CODEBUILD_CONFIG_AUTO_DISCOVER` como false si la compilación no está en la ramificación

`main`. La detección automática de informes puede deshabilitarse mediante la consola o mediante variables de entorno del proyecto.

Temas

- Configuración de la detección automática de informes mediante la consola
- Configuración de la detección automática de informes mediante variables de entorno del proyecto

Configuración de la detección automática de informes mediante la consola

Utilice el siguiente procedimiento para configurar 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 uno para editarlo. Para obtener información, consulte Creación de un proyecto de compilación en AWS CodeBuild o Cambie la configuración del proyecto de compilación en AWS CodeBuild.
2. En Entorno, seleccione Configuración adicional.
3. Para deshabilitar la detección automática de informes, en Detección automática de informes, seleccione Desactivar la detección automática de informes.
4. (Opcional) En el Auto-discover directorio (opcional), introduzca un patrón de directorio CodeBuild para buscar los archivos con formato de informe compatibles. Tenga en cuenta que la CodeBuild búsqueda se realiza `**/*` de forma predeterminada.

Configuración de la detección automática de informes mediante variables de entorno del proyecto

Utilice el siguiente procedimiento para configurar la detección automática de informes mediante las 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 uno para editarlo. Para obtener información, consulte Creación de un proyecto de compilación en AWS CodeBuild o Cambie la configuración del proyecto de compilación en AWS CodeBuild.
2. En Variables del entorno, haga lo siguiente:

a. Para deshabilitar la detección automática de informes, en Nombre, escriba **`CODEBUILD_CONFIG_AUTO_DISCOVER`** y en Valor, especifique **`false`**. Al hacerlo, se deshabilita la detección automática de informes.

b. (Opcional) En Nombre, introduzca **`CODEBUILD_CONFIG_AUTO_DISCOVER_DIR`** y en Valor, introduzca el directorio en el que CodeBuild debe buscar los archivos con formato de informe compatibles. Por ejemplo, `output/*xml` busca archivos `.xml` en el directorio `output`.

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
- Nomenclatura de grupos de informes
- Uso compartido de grupos de informes
- Especificar archivos de prueba
- Especificar comandos de prueba
- Etiquetar un grupo de informes en AWS CodeBuild
- Actualizar un grupo de informes

Crear un grupo de informes

Puedes usar la CodeBuild consola AWS CLI, el archivo o un archivo de especificaciones de compilación 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 Permisos de informes de prueba.

Temas

- Crear un grupo de informes (buildspec)
- Crear un grupo de informes (consola)
- Crear un grupo de informes (CLI)
- Cree un grupo de informes (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. Elige un nombre de grupo de informes que no esté asociado a ningún grupo de informes de tu 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)

Utilice el siguiente procedimiento para crear un grupo de informes con la Consola de administración de AWS.

Para crear un grupo de informes

1. Abre 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 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 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. Puedes 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. Seleccione una de las siguientes opciones:

    - Clave AWS gestionada predeterminada Clave administrada de AWS para usar en 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 CodeBuild de servicio especificado en el proyecto se usa para los permisos de carga en el bucket de S3.

7. Elija Create report group (Crear grupo de informes).

Crear un grupo de informes (CLI)

Utilice el siguiente procedimiento para crear un grupo de informes con la AWS 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"
      }
    ]
  }
}
```

- *<bucket-name>*Sustitúyalo por el nombre de su bucket de Amazon S3 y *<path>* por la ruta de su bucket a la que 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 CodeBuild de servicio especificado en el proyecto se usa para los permisos de carga en el bucket de S3.

3. Use el siguiente comando:

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

Cree un grupo de informes (CloudFormation)

Siga las instrucciones a continuación para crear un grupo de informes con la plantilla de CloudFormation .

Para crear un grupo de informes con la plantilla de CloudFormation

Puede usar un archivo de CloudFormation plantilla para crear y aprovisionar un grupo de informes. Para obtener más información, consulte la CloudFormation Guía del usuario de .

La siguiente plantilla de CloudFormation YAML crea un grupo de informes que no exporta los archivos de resultados de las pruebas sin procesar.

```
Resources:
  CodeBuildReportGroup:
    Type: AWS::CodeBuild::ReportGroup
    Properties:
      Name: my-report-group-name
      Type: TEST
      ExportConfig:
        ExportConfigType: NO_EXPORT
```

La siguiente plantilla de CloudFormation YAML crea un grupo de informes que exporta los archivos de resultados de las 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: amzn-s3-demo-bucket
          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 los permisos de carga en el bucket de S3.

Nomenclatura de grupos de informes

Cuando usa la AWS CLI 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 deseas CodeBuild crear un nuevo grupo de informes, especifica 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 ningún grupo de informes en tu AWS cuenta con ese nombre `new-report-group` y lo especificas 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`

Uso compartido de grupos de informes

El uso compartido de grupos de informes permite que varias AWS cuentas o usuarios vean un grupo de informes, sus informes vigentes 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.

Temas

- Uso compartido de un grupo de informes
- Servicios relacionados
- Acceso a los grupos de informes compartidos con usted
- Cómo dejar de compartir un grupo de informes compartido
- Identificación de un grupo de informes compartido
- Permisos de un grupo de informes compartido

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 usarlo AWS CLI para ver el grupo de informes, sus informes y los resultados de los casos de prueba de cada informe. El consumidor no puede:

- Ve un grupo de informes compartido o sus informes en la CodeBuild consola.
- 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 usar la CodeBuild consola para agregar 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 CodeBuild consola, la AWS RAM consola o AWS CLI para compartir los grupos de informes de su propiedad.

Requisito previo

Para compartir un grupo de informes, su AWS cuenta debe ser la propietaria del mismo. No puede compartir un grupo de informes que se haya compartido con usted.

Para compartir un grupo de informes de tu propiedad (CodeBuild consola)

1. Abre 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 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 los grupos de informes de su propiedad (AWS RAM consola)

Consulte Creación de un uso compartido de recursos en la Guía del usuario de AWS RAM .

Para compartir los grupos de informes de su propiedad (AWS RAM (comando)

Utilice el comando create-resource-share.

Para compartir un grupo de informes de su propiedad (CodeBuild comando)

Utilice el comando put-resource-policy:

1. Cree un archivo llamado `policy.json` y copie lo siguiente en él.

   JSON

```
{
   "Version":"2012-10-17",
   "Statement":[{
     "Effect":"Allow",
     "Principal":{
       "AWS":"111122223333"
     },
     "Action":[
       "codebuild:BatchGetReportGroups",
       "codebuild:BatchGetReports",
       "codebuild:ListReportsForReportGroup",
       "codebuild:DescribeTestCases"],
     "Resource":"arn:aws:iam::*:role/Service*"
   }]
 }
```

2. Actualice `policy.json` con el ARN del grupo de informes y los identificadores para compartirlo. El siguiente ejemplo concede acceso de solo lectura al grupo de informes con el ARN a Alice y `arn:aws:codebuild:us-west-2:123456789012:report-group/my-report-group` al usuario raíz de la AWS cuenta identificada con 123456789012.

   JSON

```
{
   "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 comando siguiente.

```
aws codebuild put-resource-policy --resource-arn report-group-arn --policy file://
policy.json
```

Servicios relacionados

El uso compartido de grupos de informes se integra con AWS Resource Access Manager (AWS RAM), un servicio que permite compartir los recursos con cualquier cuenta o a través de ella. AWS AWS AWS Organizations Con AWS RAM, compartes los recursos de tu propiedad creando un recurso compartido que especifica los recursos y los consumidores con los que compartirlos. Los consumidores pueden ser AWS cuentas individuales, unidades organizativas o toda una organización AWS Organizations. AWS Organizations

Para obtener más información, consulte la Guía del usuario de AWS RAM.

Acceso a los 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.

Cómo dejar de compartir un grupo de informes compartido

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 dejas de compartir un grupo de informes, cualquier AWS cuenta o usuario con el que lo hayas compartido anteriormente no podrá acceder al grupo de informes, a sus informes ni a los resultados de los casos de prueba incluidos en los informes.

Para dejar de compartir un grupo de informes compartido que posee, debe quitarlo del recurso compartido. Puedes usar la AWS RAM consola o AWS CLI hacer esto.

Para dejar de compartir un grupo de informes compartido de su propiedad (AWS RAM consola)

Consulte Actualización de un recurso compartido en la Guía del usuario de AWS RAM .

Para dejar de compartir un grupo de informes compartido de su propiedad (AWS RAM (comando)

Utilice el comando disassociate-resource-share.

Para dejar de compartir un grupo de informes de su propiedad (CodeBuild comando)

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 utilizarla AWS CLI para identificar los 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.

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.

Etiquetar un grupo de informes en AWS CodeBuild

Una etiqueta es una etiqueta de atributo personalizada que usted o AWS asigna a un AWS recurso. Cada AWS etiqueta consta de dos partes:

- Una clave de etiqueta (por ejemplo, `CostCenter`, `Environment`, `Project` o `Secret`). Las claves de etiquetas distinguen entre mayúsculas y minúsculas.
- Un campo opcional que se denomina valor de etiqueta (por ejemplo, `111122223333`, `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 etiquetas, los valores de las etiquetas distinguen mayúsculas de 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 Tags.

Las etiquetas ayudan a identificar y organizar AWS los recursos. Muchos AWS servicios 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 CodeBuild informes que 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 usar la CodeBuild consola AWS CLI, las CodeBuild API o AWS los SDK para agregar, administrar y eliminar 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 AWS CodeBuild recursos.

Temas

- Adición de etiquetas a un grupo de informes
- Consultar etiquetas de un grupo de informes
- Editar etiquetas de un grupo de informes
- Eliminación de etiquetas de un grupo de informes

Adición de etiquetas a un grupo de informes

Agregar etiquetas a un grupo de informes puede ayudarle a identificar y organizar sus AWS recursos 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 Tags. 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 CodeBuild consola 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 AWS CodeBuild recursos.

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 CodeBuild consola para agregar una o más etiquetas a un grupo de CodeBuild informes.

1. Abra la CodeBuild consola 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 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 Clave, escriba un nombre para la etiqueta. Puede agregar un valor opcional para la etiqueta en 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 la AWS Command Line Interface.

Consultar etiquetas de un grupo de informes

Las etiquetas pueden ayudarle a identificar y organizar sus AWS recursos y a 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 CodeBuild consola para ver las etiquetas asociadas a un grupo de CodeBuild informes.

1. Abra la CodeBuild consola 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 Configuración.

Consultar etiquetas de un grupo de informes (AWS CLI)

Siga estos pasos para usar el AWS CLI para ver las AWS etiquetas de un grupo de informes. Si no se han agregado etiquetas, la lista de etiquetas devueltas está vacía.

1. Utilice la consola o el AWS CLI para localizar el ARN de su grupo de informes. Anótelo.

   AWS CLI

   Ejecute el comando siguiente.

   ```
   aws list-report-groups
   ```

   Este comando devuelve JSON-formatted información 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 CodeBuild consola 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 comando siguiente. 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 se ejecuta correctamente, este comando devuelve JSON-formatted información que contiene una `tags` sección 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 agregar 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 Tags.

> **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 CodeBuild consola para editar las etiquetas asociadas a un grupo de CodeBuild informes.

1. Abra la CodeBuild consola 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 Configuración.
4. Elija Edit (Edición de).
5. Realice 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`.

Eliminación de etiquetas 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 AWS recursos que estén asociados a 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 AWS CodeBuild recursos.

Quitar una etiqueta de un grupo de informes (consola)

Puede usar la CodeBuild consola para eliminar la asociación entre una etiqueta y un grupo de CodeBuild informes.

1. Abra la CodeBuild consola 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 Configuración.
4. Elija Edit (Edición de).
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 usar el AWS CLI para eliminar una etiqueta de un grupo de CodeBuild informes. 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 CodeBuild informes, todas las asociaciones de etiquetas se eliminan 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 `tags` sección de JSON-formatted los datos con una lista actualizada de etiquetas que no contenga las que desee eliminar. Si desea eliminar todas las etiquetas, actualice la sección `tags` para:

```
"tags: []"
```

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 lo usa AWS CLI para actualizar un grupo de informes, también puede actualizar o agregar etiquetas. Para obtener más información, consulte Etiquetar un grupo de informes en AWS CodeBuild.

> **Note**
>
> El rol CodeBuild de servicio especificado en el proyecto se usa para los 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)

Utilice el siguiente procedimiento para actualizar un grupo de informes con la Consola de administración de AWS.

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 Edit (Edición de).
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 gestionada predeterminada Clave administrada de AWS para usar en 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)

Utilice el siguiente procedimiento para actualizar un grupo de informes con la AWS 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. Use el siguiente comando:

```
aws codebuild update-report-group \
--cli-input-json file://UpdateReportGroupInput.json
```

Marcos de pruebas

Los temas de esta sección muestran cómo configurar los informes de prueba AWS CodeBuild para varios marcos de prueba.

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 los informes de pruebas en el marco de pruebas AWS CodeBuild de JasmineBDD.

El procedimiento requiere los siguientes requisitos previos:

- Tienes un proyecto existente. CodeBuild
- Tu proyecto es un Node.js proyecto 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 JavaScript informadoras 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 reporteros de prueba de Jasmine:

JUnitXmlReporter

Se utiliza para generar informes en el formato `JunitXml`.

NUnitXmlReporter

Se utiliza para generar informes en el formato `NunitXml`.

Un Node.js proyecto con Jasmine tendrá, de forma predeterminada, un `spec` subdirectorio 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 prueba se exportan al archivo especificado por*<test report directory>*/*<report filename>*.

En su `buildspec.yml` archivo, add/update 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 `NunitXml` informe, cambie el `file-format` valor por el siguiente.

```
    file-format: NUNITXML
```

Configurar generación de informes de pruebas con Jest

El siguiente procedimiento muestra cómo configurar los informes de pruebas AWS CodeBuild con el marco de pruebas de Jest.

El procedimiento requiere los siguientes requisitos previos:

- Tienes un CodeBuild proyecto existente.
- Su proyecto es un Node.js proyecto que está configurado para usar el marco de pruebas de Jest.

Agrega el `jest-junit`paquete a la `devDependencies` sección del `package.json` archivo de tu proyecto. CodeBuild utiliza este paquete para generar informes en este `JunitXml` formato.

```
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 las pruebas se exportan al archivo especificado por*<test report directory>*/*<report filename>*.

```
module.exports = {
  reporters: [
    'default',
    [ 'jest-junit', {
      outputDirectory: <test report directory>,
      outputName: <report filename>,
    } ]
  ]
};
```

En su `buildspec.yml` archivo, add/update 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 los informes de pruebas AWS CodeBuild con el marco de pruebas de pytest.

El procedimiento requiere los siguientes requisitos previos:

- Tiene un CodeBuild proyecto 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 descubre automáticamente las pruebas en el directorio actual y exporta los informes de las pruebas al archivo especificado por*<test report directory>*/*<report filename>*. El informe utiliza el formato `JunitXml`.

```
      - python -m pytest --junitxml=<test report directory>/<report filename>
```

En su `buildspec.yml` archivo, add/update 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 los informes de pruebas AWS CodeBuild con el marco de pruebas RSpec.

El procedimiento requiere los siguientes requisitos previos:

- Tiene un proyecto existente CodeBuild .
- Su proyecto es un proyecto Ruby que está configurado para usar el marco de pruebas de RSpec.

Add/update lo siguiente en su `buildspec.yml` archivo. Este código ejecuta las pruebas en el *<test source directory>* directorio y exporta los informes de las pruebas al archivo especificado por*<test report directory>*/*<report filename>*. 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
 directory>/<report filename>
reports:
    rspec_reports:
        files:
            - <report filename>
        base-directory: <test report directory>
        file-format: JUNITXML
```

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
- Vea los informes de las pruebas en su AWS inscrita

Consultar los informes de pruebas de una compilación

Para ver los informes de pruebas de una compilación

1. Abre 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. Abre 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.

Vea los informes de las pruebas en su AWS inscrita

Para ver los informes de las pruebas en su AWS inscrita

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.

Permisos de informes de prueba

En este tema se describe información importante acerca de los permisos relacionados con los informes de pruebas.

Temas

- Rol de IAM para informes de pruebas
- Permisos para las operaciones de informes de pruebas
- Ejemplos de permisos de informes de pruebas

Rol de IAM para 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 AWS administradas predefinidas. 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 puedes llamar a un AWS CLI comando o método de 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 los 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 prueba:

- `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

Estados de un informe de pruebas

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 Permisos de informes de prueba.
  - 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 según el estado `FAILED` y trunca los resultados de los casos de prueba.

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 VPC-specific configuración adicional en la configuración CodeBuild del proyecto. Esto incluye el ID de VPC, los ID de subred de VPC y los ID de grupos de seguridad de VPC. VPC-enabled De este modo, las compilaciones pueden acceder a los recursos 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
- Prácticas recomendadas para las VPC
- Limitaciones de las VPC
- Permita el acceso de Amazon VPC a sus proyectos CodeBuild
- Solución de problemas de la configuración de la VPC
- Uso de puntos de enlace de la VPC
- Uso AWS CodeBuild con un servidor proxy gestionado
- Uso AWS CodeBuild con un servidor proxy
- CloudFormation Plantilla de VPC

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.
- Recupera las dependencias de repositorios de artefactos internos autohospedados, 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.

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 CloudFormation cómo configurar una VPC para usar la función CodeBuild VPC, consulte la. CloudFormation Plantilla de VPC

Limitaciones de las VPC

- La conectividad de VPC desde no CodeBuild es compatible con las VPC compartidas.

Permita 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 Security Groups (Grupos de seguridad), elija los grupos de seguridad que usa CodeBuild 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. Cree un proyecto de compilación (AWS CLI) Si utiliza la función AWS CLI with CodeBuild, la función de servicio que utilice CodeBuild para interactuar con los servicios en nombre del usuario de IAM debe tener una política adjunta. Para obtener información, consulte Permitir el CodeBuild acceso a AWS servicios necesarios para crear una interfaz de red de VPC.

El *vpcConfig* objeto debe incluir su *vpcIdsecurityGroupIds*, y*subnets*.

- *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.

Solución de problemas de la configuración de la VPC

Utilice la información que aparece en el mensaje de error para ayudarle a identificar, diagnosticar y resolver los problemas.

A continuación, se indican algunas pautas que le ayudarán a solucionar un error frecuente de la VPC de CodeBuild: `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 que utiliza CodeBuild para interactuar con los servicios en nombre del usuario de IAM tenga los permisos de esta política. Para obtener más información, consulte CodeBuild Permitir interactuar con otros AWS servicios.

   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.

Uso de puntos de enlace de la VPC

Puede mejorar la seguridad de sus compilaciones configurándolas AWS CodeBuild para usar un punto final de VPC de interfaz. Los puntos de enlace de la interfaz funcionan con una tecnología que puede utilizar para acceder de forma privada a Amazon EC2 CodeBuild y mediante direcciones IP privadas. PrivateLink PrivateLink restringe todo el tráfico de red entre las instancias gestionadas y Amazon EC2 a la red de Amazon. CodeBuild 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 sobre PrivateLink los puntos de enlace 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 VPC AWS CodeBuild, tenga en cuenta las siguientes restricciones y limitaciones.

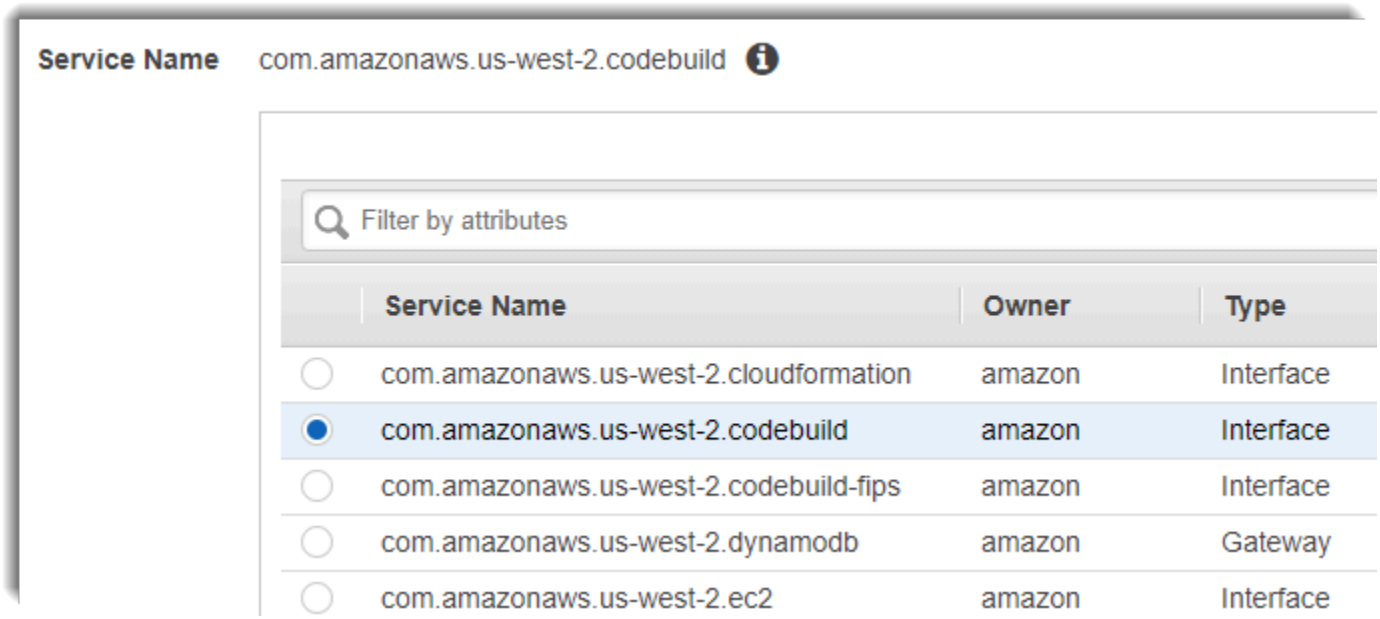
> **Note**
>
> Utilice una puerta de enlace NAT si desea utilizarla CodeBuild con AWS servicios que no admiten las conexiones de Amazon VPC PrivateLink.

- Los puntos de enlace de VPC admiten DNS únicamente Amazon-provided 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 final en la misma AWS región que cualquier depósito de S3 que almacene las entradas y salidas 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 Region-specific Amazon S3 para acceder a su bucket (por ejemplo,`<bucket-name>.s3-us-west-2.amazonaws.com`). Para obtener más información acerca de Region-specific los puntos de enlace de Amazon S3, consulte Amazon Simple Storage Service en. Referencia general de Amazon Web Services Si utilizas el AWS CLI para realizar solicitudes a Amazon S3, establece tu región predeterminada en la misma región en la que se creó el bucket o usa el `--region` parámetro en tus solicitudes.

Cree puntos de enlace de VPC para CodeBuild

Siga las instrucciones de Creación de un punto de conexión de interfaz para crear el punto de conexión `com.amazonaws.region.codebuild`. Este es un punto final de VPC para. AWS CodeBuild



*region*representa el identificador de región de una AWS región compatible con CodeBuild, por ejemplo, la región EE.UU. Este (Ohio). `us-east-2` Para obtener una lista de AWS las regiones compatibles, consulte CodeBuildla Referencia AWS general. El punto final se rellena automáticamente con la región que especificó al iniciar sesión. AWS Si cambia la región, el punto de conexión de la VPC se actualizará en consecuencia.

Cree una política de puntos de conexión de VPC para CodeBuild

Puede crear una política para los puntos de enlace de Amazon VPC AWS CodeBuild en la que pueda especificar:

- La entidad principal 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.

Uso AWS CodeBuild con un servidor proxy gestionado

Para ejecutar flotas de capacidad AWS CodeBuild reservada en un servidor proxy administrado, debe configurar el servidor proxy para permitir o denegar el tráfico hacia y desde sitios externos mediante reglas de proxy. Tenga en cuenta que la ejecución de flotas de capacidad reservada en un servidor proxy administrado no es compatible con VPC, Windows o macOS.

> **Important**
>
> Hay costos adicionales en función del tiempo que una configuración de proxy esté presente en la flota. Para obtener más información, consulte https://aws.amazon.com/codebuild/pricing/.

Temas

- Configuración de un proxy administrado para las flotas de capacidad reservada
- Gestione una flota de capacidad CodeBuild reservada

Configuración de un proxy administrado para las flotas de capacidad reservada

Si quiere configurar un servidor proxy administrado para su flota de capacidad reservada, debe habilitar esta característica al crear la flota en la consola o al utilizar la AWS CLI. Hay varias propiedades que debe definir:

Definir configuraciones de proxy (opcional)

Configuraciones de proxy que aplican el control de acceso a la red a las instancias de capacidad reservada.

Comportamiento predeterminado

Define el comportamiento del tráfico saliente.

Permitir

Permite el tráfico saliente a todos los destinos de forma predeterminada.

Denegar

Deniega el tráfico saliente a todos los destinos de forma predeterminada.

Reglas de proxy

Especifica los dominios de destino a los que restringir el control de acceso a la red.

Para definir las configuraciones del proxy en la consola, consulte las instrucciones de Creación de una flota de capacidad reservada. Para definir las configuraciones de proxy mediante el AWS CLI, puede hacerlo modificando la siguiente sintaxis de JSON y guardando los resultados:

```
"proxyConfiguration": {
    "defaultBehavior": "ALLOW_ALL" | "DENY_ALL",
    "orderedProxyRules": [
        {
            "type": "DOMAIN" | "IP",
```

```
            "effect": "ALLOW" | "DENY",
            "entities": [
                "destination"
            ]
        }
    ]
}
```

El archivo JSON puede ser similar a lo siguiente:

```
"proxyConfiguration": {
    "defaultBehavior": "DENY_ALL",
    "orderedProxyRules": [
        {
            "type": "DOMAIN",
            "effect": "ALLOW",
            "entities": [
                "github.com"
            ]
        }
    ]
}
```

Gestione una flota de capacidad CodeBuild reservada

Al ejecutar flotas de capacidad AWS CodeBuild reservada con su servidor proxy administrado, CodeBuild configurará automáticamente sus variables `HTTP_PROXY` y las de `HTTPS_PROXY` entorno con las direcciones proxy administradas. Si el software de dependencia tiene su propia configuración y no se ajusta a las variables de entorno, puede consultar estos valores y actualizar la configuración del software en los comandos de compilación para enrutar el tráfico de compilación de forma correcta a través del proxy administrado. Para obtener más información, consulte Creación de un proyecto de compilación en AWS CodeBuild y Cambie la configuración del proyecto de compilación en AWS CodeBuild.

Uso AWS CodeBuild con un servidor proxy

Puede usarlo AWS CodeBuild con un servidor proxy para regular el tráfico HTTP y HTTPS hacia y desde Internet. Para ejecutarlo CodeBuild con un servidor proxy, instálalo en una subred pública y CodeBuild en una subred privada en una VPC.

Hay dos casos de uso principales para ejecutar CodeBuild 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.

Se puede usar CodeBuild con dos tipos de servidores proxy. En ambos casos, el servidor proxy se ejecuta en una subred pública y CodeBuild en una subred privada.

- Proxy explícito: si usa un servidor proxy explícito, debe configurar `NO_PROXY` y configurar las variables de `HTTPS_PROXY` entorno CodeBuild a nivel de proyecto. `HTTP_PROXY` Para obtener más información, consulte Cambie la configuración del 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

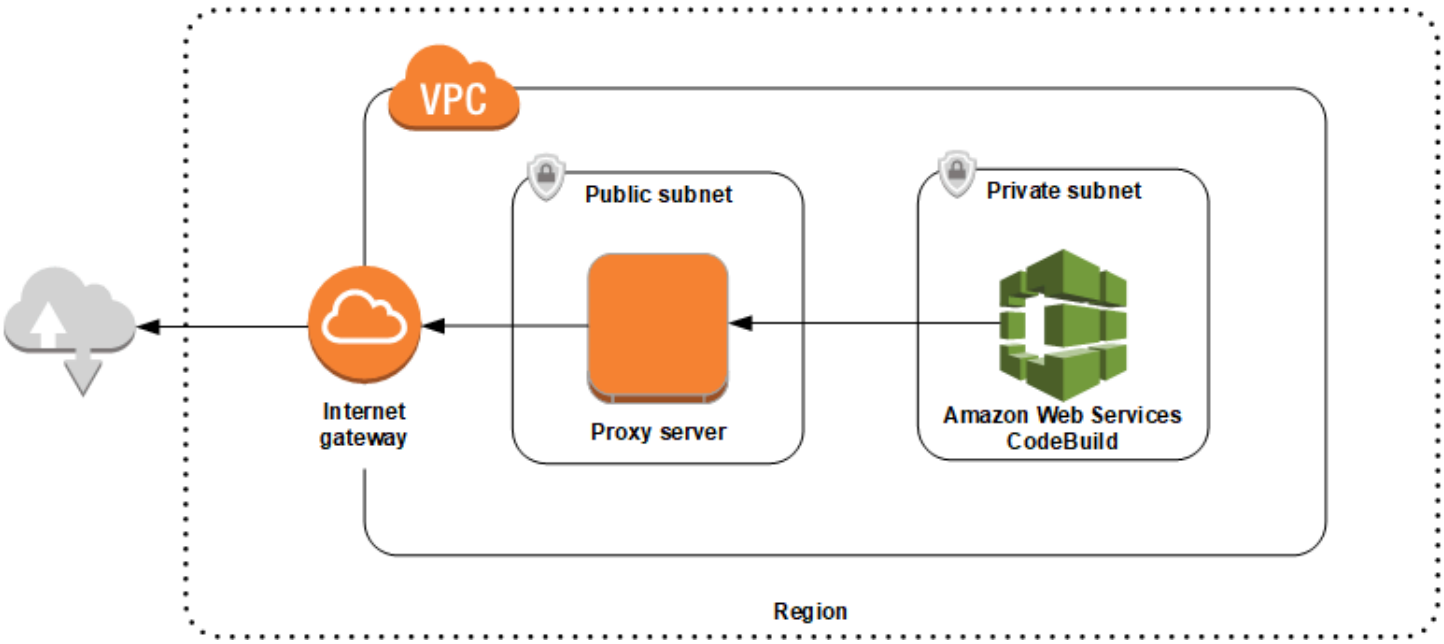
- Configure los componentes necesarios para ejecutarse CodeBuild en un servidor proxy
- Se ejecuta CodeBuild en un servidor proxy explícito
- Se ejecuta CodeBuild en un servidor proxy transparente
- Ejecución de un administrador de paquetes y otras herramientas en un servidor proxy

Configure los componentes necesarios para ejecutarse CodeBuild en un servidor proxy

Necesita estos componentes para ejecutarse AWS CodeBuild en un servidor proxy transparente o explícito:

- Una VPC.
- Una subred pública en la VPC para el servidor proxy.
- Una subred privada en la VPC para CodeBuild.
- Una puerta de enlace a Internet que permite 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

Los siguientes pasos son necesarios para ejecutarlo 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 es una subred privada denominada `Private Subnet` en la que CodeBuild se ejecuta.

   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 rutas predeterminada que dirija el tráfico saliente desde la VPC (0.0.0). 0/0) a la puerta de enlace 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 su VPC que permita la entrada del tráfico SSH (TCP 22) desde su 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 IP Auto-assign pública, elija 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 ejecutada la instancia EC2, desactive las source/destination comprobaciones. Para obtener más información, consulte la sección sobre la desactivación de las Source/Destination comprobaciones 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 direccione el tráfico destinado a Internet al servidor proxy. Asocie esta tabla de enrutamiento con la subred privada. Esto es necesario para que las solicitudes salientes de las instancias de su subred privada, donde CodeBuild se ejecutan, siempre se dirijan a través del servidor proxy.

Instalación y configuración de un servidor proxy

Existen muchos servidores proxy entre los que elegir. Aquí se utiliza un servidor proxy de código abierto, Squid, para demostrar cómo AWS CodeBuild funciona 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 función denominada SslPeekAndSplice para recuperar la indicación del nombre del servidor (SNI) de la iniciación de TLS que contiene el host de Internet solicitado. Esta operación es necesaria, ya que Squid no necesita descifrar el tráfico HTTPS. Para habilitarla SslPeekAndSplice, Squid requiere 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. De todos los mensajes de HTTP/1.1 solicitud, puede recuperar el campo del encabezado del host, que especifica el host de Internet que se solicita.

Se ejecuta CodeBuild en un servidor proxy explícito

Para ejecutarlo AWS CodeBuild en un servidor proxy explícito, debe configurar el servidor proxy para permitir o denegar el tráfico hacia y desde sitios externos y, a continuación, configurar las variables de `HTTPS_PROXY` entorno `HTTP_PROXY` y.

Temas

- Configuración de Squid como un servidor proxy explícito
- Cree un proyecto CodeBuild
- Ejemplo de archivo squid.conf de servidor proxy explícito

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. Edite la expresión regular de la última línea para especificar los buckets de S3 o un CodeCommit repositorio en una AWS región. 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 su fuente es AWS CodeCommit, utilícela `git-codecommit.<`*`your-region`*`>.amazonaws.com` para agregar una AWS región a una lista de 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. CodeBuild Permiten acceder CloudWatch a Amazon S3. CloudWatch Es necesario acceder a para CodeBuild poder crear CloudWatch registros. 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 RequestError tiempo de espera, consulteRequestError error de tiempo de espera cuando se ejecuta CodeBuild en un servidor proxy.

Para obtener más información, consulte Ejemplo de archivo `squid.conf` de servidor proxy explícito más adelante en este tema.

Cree un proyecto CodeBuild

Para ejecutarlo AWS CodeBuild con un servidor proxy explícito, defina sus variables `HTTP_PROXY` y de `HTTPS_PROXY` entorno con la dirección IP privada de la instancia EC2 que creó para su servidor proxy y el puerto 3128 a nivel de 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 Cambie la configuración del 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
```

Ejemplo de archivo squid.conf de 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
```

Se ejecuta CodeBuild en un servidor proxy transparente

Para ejecutarlo AWS CodeBuild en un servidor proxy transparente, debe configurar el servidor proxy para que acceda a los sitios web y dominios con los que interactúa.

Temas

- Configuración de Squid como un servidor proxy transparente
- Crea un proyecto CodeBuild

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 funcionar AWS CodeBuild con un servidor proxy transparente, debe concederle acceso`amazonaws.com`. También debe conceder acceso a otros CodeBuild usos de sitios web. Estos varían en función de cómo crees tus CodeBuild proyectos. Algunos ejemplos de sitios web son los de repositorios como Bitbucket GitHub, 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
```

Crea un proyecto CodeBuild

Después de configurar el servidor proxy, puede usarlo AWS CodeBuild en una subred privada sin necesidad de realizar más configuraciones. 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

Utilice los siguientes procedimientos para ejecutar 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. Añádela a la lista de permitidos del `squid.conf` archivo y añade un comando al archivo buildspec para CodeBuild conocer el punto final de tu 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
```

CloudFormation Plantilla de VPC

CloudFormation le permite crear y aprovisionar despliegues de AWS infraestructura de forma predecible y repetida, mediante el uso de archivos de plantilla para crear y eliminar un conjunto de recursos en conjunto como una sola unidad (una pila). Para obtener más información, consulte la Guía del usuario de CloudFormation.

La siguiente es una plantilla de CloudFormation YAML para configurar una VPC y utilizarla. 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
```

Inicio de sesión y supervisión AWS CodeBuild

El registro y la supervisión son una parte importante del mantenimiento de la fiabilidad, la disponibilidad y el rendimiento de AWS CodeBuild sus AWS soluciones. Debe recopilar datos de supervisión de todas las partes de la AWS solución para poder depurar más fácilmente un error multipunto, en caso de que se produzca. AWS proporciona las siguientes herramientas para supervisar sus CodeBuild recursos y compilaciones y para responder a posibles incidentes.

Temas

- Registro AWS CodeBuild Llamadas a la API con AWS CloudTrail
- El monitor CodeBuild se construye con CloudWatch

Registro AWS CodeBuild Llamadas a la API con AWS CloudTrail

AWS CodeBuild está integrado con AWS CloudTrail un servicio que proporciona un registro de las acciones realizadas por un usuario, un rol o un AWS servicio en CodeBuild. CloudTrail captura todas las llamadas a la API CodeBuild como eventos, incluidas las llamadas desde la CodeBuild consola y las llamadas de código a las CodeBuild API. Si creas un registro, puedes habilitar la entrega continua de CloudTrail eventos a un bucket de S3, incluidos los eventos para CodeBuild. Si no configuras un registro, puedes seguir viendo los eventos más recientes en la CloudTrail consola, en el Historial de eventos. Con la información recopilada por CloudTrail, puedes determinar la solicitud a la que se hizo CodeBuild, la dirección IP desde la que se hizo la solicitud, quién la hizo, cuándo se hizo y otros detalles.

Para obtener más información CloudTrail, consulta la Guía AWS CloudTrail del usuario.

Temas

- Acerca de AWS CodeBuild información en CloudTrail
- Acerca de AWS CodeBuild entradas de archivos de registros

Acerca de AWS CodeBuild información en CloudTrail

CloudTrail está activado en su AWS cuenta al crear la cuenta. Cuando se produce actividad en CodeBuild, esa actividad se registra en un CloudTrail evento junto con otros eventos de AWS servicio en el historial de eventos. Puedes ver, buscar y descargar los eventos recientes en tu AWS

cuenta. Para obtener más información, consulta la sección Visualización de CloudTrail eventos con el historial de eventos en la Guía del AWS CloudTrail usuario.

Para obtener un registro continuo de los eventos de tu AWS cuenta, incluidos los de tu cuenta CodeBuild, crea un registro. Un registro permite CloudTrail enviar archivos de registro a un bucket de S3. De manera predeterminada, cuando se crea un registro de seguimiento en la consola, el registro de seguimiento se aplica a todas las regiones de . El registro registra los eventos de todas las regiones de la AWS partición y envía los archivos de registro al bucket de S3 que especifique. Puede configurar otros AWS servicios para analizar más a fondo los datos de eventos recopilados en los CloudTrail registros y actuar en función de ellos. Para obtener más información, consulte lo siguiente:

- Introducción a la creación de registros de seguimiento
- CloudTrail servicios e integraciones compatibles
- Configurar las notificaciones de Amazon SNS para CloudTrail
- Recibir archivos de CloudTrail registro de varias regiones y Recibir archivos de CloudTrail registro de varias cuentas

Todas CodeBuild las acciones se registran en la referencia de la CodeBuild API CloudTrail y se documentan en ella. Por ejemplo, las llamadas a las acciones `CreateProject` (en AWS CLI,`create-project`), `StartBuild` (en AWS CLI,`start-project`) y `UpdateProject` (en el AWS CLI,`update-project`) generan entradas en los archivos de CloudTrail registro.

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.
- Si la solicitud se realizó con credenciales de seguridad temporales de un rol o fue un usuario federado.
- Si la solicitud fue realizada por otro AWS servicio.

Para obtener más información, consulte el elemento CloudTrail UserIdentity en la Guía del AWS CloudTrail usuario.

Acerca de AWS CodeBuild entradas de archivos de registros

Un registro es una configuración que permite la entrega de eventos como archivos de registro a un bucket de S3 que usted especifique. CloudTrail Los archivos de registro contienen una o más entradas de registro. Un evento representa una única solicitud de cualquier fuente e incluye información sobre la acción solicitada, la fecha y la hora de la acción, los parámetros de la solicitud, etc. CloudTrail los archivos de registro no son un registro ordenado de las llamadas a la API pública, por lo que no aparecen en ningún orden específico.

> **Note**
>
> Para proteger la información confidencial, en los CodeBuild registros se oculta 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.

El siguiente ejemplo muestra una entrada de CloudTrail registro que muestra 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"
}
```

El monitor CodeBuild se construye con CloudWatch

Puedes usar Amazon CloudWatch para observar tus compilaciones, informar cuando algo va mal y tomar medidas automáticas cuando sea necesario. 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 el `ProjectName` para la dimensión en CloudWatch.

AWS nivel de cuenta

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 utilización de los recursos de Build no están disponibles a nivel de AWS cuenta.

CloudWatch las métricas muestran el comportamiento de tus compilaciones a lo largo del tiempo. Por ejemplo, es posible monitorizar:

- Cuántas compilaciones se intentaron en un proyecto de compilación o en una AWS cuenta a lo largo del tiempo.
- Cuántas compilaciones tuvieron éxito en un proyecto de creación o en una AWS cuenta a lo largo del tiempo.
- Cuántas compilaciones fallaron en un proyecto de compilación o en una AWS cuenta a lo largo del tiempo.

- Cuánto tiempo se CodeBuild ha dedicado a ejecutar compilaciones en un proyecto de compilación o en una AWS cuenta 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 Ver CodeBuild métricas.

CodeBuild CloudWatch métricas

Se puede realizar un seguimiento de las siguientes métricas por AWS cuenta o proyecto de creación. Para obtener más información sobre el uso de CloudWatch con CodeBuild, consulteEl monitor CodeBuild se construye con CloudWatch.

BuildDuration

Mide la duración de la fase `BUILD` de la compilación.

Unidades: segundos

CloudWatch Estadísticas válidas: promedio (recomendado), máximo, mínimo

Compilaciones

Mide el número de compilaciones activadas.

Unidades: recuento

CloudWatch Estadísticas válidas: suma

DownloadSourceDuration

Mide la duración de la fase `DOWNLOAD_SOURCE` de la compilación.

Unidades: segundos

CloudWatch Estadísticas válidas: promedio (recomendado), máximo, mínimo

Duración

Mide la duración de todas las compilaciones con el paso del tiempo.

Unidades: segundos

CloudWatch Estadísticas válidas: promedio (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

CloudWatch Estadísticas válidas: suma

FinalizingDuration

Mide la duración de la fase `FINALIZING` de la compilación.

Unidades: segundos

CloudWatch Estadísticas válidas: promedio (recomendado), máximo, mínimo

InstallDuration

Mide la duración de la fase `INSTALL` de la compilación.

Unidades: segundos

CloudWatch Estadísticas válidas: promedio (recomendado), máximo, mínimo

PostBuildDuration

Mide la duración de la fase `POST_BUILD` de la compilación.

Unidades: segundos

CloudWatch Estadísticas válidas: promedio (recomendado), máximo, mínimo

PreBuildDuration

Mide la duración de la fase `PRE_BUILD` de la compilación.

Unidades: segundos

CloudWatch Estadísticas válidas: promedio (recomendado), máximo, mínimo

ProvisioningDuration

Mide la duración de la fase `PROVISIONING` de la compilación.

Unidades: segundos

CloudWatch Estadísticas válidas: promedio (recomendado), máximo, mínimo

QueuedDuration

Mide la duración de la fase `QUEUED` de la compilación.

Unidades: segundos

CloudWatch Estadísticas válidas: promedio (recomendado), máximo, mínimo

SubmittedDuration

Mide la duración de la fase `SUBMITTED` de la compilación.

Unidades: segundos

CloudWatch Estadísticas válidas: promedio (recomendado), máximo, mínimo

SucceededBuilds

Mide el número de compilaciones correctas.

Unidades: recuento

CloudWatch Estadísticas válidas: suma

UploadArtifactsDuration

Mide la duración de la fase `UPLOAD_ARTIFACTS` de la compilación.

Unidades: segundos

CloudWatch Estadísticas válidas: promedio (recomendado), máximo, mínimo

CodeBuild CloudWatch métricas de utilización de recursos

> **Note**
>
> CodeBuild las métricas de utilización de recursos solo están disponibles en las siguientes regiones:
>
> - 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 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)

Es posible realizar un seguimiento de las métricas de utilización de recursos siguientes. Para obtener más información sobre el uso de CloudWatch con CodeBuild, consulteEl monitor CodeBuild se construye con CloudWatch.

CPUUtilized

El número de unidades de CPU de procesamiento asignadas utilizadas por el contenedor de compilación.

Unidades: unidades de CPU

CloudWatch Estadísticas válidas: promedio (recomendado), máximo, mínimo

CPUUtilizedPercent

Porcentaje procesamiento asignado utilizado por el contenedor de compilación.

Unidad: porcentaje

CloudWatch Estadísticas válidas: promedio (recomendado), máximo, mínimo

MemoryUtilized

Número de megabytes de memoria utilizados por el contenedor de compilación.

Unidades: megabytes

CloudWatch Estadísticas válidas: promedio (recomendado), máximo, mínimo

MemoryUtilizedPercent

Porcentaje memoria asignada utilizada por el contenedor de compilación.

Unidad: porcentaje

CloudWatch Estadísticas válidas: promedio (recomendado), máximo, mínimo

StorageReadBytes

Velocidad de lectura de almacenamiento utilizada por el contenedor de compilación.

Unidades: Bytes/second

CloudWatch Estadísticas válidas: promedio (recomendado), máximo, mínimo

StorageWriteBytes

Velocidad de escritura de almacenamiento utilizada por el contenedor de compilación.

Unidades: Bytes/second

CloudWatch Estadísticas válidas: promedio (recomendado), máximo, mínimo

CodeBuild CloudWatch dimensiones

CodeBuild proporciona las siguientes dimensiones CloudWatch métricas. Si no se especifica ninguna de estas opciones, las métricas corresponden a la AWS cuenta 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.

CodeBuild CloudWatch alarmas

Puedes usar la CloudWatch consola para crear alarmas en función de CodeBuild las métricas, de modo que puedas reaccionar si algo sale mal con tus compilaciones. En las viñetas siguientes se

describen las dos métricas de mayor utilidad con las alarmas. Para obtener más información sobre el uso de CloudWatch con CodeBuild, consulteEl monitor CodeBuild se construye con CloudWatch.

- `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, usted especifica el número de segundos y el número de compilaciones fallidas que activan 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 información sobre cómo crear alarmas para CodeBuild las métricas, consulteEl monitor CodeBuild se construye con CloudWatch alarmas. Para obtener más información sobre las alarmas, consulte Creación de CloudWatch alarmas de Amazon en la Guía del CloudWatch usuario de Amazon.

Ver CodeBuild métricas

AWS CodeBuild supervisa las funciones en tu nombre e informa de 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.

Puedes usar la CodeBuild consola o la CloudWatch consola para monitorear las métricas CodeBuild. Los procedimientos siguientes muestran cómo ver las métricas.

Temas

- Ver las métricas de compilación (CodeBuild consola)
- Ver las métricas de compilación (CloudWatch consola de Amazon)

Ver las métricas de compilación (CodeBuild consola)

> **Note**
>
> No puede personalizar las métricas ni los gráficos que se utilizan para mostrarlos en la CodeBuild consola. Si quieres personalizar la visualización, usa la CloudWatch consola de Amazon para ver tus métricas de compilación.

Account-level métricas

Para ver las métricas a AWS nivel de cuenta

1. Inicie sesión en Consola de administración de AWS y abra la AWS CodeBuild consola en. https://console.aws.amazon.com/codesuite/codebuild/home
2. En el panel de navegación, seleccione Métricas de la cuenta.

Project-level métricas

Para ver las métricas de nivel de proyecto

1. Inicie sesión en Consola de administración de AWS y 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. 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).

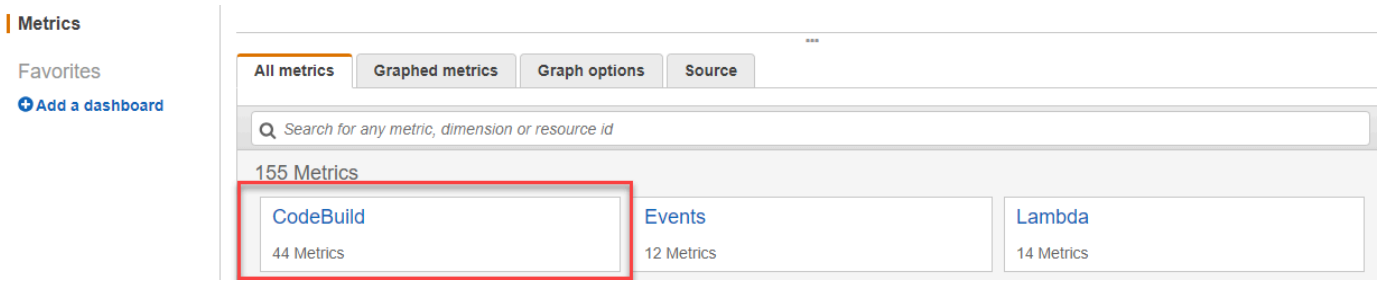
Ver las métricas de compilación (CloudWatch consola de Amazon)

Puede personalizar las métricas y los gráficos utilizados para mostrarlos con la CloudWatch consola.

Account-level métricas

Para ver las métricas de de cuenta

1. Inicie sesión en Consola de administración de AWS y abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, seleccione Métricas.
3. En la pestaña All metrics (Todas las métricas), elija CodeBuild.

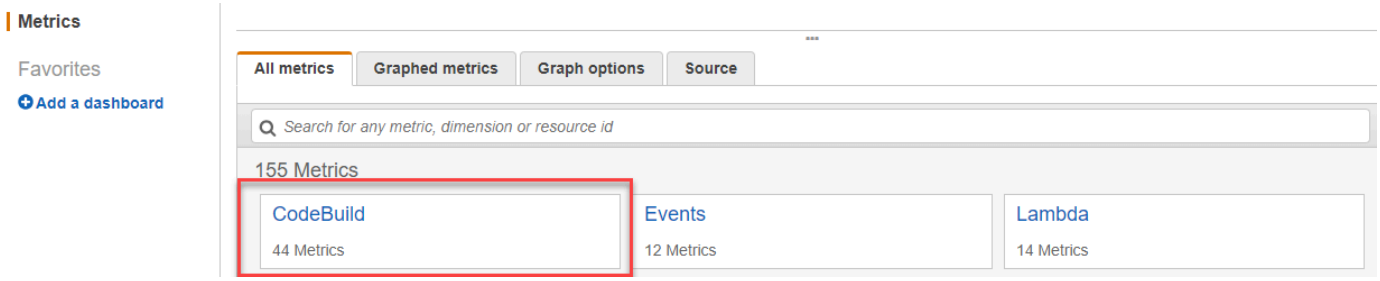


4. Elija Account Metrics (Métricas de cuenta).
5. Elija uno o varios proyectos y métricas. Para cada proyecto, puedes elegir las métricas SucceededBuildsFailedBuilds, las compilaciones y la duración. Todas las combinaciones de métricas y proyectos seleccionados se muestran en el gráfico de la página.

Project-level métricas

Para ver las métricas de nivel de proyecto

1. Inicie sesión en Consola de administración de AWS y abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, seleccione Métricas.
3. En la pestaña All metrics (Todas las métricas), elija CodeBuild.



4. Elija Por proyecto.
5. Elija uno o varios combinaciones de proyectos y métricas. Para cada proyecto, puedes elegir las métricas SucceededBuildsFailedBuilds, las compilaciones y la duración. 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, consulta Graph Metrics y Ver métricas disponibles en la Guía del CloudWatch usuario de Amazon.

Ver las métricas CodeBuild de utilización de recursos

AWS CodeBuild supervisa la utilización de los recursos de construcción en tu nombre e informa de las métricas a través de Amazon CloudWatch. Estas incluyen métricas como utilización de CPU, memoria y almacenamiento.

> **Note**
>
> CodeBuild Las métricas de uso de recursos solo se registran para las compilaciones que se ejecutan durante más de un minuto.

Puede utilizar la CodeBuild consola o la CloudWatch consola para supervisar las métricas de utilización de los recursos CodeBuild.

> **Note**
>
> CodeBuild las métricas de utilización de recursos solo están disponibles en las siguientes regiones:
>
> - 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 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

- Acceda a las métricas de utilización de los recursos (CodeBuild consola)

- Acceder a las métricas de utilización de los recursos (CloudWatch consola de Amazon)

Acceda a las métricas de utilización de los recursos (CodeBuild consola)

> **Note**
>
> No puede personalizar las métricas ni los gráficos que se utilizan para mostrarlos en la CodeBuild consola. Si quieres personalizar la visualización, usa la CloudWatch consola de Amazon para ver tus métricas de compilación.

Project-level métricas de utilización de recursos

Para acceder a las métricas de utilización de recursos en el nivel de proyecto

1. Inicie sesión en Consola de administración de AWS y 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. 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 recursos a nivel de proyecto en la CloudWatch consola, selecciona Ver en CloudWatch en la sección Métricas de utilización de recursos.

Build-level métricas de utilización de recursos

Para acceder a las métricas de utilización de recursos en el nivel de compilación

1. Inicie sesión en Consola de administración de AWS y abra la AWS CodeBuild consola 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 a nivel de compilación en la CloudWatch consola, selecciona Ver en CloudWatch en la sección Métricas de uso de recursos.

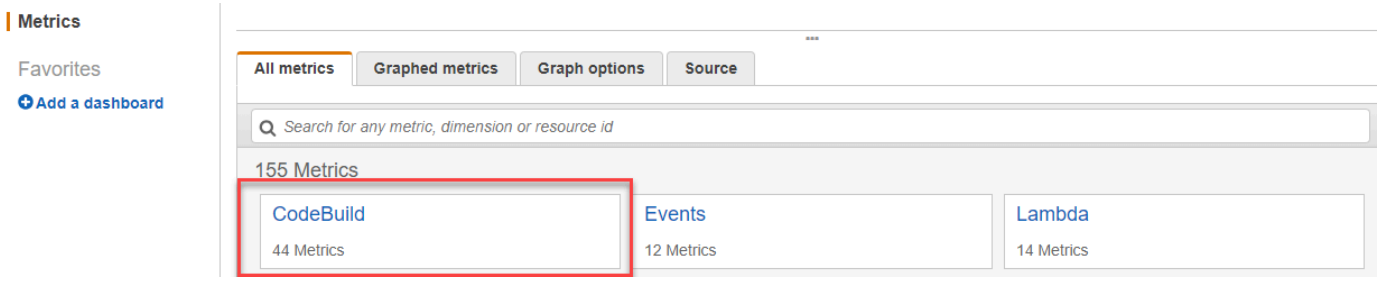
Acceder a las métricas de utilización de los recursos (CloudWatch consola de Amazon)

La CloudWatch consola de Amazon se puede utilizar para acceder a las métricas CodeBuild de utilización de los recursos.

Project-level métricas de utilización de recursos

Para acceder a las métricas de utilización de recursos en el nivel de proyecto

1. Inicie sesión en Consola de administración de AWS y abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, seleccione Métricas.
3. En la pestaña All metrics (Todas las métricas), elija CodeBuild.



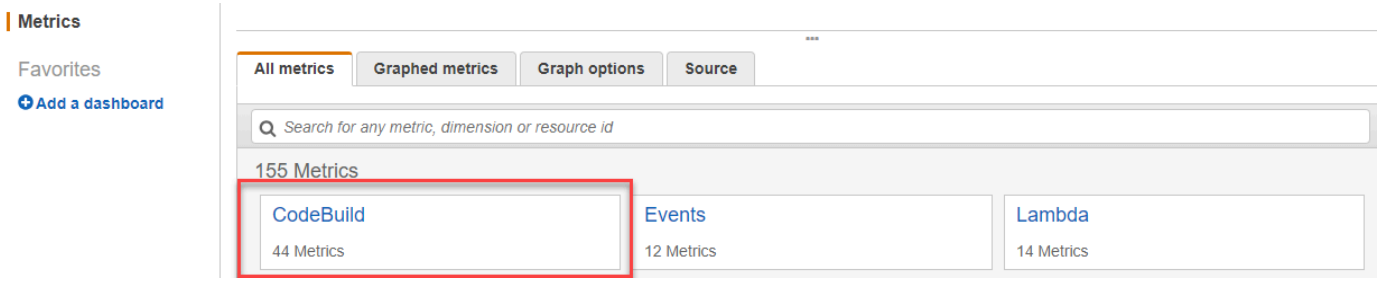
4. Elija Por proyecto.
5. Elija una o más combinaciones de proyectos y métricas para añadirlas al gráfico. Todas las combinaciones de métricas y proyectos seleccionados se muestran en el gráfico de la página.
6. (Opcional) Es posible personalizar las métricas y los gráficos desde la pestaña Métricas 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, consulta Cómo graficar las métricas y Ver las métricas disponibles en la Guía del CloudWatch usuario de Amazon.

Build-level métricas de utilización de recursos

Para acceder a las métricas de utilización de recursos en el nivel de compilación

1. Inicie sesión en Consola de administración de AWS y abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, seleccione Métricas.
3. En la pestaña All metrics (Todas las métricas), elija CodeBuild.



4. Elige 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, consulta Cómo graficar las métricas y Ver las métricas disponibles en la Guía del CloudWatch usuario de Amazon.

El monitor CodeBuild se construye con CloudWatch alarmas

Puede crear una CloudWatch alarma para sus edificios. 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 CloudWatch alarma nativa, puedes especificar cualquiera de las acciones que se admiten 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 CloudWatch alarma para una CodeBuild métrica

1. Inicie sesión en Consola de administración de AWS y abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Alarmas.
3. Elija Create Alarm (Crear alarma).
4. En CloudWatch Métricas por categoría, selecciona CodeBuildMétricas. 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, consulta Cómo crear CloudWatch alarmas de Amazon en la Guía del CloudWatch usuario de Amazon. 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. Seleccione Crear alarma.

Seguridad en AWS CodeBuild

La seguridad en la nube AWS es la máxima prioridad. Como AWS cliente, usted se beneficia de los centros de datos y las arquitecturas de red diseñados para cumplir con los requisitos de las organizaciones más sensibles a la seguridad.

La seguridad es una responsabilidad compartida entre AWS usted y usted. El modelo de responsabilidad compartida la describe como seguridad de la nube y seguridad en la nube:

- Seguridad de la nube: AWS es responsable de proteger la infraestructura que ejecuta AWS los servicios en la Nube de AWS. AWS también le proporciona servicios que puede utilizar de forma segura. Third-party los auditores prueban y verifican periódicamente la eficacia de nuestra seguridad como parte de los programas de AWS cumplimiento. Para obtener más información sobre los programas de cumplimiento aplicables AWS CodeBuild, consulte los AWS servicios incluidos en el ámbito de aplicación por programa de cumplimiento.
- Seguridad en la nube: su responsabilidad viene determinada por el AWS servicio que utilice. También es responsable de otros factores, incluida la confidencialidad de los datos, los requisitos de la empresa y la legislación y los reglamentos vigentes.

Esta documentación le ayuda a comprender cómo aplicar el modelo de responsabilidad compartida cuando se utiliza CodeBuild. Los siguientes temas muestran cómo configurarlo CodeBuild para cumplir sus objetivos de seguridad y conformidad. También aprenderá a utilizar otros AWS servicios que le ayudan a supervisar y proteger sus CodeBuild recursos.

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 fuentes en CodeBuild
- Cross-service prevención confusa de diputados

Protección de datos en AWS CodeBuild

El modelo de responsabilidad compartida de AWS se aplica a la protección de datos en AWS CodeBuild. Como se describe en este modelo, AWS es responsable de proteger la infraestructura global en la que se ejecutan todos los Nube de AWS. Eres responsable de mantener el control sobre el contenido alojado en esta infraestructura. También eres responsable de las tareas de administración y configuración de seguridad para los Servicios de AWS que utiliza. Para obtener más información sobre la privacidad de los datos, consulte las Preguntas frecuentes sobre privacidad de datos y los . Para obtener más información sobre la protección de datos en Europa, consulte el Centro del Reglamento General de Protección de Datos (RGPD).

Para fines de protección de datos, le recomendamos que proteja Cuenta de AWS las credenciales y configure 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:

- Utiliza la autenticación multifactor (MFA) en cada cuenta.
- Se utiliza SSL/TLS para comunicarse con AWS los recursos. Exigimos TLS 1.2 y recomendamos TLS 1.3.
- Configure la API y el registro de actividad de los usuarios con AWS CloudTrail. Para obtener información sobre el uso de CloudTrail senderos para capturar AWS actividades, consulte Cómo trabajar con CloudTrail senderos en la Guía del AWS CloudTrail usuario.
- Utilice soluciones de AWS cifrado, junto con todos los controles de seguridad predeterminados que contienen Servicios de AWS.
- Utiliza servicios de seguridad administrados avanzados, como Amazon Macie, que lo ayuden a detectar y proteger la información confidencial almacenada en Amazon S3.
- Si necesita módulos criptográficos validados por FIPS 140-3 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) 140-3.

Se recomienda encarecidamente no introducir nunca información confidencial o sensible, como por ejemplo, direcciones de correo electrónico de clientes, en etiquetas o campos de formato libre, tales como el campo Nombre. Esto incluye cuando trabaja CodeBuild o Servicios de AWS utiliza la consola, la API o los SDK. AWS CLI AWS Cualquier dato que introduzca en etiquetas o campos

de formato libre utilizados para los nombres se pueden emplear para los registros de facturación o diagnóstico. Si proporciona una URL a un servidor externo, recomendamos encarecidamente que no incluya información de credenciales en la URL a fin de validar la solicitud para ese servidor.

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 buildspec`env/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.

> **Note**
>
> El enmascaramiento coincide con el valor exacto almacenado en el almacén de parámetros de Amazon EC2 Systems Manager o. AWS Secrets Manager Si un comando de compilación transforma un secreto antes de que aparezca en los registros, el resultado es una cadena diferente que no está enmascarada. Por ejemplo, si se codifica con Base64, se invierte, se cambia de mayúsculas y minúsculas o se concatena con otro texto, se crea un valor diferente que puede aparecer en los registros de compilación.
> Para mantener un valor enmascarado, guárdelo en el formato que utiliza su compilación. Por ejemplo, si la compilación requiere un Base64-encoded valor, almacene la Base64-encoded cadena directamente en AWS Secrets Manager el almacén de parámetros de Amazon EC2 Systems Manager. De esta forma, el valor registrado coincide con el valor almacenado.

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 CodeBuild seguridad. 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 de 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 de 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 SHA-256 certificados gestionados por. AWS Autoridad de certificación privada 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 Cifrado de las salidas de compilación 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 Cifrado de las salidas de compilación con una clave administrada por el cliente, Creación de un proyecto de compilación en AWS CodeBuild, Ejecute AWS CodeBuild compilaciones manualmente 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 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 PrivateLinkAWSlos servicios mediante. PrivateLink

Administración de identidades y accesos en AWS CodeBuild

El acceso a 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 los CloudWatch registros de Amazon Logs para compilaciones. En las siguientes secciones, se describe cómo puede utilizar AWS Identity and Access Management(IAM) y cómo ayudar CodeBuild a proteger el acceso a sus recursos:

Descripción general de la administración de los permisos de acceso a su AWS CodeBuild recursos

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	`arn:aws:codebuild:` *`region-ID`* `:`*`account-ID`* `:project/`*`project-name`*

Tipo de recurso	Formato de ARN
Build	`arn:aws:codebuild:` *region-ID* `:` *account-ID* `:build/` *build-ID*
Grupo de informes	`arn:aws:codebuild:` *region-ID* `:` *account-ID* `:report-group/` *report-group-name*
Informar	`arn:aws:codebuild:` *region-ID* `:` *account-ID* `:report/` *report-ID*
Fleet	`arn:aws:codebuild:` *region-ID* `:` *account-ID* `:fleet/` *fleet-ID*
CodeBuild Todos los recursos	`arn:aws:codebuild:*`
Todos CodeBuild los recursos que pertenecen a la cuenta especificada en la AWS región especificada	`arn:aws:codebuild:` *region-ID* `:` *account-ID* `:*`

Important

Cuando se usa la característica de capacidad reservada, otros proyectos de la misma cuenta pueden acceder a los datos almacenados en caché en las instancias de flota (incluidos los archivos de origen, las capas de Docker y los directorios almacenados en caché que se indican en la especificación de compilación). Esto es así por diseño y permite que los proyectos de la misma cuenta compartan instancias de flotas.

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, tu AWS cuenta, a la que pertenece el usuario, es la 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:GetBucketAcl`
- `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 Permita a los usuarios interactuar con CodeBuild y CodeBuild Permitir interactuar con otros AWS servicios.

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 laAWS 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 su AWS CodeBuild recursos.

Temas

- Permisos necesarios para usar el AWS CodeBuild consola
- Se requieren permisos para AWS CodeBuild para conectarse a Amazon Elastic Container Registry
- Permisos necesarios para AWS CodeBuild consola para conectarse a los proveedores de fuentes
- AWS políticas gestionadas (predefinidas) para AWS CodeBuild
- CodeBuild políticas y notificaciones gestionadas
- CodeBuild actualizaciones de AWS políticas administradas
- Customer-managed ejemplos de políticas

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`:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:BatchGetProjects",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333:project/my*"
    }
  ]
}
```

Permisos necesarios para usar el AWS CodeBuild consola

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 Se requieren permisos para AWS CodeBuild para conectarse 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.

Se requieren permisos para AWS CodeBuild para conectarse 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 Permita a los usuarios interactuar con CodeBuild.

Permisos necesarios para AWS CodeBuild consola para conectarse a los proveedores de fuentes

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 puede crear sus propias políticas de IAM personalizadas para conceder permisos a las acciones y recursos de CodeBuild. 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.

Para ver la política gestionada completa, consulte la referencia de políticas AWSCodeBuildAdminAccessgestionadas de IAM.

AWSCodeBuildDeveloperAccess

La `AWSCodeBuildDeveloperAccess` política permite el acceso a todas las funciones de los recursos relacionados con los grupos de proyectos CodeBuild e informes, así como a los recursos relacionados con ellos. 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.

Para ver la política gestionada completa, consulte AWSCodeBuildDeveloperAccessla referencia de políticas gestionadas de IAM.

AWSCodeBuildReadOnlyAccess

La `AWSCodeBuildReadOnlyAccess` política otorga acceso de solo lectura a los recursos relacionados de otros servicios CodeBuild y a los recursos relacionados. AWS 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.

Para ver la política gestionada completa, consulte la referencia AWSCodeBuildReadOnlyAccessde políticas gestionadas de IAM.

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 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" : {
           "ArnLike" : {"codestar-notifications:NotificationsForResource" :
"arn:aws:codebuild:*:*:project/*"}
       }
   },
   {
       "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" : {
            "ArnLike" : {"codestar-notifications:NotificationsForResource" :
"arn:aws:codebuild:*:*:project/*"}
        }
    },
    {
        "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 AWS políticas administradas

Vea 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, suscríbase a la fuente RSS en AWS CodeBuild Historial de documentos de la Guía del usuario.

Cambio	Descripción	Fecha
`AWSCodeBuildAdminAccess` y `AWSCodeBuildDeveloperAccess` : actualización de políticas existentes	CodeBuild agregó el `ssmmessages:OpenDataChannel` permiso a estas políticas para admitir la depuración interactiva de compilaciones del Session Manager. Las `AWSCodeBuildDeveloperAccess` políticas `AWSCodeBuildAdminAccess` y ahora incluyen la `ssmmessages:OpenDataChannel` acción para que los recursos de sesión del administrador de sesiones (`arn:aws:ssm:*:*:session/*`) respalden la aplicación de SiGv4 en esta WebSocket API.	1 de diciembre de 2025
`AWSCodeBuildAdminAccess` , `AWSCodeBuildDeveloperAccess` y `AWSCodeBuildReadOnlyAccess` : actualización de políticas existentes	CodeBuild actualizó un recurso sobre estas políticas. Las políticas `AWSCodeBuildAdminAccess` , `AWSCodeBuildDeveloperAccess` y `AWSCodeBu`	15 de noviembre de 2024

Cambio	Descripción	Fecha
	`ildReadOnlyAccess` se han modificado para actualizar un recurso existente. El recurso original `arn:aws:codebuild:*` se ha actualizado a `arn:aws:codebuild:*:*:project/*` .	
`AWSCodeBuildAdminAccess` , `AWSCodeBuildDeveloperAccess` y `AWSCodeBuildReadOnlyAccess` : actualización de políticas existentes	CodeBuild agregó un recurso a estas políticas para respaldar el cambio de AWS CodeConnections marca. Las políticas `AWSCodeBuildAdminAccess` , `AWSCodeBuildDeveloperAccess` y `AWSCodeBuildReadOnlyAccess` se han modificado para agregar un recurso, `arn:aws:codeconnections:*:*:*` .	18 de abril de 2024

Cambio	Descripción	Fecha
`AWSCodeBuildAdminAccess` y `AWSCodeBuildDeveloperAccess` : actualización de políticas existentes	CodeBuild agregó un permiso a estas políticas para admitir un tipo de notificación adicional utilizando Amazon Q Developer en aplicaciones de chat. Las políticas de `AWSCodeBuildAdminAccess` y `AWSCodeBuildDeveloperAccess` se han modificado para añadir un permiso, `chatbot:ListMicrosoftTeamsChannelConfigurations` .	16 de mayo de 2023
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

Customer-managed ejemplos de políticas

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 información, consulte Permisos necesarios para usar el AWS CodeBuild consola.

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

- Cómo permitir a un usuario obtener información sobre las flotas
- 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
- Cómo permitir a un usuario crear una flota
- Permitir que un usuario pueda crear un grupo de informes
- Cómo permitir a un usuario eliminar una flota
- 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
- Cómo permitir a un usuario cambiar una flota
- 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
- Cómo permitir a un usuario obtener una lista de flotas
- 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
- Cómo permitir a un usuario agregar una política de permisos para un rol de servicio de flota

- Permitir el CodeBuild acceso a AWS servicios necesarios para crear una interfaz de red de VPC
- Utilice una sentencia de denegación para evitar AWS CodeBuild de desconectarse 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`:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:BatchGetProjects",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333:project/my*"
    }
  ]
}
```

Cómo permitir a un usuario obtener información sobre las flotas

El ejemplo siguiente de declaración de política permite a un usuario obtener información sobre las flotas en la región `us-east-2` para la cuenta `123456789012`:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:BatchGetFleets",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333:fleet/*"
    }
```

```
  ]
}
```

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`:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:BatchGetReportGroups",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333: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`:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:BatchGetReports",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333: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:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:CreateProject",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333:project/*"
    },
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::111122223333: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.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:CreateProject",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333:project/*"
    },
    {
```

```
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::111122223333:role/CodeBuildServiceRole",
      "Condition": {
          "StringEquals": {"iam:PassedToService": "codebuild.amazonaws.com"}
      }
    }
  ]
}
```

Cómo permitir a un usuario crear una flota

El ejemplo siguiente de declaración de política permite a un usuario crear una flota en la región `us-east-2` para la cuenta `123456789012`:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:CreateFleet",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333:fleet/*"
    }
  ]
}
```

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`:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
```

```
      "Effect": "Allow",
      "Action": "codebuild:CreateReportGroup",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333:report-group/*"
    }
  ]
}
```

Cómo permitir a un usuario eliminar una flota

El ejemplo siguiente de declaración de política permite a un usuario eliminar una flota en la región `us-east-2` para la cuenta `123456789012`:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:DeleteFleet",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333:fleet/*"
    }
  ]
}
```

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`:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:DeleteReportGroup",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333: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`:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:DeleteReport",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333: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`:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:DeleteProject",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333: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:

JSON

```
{
  "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:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:UpdateProject",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333:project/*"
    },
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",
```

```
      "Resource": "arn:aws:iam::111122223333:role/CodeBuildServiceRole"
    }
  ]
}
```

Cómo permitir a un usuario cambiar una flota

El ejemplo siguiente de declaración de política permite a un usuario cambiar una flota en la región `us-east-2` para la cuenta `123456789012`:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:UpdateFleet",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333:fleet/*"
    }
  ]
}
```

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`:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:UpdateReportGroup",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333: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`:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:BatchGetBuilds",
      "Resource": [
        "arn:aws:codebuild:us-east-2:111122223333:project/my-build-project",
        "arn:aws:codebuild:us-east-2:111122223333: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`:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:ListBuildsForProject",
```

```
      "Resource": [
        "arn:aws:codebuild:us-east-2:111122223333:project/my-build-project",
        "arn:aws:codebuild:us-east-2:111122223333: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:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:ListBuilds",
      "Resource": "*"
    }
  ]
}
```

Cómo permitir a un usuario obtener una lista de flotas

El ejemplo siguiente de declaración de política permite a un usuario obtener una lista de las flotas en la región `us-east-2` para la cuenta `123456789012`:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:ListFleets",
```

```
      "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`:

JSON

```
{
  "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`:

JSON

```
{
  "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`:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:ListReportsForReportGroup",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333: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`:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:DescribeTestCases",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333: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`:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:StartBuild",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333: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`:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:StopBuild",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333: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`:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:BatchDeleteBuilds",
      "Resource": "arn:aws:codebuild:us-east-2:111122223333:project/my*"
    }
  ]
}
```

Permite a un usuario obtener información sobre las imágenes de Docker administradas por CodeBuild

La siguiente instrucción de política de ejemplo permite a un usuario obtener información acerca de todas las imágenes de Docker administradas por CodeBuild:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "codebuild:ListCuratedEnvironmentImages",
      "Resource": "*"
    }
  ]
}
```

Cómo permitir a un usuario agregar una política de permisos para un rol de servicio de flota

El ejemplo siguiente de declaración de política de recursos permite a un usuario agregar una política de permisos de VPC para un rol de servicio de flota:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "CodeBuildFleetVpcCreateNI",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateNetworkInterface"
            ],
            "Resource": [
                "arn:aws:ec2:us-west-2:111122223333:subnet/subnet-id-1",
                "arn:aws:ec2:us-west-2:111122223333:security-group/security-
group-id-1",
                "arn:aws:ec2:us-west-2:111122223333:network-interface/*"
            ]
        },
        {
            "Sid": "CodeBuildFleetVpcPermission",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeDhcpOptions",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcs",
                "ec2:ModifyNetworkInterfaceAttribute",
                "ec2:DeleteNetworkInterface"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CodeBuildFleetVpcNIPermission",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateNetworkInterfacePermission"
            ],
```

```
            "Resource": "arn:aws:ec2:us-west-2:111122223333:network-interface/*",
            "Condition": {
                "StringEquals": {
                    "ec2:Subnet": [
                        "arn:aws:ec2:us-west-2:111122223333:subnet/subnet-id-1"
                    ]
                }
            }
        }
    ]
}
```

El ejemplo siguiente de declaración de política de recursos permite a un usuario agregar una política de Imágenes gestionadas por Amazon (AMI) para un rol de servicio de flota:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:DescribeImages",
            "Resource": "*"
        }
    ]
}
```

El ejemplo siguiente de declaración de política de confianza permite a un usuario agregar una política de permisos para un rol de servicio de flota:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "CodeBuildFleetVPCTrustPolicy",
      "Effect": "Allow",
```

```
      "Principal": {
        "Service": "codebuild.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "111122223333"
        }
      }
    }
  ]
}
```

Permitir el CodeBuild acceso a AWS 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:

JSON

```
{
    "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:us-west-2:111122223333:network-interface/*",
```

```
            "Condition": {
                "StringEquals": {
                    "ec2:AuthorizedService": "codebuild.amazonaws.com"
                },
                "ArnEquals": {
                    "ec2:Subnet": [
                        "arn:aws:ec2:us-west-2:111122223333:subnet/subnet-id-1",
                        "arn:aws:ec2:us-west-2:111122223333:subnet/subnet-id-2"
                    ]
                }
            }
        }
    ]
}
```

Utilice una sentencia de denegación para evitar AWS CodeBuild de desconectarse 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:DeleteOAuthToken`, que es lo contrario de `codebuild:PersistOAuthToken` y `codebuild:ImportSourceCredentials`, para establecer conexión con los proveedores de código fuente. Para obtener más información, consulte Permisos necesarios para AWS CodeBuild consola para conectarse a los proveedores de fuentes.

JSON

```
{
  "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`

BatchPutTestCases[1]

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`

CreateReport[1]

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`*

UpdateReport[1]

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 AWS CodeBuild recursos

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.

> **Important**
>
> Cuando se usa la característica de capacidad reservada, otros proyectos de la misma cuenta pueden acceder a los datos almacenados en caché en las instancias de flota (incluidos los archivos de origen, las capas de Docker y los directorios almacenados en caché que se indican en la especificación de compilación). Esto es así por diseño y permite que los proyectos de la misma cuenta compartan instancias de flotas.

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.

JSON

```
{
  "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).

JSON

```
{
  "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:StringEqualsIgnoreCase": {
          "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 y`Secret`, a continuación, aplicar esa política al grupo de IAM que creó para desarrolladores generales ()*`Developers`*. A continuación, debe asegurarse de que los desarrolladores que trabajan en esos grupos de informes etiquetados no sean miembros de ese *`Developers`* grupo general, sino que pertenezcan a un grupo de IAM diferente al que no se haya aplicado la política restrictiva ()`SecretDevelopers`.

El siguiente ejemplo deniega todas CodeBuild las acciones en los grupos de informes etiquetados con la clave `Status` y el valor clave de`Secret`:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "codebuild:BatchGetReportGroups",
        "codebuild:CreateReportGroup",
        "codebuild:DeleteReportGroup",
        "codebuild:ListReportGroups",
        "codebuild:ListReportsForReportGroup",
        "codebuild:UpdateReportGroup"
      ],
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "aws:RequestedRegion": "us-east-1"
        }
      }
    }
  ]
}
```

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:

JSON

```
{
   "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 necesita el `ListRepositories` permiso para mostrar una lista de los repositorios de su AWS cuenta en la AWS región en la que ha 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

Nuestra nueva experiencia AWS de registro no está diseñada para cargas de trabajo reguladas. Si utilizas nuestra nueva experiencia de AWS registro, pero quieres utilizarla AWS para cargas de trabajo reguladas, puedes suscribirte a una AWS (avanzada) o activar funciones avanzadas para tu entorno. AWS

Para saber si un programa de cumplimiento Servicio de AWS se encuentra dentro del ámbito de aplicación de programas de cumplimiento específicos, consulte Servicios de AWS Alcance por programa de cumplimiento Servicios de AWS y elija el programa de cumplimiento que le interese. Para obtener información general, consulte Programas de AWS cumplimiento > Programas AWS .

Puede descargar informes de auditoría de terceros utilizando AWS Artifact. Para obtener más información, consulte Descarga de informes en AWS Artifact .

Su responsabilidad de cumplimiento al Servicios de AWS utilizarlos viene determinada por la confidencialidad de sus datos, los objetivos de cumplimiento de su empresa y las leyes y reglamentos aplicables. Para obtener más información sobre su responsabilidad de conformidad al utilizarlos Servicios de AWS, consulte AWS la documentación de seguridad.

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.
- Paquetes de cifrado con perfecto secreto directo (PFS), como el DHE (Ephemeral) o el ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Diffie-Hellman La mayoría de los sistemas modernos como Java 7 y posteriores son compatibles con estos modos.

Acceda a su proveedor de fuentes en CodeBuild

Para GitHub GitHub Enterprise Server, se utiliza un token de acceso personal, un secreto de Secrets Manager, una conexión o una aplicación OAuth para acceder al proveedor de origen. En Bitbucket, utilizas un token de acceso, una contraseña de aplicación, un token de API, un secreto de Secrets Manager, una conexión o una aplicación OAuth para acceder al proveedor de origen.

Temas

- Creación y almacenamiento de un token en un secreto de Secrets Manager
- GitHub y acceso a GitHub Enterprise Server en CodeBuild
- Acceda a Bitbucket en CodeBuild
- GitLab acceder en CodeBuild

Creación y almacenamiento de un token en un secreto de Secrets Manager

Si decides almacenar tu token de acceso mediante Secrets Manager, puedes usar una conexión secreta existente o crear una nueva. Para crear un secreto, haga lo siguiente:

Consola de administración de AWS

Para crear un secreto de Secrets Manager en el Consola de administración de AWS

1. En el caso del proveedor de código fuente GitHub, selecciona Bitbucket o GitHub Enterprise.
2. En Credencial, realice una de las siguientes acciones:
   - Elija Credencial de origen predeterminada para aplicar la credencial de origen predeterminada de su cuenta a todos los proyectos.
     a. Si no está conectado a su proveedor de origen, elija Administrar la credencial de origen predeterminada.
     b. Para el tipo de credencial, elige un tipo de credencial que no sea. AWS CodeConnections
     c. En Servicio, elija Secrets Manager y para Secretos elija Nuevo secreto.
     d. En Nombre del secreto, especifique el nombre de su secreto.
     e. En Descripción del secreto (opcional), escriba una descripción para su secreto.
     f. Según el proveedor de origen que elija, introduzca su token o nombre de usuario y la contraseña de la aplicación y, a continuación, seleccione Guardar. Para un token de API de Bitbucket, introduce la dirección de correo electrónico y el token de API de tu cuenta de Atlassian en su lugar.
   - Elija Credencial de origen personalizada a fin de usar una credencial de este tipo para anular la configuración predeterminada de su cuenta.
     a. Para el tipo de credencial, elige un tipo de credencial que no sea. AWS CodeConnections
     b. En Conexión, elija Crear un secreto.

c. En Nombre del secreto, especifique el nombre de su secreto.

d. En Descripción del secreto (opcional), escriba una descripción para su secreto.

e. En función del proveedor de origen que haya elegido, especifique su token o nombre de usuario y la contraseña de aplicación y elija Crear. Para un token de API de Bitbucket, introduce la dirección de correo electrónico de tu cuenta de Atlassian y el token de API en su lugar.

AWS CLI

Para crear un secreto de Secrets Manager en el AWS CLI

- Abra un terminal (Linux, macOS o Unix) o un símbolo del sistema (Windows). Utilice el AWS CLI para ejecutar el create-secret comando Secrets Manager.

```
aws secretsmanager create-secret --region <aws-region> \
            --name '<secret-name>' \
            --description '<secret-description>' \
            --secret-string '{
                "ServerType":"<server-type>",
                "AuthType":"<auth-type>",
                "Token":"<token>"
                }' \
            --tags Key=codebuild:source,Value='' \
                Key=codebuild:source:type,Value=<type> \
                Key=codebuild:source:provider,Value=<provider>
```

Los secretos de Secrets Manager que CodeBuild acepta deben estar en la misma cuenta y AWS región que el CodeBuild proyecto y deben estar en el siguiente formato JSON:

```
{
            "ServerType": ServerType,
            "AuthType: AuthType,
            "Token": string,
            "Username": string // Optional and is only used for Bitbucket app
 password or API token
       }
```

Campo	Valores válidos	Description (Descripción)
ServerType	GITHUB GITHUB_ENTERPRISE BITBUCKET	El proveedor de origen externo para su secreto de Secrets Manager.
AuthType	PERSONAL_ACCESS_TOKEN BASIC_AUTH	El tipo de token de acceso que utilizan las credenciales. Para ello GitHub, solo PERSONAL_ACCESS_TOKEN es válido. BASIC_AUTH solo es válido para la contraseña de la aplicación Bitbucket o el token de API.
Token	*string*	Para GitHub GitHub empresas, este es el token de acceso personal. En el caso de Bitbucket, se trata del token de acceso, la contraseña de la aplicación de Bitbucket o el token de la API de Bitbucket.
Nombre de usuario	*string*	El nombre de usuario de Bitbucket cuando es BASIC_AUTH. AuthType Si utilizas un token de API de Bitbucket, esta es la dirección de correo electrónico asociada a tu cuenta de Atlassian. Este parámetro no es válido para otros tipos de proveedores de origen.

Además, CodeBuild utiliza las siguientes etiquetas de recursos en el secreto para garantizar que los secretos se puedan seleccionar fácilmente al crear o editar proyectos.

Clave de etiqueta	Valor de etiqueta	Description (Descripción)
codebuild:source:provider	github github_enterprise bitbucket	Indica a CodeBuild qué proveedor está destinado este secreto.
codebuild:source:type	personal_access_token basic_auth	Indica CodeBuild el tipo de token de acceso de este secreto.

GitHub y acceso a GitHub Enterprise Server en CodeBuild

Para GitHub ello, puedes usar un token de acceso personal, una aplicación OAuth, un secreto de Secrets Manager o una conexión de GitHub aplicación para acceder al proveedor de origen. Para GitHub Enterprise Server, puedes usar un token de acceso personal, un secreto de Secrets Manager o una conexión de GitHub aplicación para acceder al proveedor de origen.

Temas

- GitHub Conexiones de aplicaciones para GitHub un servidor GitHub empresarial
- GitHub y un token de acceso a GitHub Enterprise Server
- GitHub Aplicación OAuth

GitHub Conexiones de aplicaciones para GitHub un servidor GitHub empresarial

Puede usar GitHub la aplicación para conectarse con CodeBuild. GitHub Las conexiones de aplicaciones son compatibles a través de AWS CodeConnections.

El acceso del proveedor de origen te permite activar una compilación suscribiéndote al GitHub eventos de webhook uso CreateWebhooko al uso Tutorial: Configurar un ejecutor de CodeBuild-hosted GitHub acciones en CodeBuild.

> **Note**
>
> AWS CodeConnections está disponible en menos regiones que. CodeBuild Puede utilizar conexiones entre regiones en CodeBuild. Las conexiones creadas en regiones en las que se ha optado por participar no se pueden utilizar en otras regiones. Para obtener más información, consulte Puntos de conexión y cuotas de AWS CodeConnections.

Temas

- Paso 1: Crear una conexión a la GitHub aplicación (consola)
- Paso 2: Otorgue acceso al rol de IAM del CodeBuild proyecto para usar la conexión
- Paso 3: CodeBuild Configúralo para usar la nueva conexión
- Solución de problemas con la GitHub aplicación

Paso 1: Crear una conexión a la GitHub aplicación (consola)

Sigue estos pasos para usar la CodeBuild consola y añadir una conexión a tu proyecto GitHub.

Para crear una conexión a GitHub

- Siga las instrucciones de la Guía del usuario de las herramientas para desarrolladores para crear una conexión a GitHub.

> **Note**
>
> En lugar de crear o usar una conexión existente en tu cuenta, puedes usar una conexión compartida desde otra AWS cuenta. Para obtener más información, consulta Compartir conexiones con AWS cuentas.

Paso 2: Otorgue acceso al rol de IAM del CodeBuild proyecto para usar la conexión

Puedes conceder acceso a la función de IAM del CodeBuild proyecto para usar los GitHub tokens que vende tu conexión.

Para conceder acceso a la función de CodeBuild IAM del proyecto

1. Cree un rol de IAM para su CodeBuild proyecto siguiendo las instrucciones CodeBuild Permitir interactuar con otros AWS servicios para su CodeBuild proyecto.
2. Siguiendo las instrucciones, añade la siguiente política de IAM a tu rol en el CodeBuild proyecto para conceder el acceso a la conexión.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "codeconnections:GetConnectionToken",
                "codeconnections:GetConnection"
            ],
            "Resource": [
                "arn:aws:iam::*:role/Service*"
            ]
        }
    ]
}
```

Paso 3: CodeBuild Configúralo para usar la nueva conexión

Puede configurar una conexión como credencial de cuenta y usarla en un proyecto.

Consola de administración de AWS

Para configurar una conexión como una credencial a nivel de cuenta en el Consola de administración de AWS

1. En Proveedor de origen, elija GitHub.
2. En Credencial, realice una de las siguientes acciones:
   - Elija Credencial de origen predeterminada para aplicar la credencial de origen predeterminada de su cuenta a todos los proyectos.

a. Si no está conectado a GitHub, elija Administrar la credencial de origen predeterminada.

b. En Tipo de credencial, selecciona GitHub Aplicación.

c. En Conexión, elija usar una conexión existente o cree una nueva.

- Elija Credencial de origen personalizada a fin de usar una credencial de este tipo para anular la configuración predeterminada de su cuenta.

a. En Tipo de credencial, elija GitHub Aplicación.

b. En Conexión, elija usar una conexión existente o cree una nueva.

AWS CLI

Para configurar una conexión como una credencial a nivel de cuenta en el AWS CLI

- Abra un terminal (Linux, macOS o Unix) o un símbolo del sistema (Windows). Utilice el AWS CLI para ejecutar el import-source-credentials comando, especificando el `--auth-type`-- `server-type`, y `--token` para la conexión.

  Utilice el siguiente comando :

```
aws codebuild import-source-credentials --auth-type CODECONNECTIONS --server-
type GITHUB --token <connection-arn>
```

También puedes configurar varios tokens para tus CodeBuild proyectos. Para obtener más información, consulte Configuración de varios tokens como credenciales de origen.

Solución de problemas con la GitHub aplicación

La siguiente información puede ayudarle a solucionar problemas comunes con la GitHub aplicación.

Temas

- Instala la AWS Conector para una GitHub aplicación en una región no deseada
- La conexión de la GitHub aplicación no tiene acceso a los repositorios
- La AWS al rol de IAM del servicio le faltan los permisos de IAM necesarios.

Instala la AWS Conector para una GitHub aplicación en una región no deseada

Problema: instaló el AWS conector GitHub desde GitHub Marketplace, pero la conexión se creó en una región no deseada. Si intentas volver a configurar la aplicación en el GitHub sitio web, no funcionará porque la aplicación ya está instalada en tu GitHub cuenta.

Causa posible: la aplicación ya está instalada en tu GitHub cuenta, por lo que solo puedes volver a configurar los permisos de la aplicación.

Solución recomendada: Puede crear una conexión con el ID de instalación en la región deseada.

1. Abre la AWS CodeConnections consola en https://console.aws.amazon.com/codesuite/settings/connectionsy navega hasta la región deseada con el selector de regiones de la barra de navegación de la AWS consola.
2. Siga las instrucciones de la Guía del usuario de las herramientas para desarrolladores para crear una conexión a GitHub.

> **Note**
>
> Como ya ha instalado el AWS conector para la GitHub aplicación, puede elegirlo en lugar de instalar una aplicación nueva.

La conexión de la GitHub aplicación no tiene acceso a los repositorios

Problema: un AWS servicio que utiliza la conexión, como CodeBuild o CodePipeline, informa que no tiene acceso al repositorio o que el repositorio no existe. Estos son algunos de los posibles mensajes de error:

- `Authentication required for primary source.`
- `Unable to create webhook at this time. Please try again later.`
- `Failed to create webhook. GitHub API limit reached. Please try again later.`

Causa posible: es posible que hayas estado usando la GitHub aplicación y no hayas concedido el alcance de los permisos del webhook.

Solución recomendada: para conceder el alcance del permiso requerido, sigue las instrucciones de Cómo ir a la GitHub aplicación que quieras revisar o modificar para configurar la aplicación

instalada. En la sección de permisos, verá que la aplicación no tiene permisos de webhook y que hay una opción para revisar los permisos recién solicitados. Revise y acepte los nuevos permisos. Para obtener más información, consulte Aprobar los permisos actualizados para una GitHub aplicación.

Causa posible: la conexión funcionaba según lo previsto, pero de repente no tiene acceso a los repositorios.

Posible solución: comience por revisar sus autorizaciones y sus instalaciones y, a continuación, compruebe que la GitHub aplicación esté autorizada e instalada. Si la instalación de la GitHub aplicación está suspendida, tendrá que anular la suspensión. Si la GitHub aplicación no está autorizada para una conexión UAT (token de acceso de usuario) o no está instalada para una conexión IAT (token de acceso a la instalación), la conexión existente ya no se podrá utilizar y tendrá que crear una nueva conexión. Tenga en cuenta que volver a instalar la GitHub aplicación no restablecerá la conexión anterior que estaba asociada a la instalación anterior.

Solución posible: si la conexión es una conexión UAT, asegúrese de que la conexión no se utilice simultáneamente, por ejemplo, en varias ejecuciones CodeBuild simultáneas de compilación. Esto se debe a que invalida GitHub inmediatamente un UAT emitido anteriormente si la conexión actualiza un token que está caducando. Si necesitas usar una conexión UAT para varias CodeBuild compilaciones simultáneas, puedes crear varias conexiones y usar cada conexión de forma independiente.

Posible solución: si la conexión UAT no se ha utilizado en los últimos 6 meses, la conexión quedará invalidada por. GitHub Para solucionarlo, cree una conexión nueva.

Causa posible: Puede que haya estado usando una conexión de token de acceso de usuario sin instalar la aplicación.

Solución recomendada: aunque para crear una conexión UAT no es necesario asociar la conexión a una instalación de la GitHub aplicación, es necesaria una instalación para poder acceder al repositorio. Sigue las instrucciones para revisar las instalaciones y asegurarte de que la GitHub aplicación esté instalada. Si no está instalada, diríjase a la GitHub página de la aplicación para instalarla. Para obtener más información sobre el acceso de los tokens del tipo mencionado, consulte Acerca de los tokens de acceso de usuario.

La AWS al rol de IAM del servicio le faltan los permisos de IAM necesarios.

Problema: Aparece alguno de los siguientes mensajes de error:

- `Access denied to connection <connection-arn>`
- `Failed to get access token from <connection-arn>`

Solución recomendada: normalmente se utiliza una conexión con un AWS servicio, como CodePipeline o. CodeBuild Cuando asignas al AWS servicio una función de IAM, el AWS servicio puede usar el permiso de la función para actuar en tu nombre. Asegúrese de que el rol de IAM tiene los permisos necesarios. Para obtener más información sobre el permiso de IAM necesario, consulte Otorgar acceso al rol de IAM CodeBuild del proyecto para usar la conexión y la administración de identidad y acceso para AWS CodeStar las notificaciones y AWS CodeConnections en la Guía del usuario de la consola Developer Tools.

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 read/write acceso 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`.
- admin:org_hook: Concede control total de los enlaces de la organización. Este ámbito solo es obligatorio si usa la característica de webhook de la organización.

Para obtener más información, consulta Cómo entender los alcances de las aplicaciones de OAuth en el sitio web. GitHub

Si usa tokens de acceso personal detallado puede que su token correspondiente necesite los siguientes permisos, en función de su caso de uso:

- Contenido: Read-only: Concede acceso a repositorios privados. Este permiso es obligatorio si usa repositorios privados como origen.
- Estados de confirmación: lectura y escritura: concede permiso para crear estados de confirmación. Este permiso es obligatorio si su proyecto tiene webhook configurado o si tiene habilitada la característica para notificar el estado de compilación.

- Webhooks: lectura y escritura: concede permiso para administrar los webhooks. Este permiso es obligatorio si su proyecto tiene webhook configurado.
- Solicitudes de extracción Read-only: otorga permiso para acceder a las solicitudes de extracción. Este permiso es obligatorio si su webhook tiene un filtro `FILE_PATH` para los eventos de solicitudes de extracción.
- Administración: lectura y escritura: este permiso es necesario si utilizas la función GitHub Actions Runner autohospedada con CodeBuild. Para obtener más información, consulte Create a registration token for a repository y Tutorial: Configurar un ejecutor de CodeBuild-hosted GitHub acciones.

> **Note**
>
> Si quiere acceder a los repositorios de la organización, asegúrese de especificar la organización como propietario 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 información, consulte Creación de un proyecto de compilación (consola).

1. En Proveedor de código fuente, selecciona GitHub.
2. En Credencial, realice una de las siguientes acciones:
   - Elija utilizar credenciales de cuenta para aplicar la credencial de origen predeterminada de su cuenta a todos los proyectos.
     a. Si no estás conectado a GitHub, selecciona Administrar credenciales de la cuenta.
     b. En Tipo de credencial, elija Token de acceso personal.
   - Si ha elegido utilizar credenciales de nivel de cuenta para Servicio, elija qué servicio desea usar para almacenar el token y haga lo siguiente:
     a. Si decide usar Secrets Manager, puede elegir entre usar una conexión de secreto existente o crear un secreto nuevo y elegir Guardar. Para obtener más información

sobre cómo crear un secreto nuevo, consultaCreación y almacenamiento de un token en un secreto de Secrets Manager.

b. Si decide usarlo CodeBuild, introduzca su token de acceso GitHub personal y, a continuación, seleccione Guardar.

- Seleccione Usar credenciales de anulación solo para este proyecto para utilizar una credencial de origen personalizada para anular la configuración de credencial de la cuenta.

a. En la lista de credenciales rellenada, elija una de las opciones de Token de acceso personal.

b. También puedes crear un nuevo token de acceso personal seleccionando crear una nueva conexión de token de acceso personal en la descripción.

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 laReferencia de la línea de comandos.

1. Ejecute el comando import-source-credentials:

```
aws codebuild import-source-credentials --generate-cli-skeleton
```

JSON-formatted los datos aparecen en la salida. Copie los datos a 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:

- *server-type*: valor obligatorio. El proveedor de código fuente utilizado para esta credencial. Los valores válidos son GITHUB, BITBUCKET, GITHUB_ENTERPRISE, GITLAB y GITLAB_SELF_MANAGED.
- *auth-type*: Valor obligatorio. El tipo de autenticación que se utiliza para conectarse a un repositorio. Los valores válidos son OAUTH, BASIC_AUTH, PERSONAL_ACCESS_TOKEN, CODECONNECTIONS y SECRETS_MANAGER. Para GitHub, solo se permite PERSONAL_ACCESS_TOKEN. BASIC_AUTH solo está permitido con la contraseña de la aplicación Bitbucket o el token de API.
- *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. En el caso de Bitbucket, se trata del token de acceso personal, la contraseña de la aplicación o el token de API. Para el tipo de autenticación CODECONNECTIONS, es el ARN de la conexión. Para el tipo de autenticación SECRETS_MANAGER, es el ARN del secreto.
- *username*: valor opcional. Este parámetro se omite para GitHub los proveedores de origen de GitHub Enterprise Server. En el caso de la contraseña de una aplicación de Bitbucket, este es tu nombre de usuario de Bitbucket. En el caso de un token de API de Bitbucket, esta es la dirección de correo electrónico asociada a tu cuenta de Atlassian.

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
```

JSON-formatted los datos aparecen en la salida 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 su cuenta esté conectada con un token de acceso, podrá usarlo `create-project` para crear su CodeBuild proyecto. Para obtener más información, consulte Cree 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
```

Aparece un JSON-formatted `sourceCredentialsInfos` objeto en la salida:

```
{
        "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`, `PERSONAL_ACCESS_TOKEN`, `CODECONNECTIONS` o `SECRETS_MANAGER`.
- El `serverType` es el tipo de proveedor de código fuente. Este valor puede ser `GITHUB`, `GITHUB_ENTERPRISE`, `BITBUCKET`, `GITLAB` o `GITLAB_SELF_MANAGED`.
- 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
```

JSON-formatted los datos 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 información, consulte Creación de un proyecto de compilación (consola).

1. En Proveedor de fuentes, selecciona. GitHub
2. En Credencial, realice una de las siguientes acciones:
   - Elija utilizar credenciales de cuenta para aplicar la credencial de origen predeterminada de su cuenta a todos los proyectos.
     a. Si no estás conectado a GitHub, selecciona Administrar credenciales de la cuenta.
     b. En Tipo de credencial, elija Aplicación de OAuth.
   - Si ha elegido utilizar credenciales de nivel de cuenta para Servicio, elija qué servicio desea usar para almacenar el token y haga lo siguiente:
     a. Si decide usar Secrets Manager, puede elegir entre usar una conexión de secreto existente o crear un secreto nuevo y elegir Guardar. Para obtener más información sobre cómo crear un secreto nuevo, consultaCreación y almacenamiento de un token en un secreto de Secrets Manager.
     b. Si elige usar CodeBuildy, a continuación, selecciona Guardar.
   - Seleccione Usar credenciales de anulación solo para este proyecto para utilizar una credencial de origen personalizada para anular la configuración de credencial de la cuenta.
     a. En la lista de credenciales rellenada, elija una de las opciones de Aplicación de OAuth.

b. También puedes crear un nuevo token de aplicación OAuth seleccionando crear una nueva conexión de token de aplicación OAuth en la descripción.

Para revisar sus aplicaciones OAuth autorizadas, vaya a Aplicaciones en y compruebe que aparezca en la GitHub lista una aplicación cuyo nombre `AWS CodeBuild (region)` sea propiedad de aws-codesuite.

Acceda a Bitbucket en CodeBuild

En el caso de Bitbucket, utilizas un token de acceso, una contraseña de aplicación, un token de API, una aplicación OAuth o una conexión de Bitbucket para acceder al proveedor de origen.

Temas

- Conexiones de aplicación de Bitbucket
- Contraseña, token de API o token de acceso de la aplicación Bitbucket
- Aplicación OAuth de Bitbucket

Conexiones de aplicación de Bitbucket

Puedes usar Bitbucket para conectarte con. CodeBuild Se admiten las conexiones de aplicación de Bitbucket mediante AWS CodeConnections.

> **Note**
>
> AWS CodeConnections está disponible en menos regiones que. CodeBuild Puede utilizar conexiones entre regiones en CodeBuild. Las conexiones creadas en regiones en las que se ha optado por participar no se pueden utilizar en otras regiones. Para obtener más información, consulte Puntos de conexión y cuotas de AWS CodeConnections.

Temas

- Paso 1: creación de una conexión a Bitbucket (consola)
- Paso 2: Otorgue acceso al rol de IAM del CodeBuild proyecto para usar la conexión
- Paso 3: CodeBuild Configúralo para usar la nueva conexión
- Tokens de actualización rotativos de OAuth de Bitbucket

Paso 1: creación de una conexión a Bitbucket (consola)

Sigue estos pasos para usar la CodeBuild consola y añadir una conexión a tu proyecto en Bitbucket.

Para crear una conexión a Bitbucket

- Siga las instrucciones de la Guía del usuario de Herramientas para desarrolladores para la Creación de una conexión a Bitbucket.

> **Note**
>
> En lugar de crear o usar una conexión existente en tu cuenta, puedes usar una conexión compartida desde otra AWS cuenta. Para obtener más información, consulta Compartir conexiones con AWS cuentas.

Paso 2: Otorgue acceso al rol de IAM del CodeBuild proyecto para usar la conexión

Puedes conceder acceso a la función de IAM CodeBuild del proyecto para usar los tokens de Bitbucket que vende tu conexión.

Para conceder acceso a la función de IAM CodeBuild del proyecto

1. Cree un rol de IAM para su CodeBuild proyecto siguiendo las instrucciones CodeBuild Permitir interactuar con otros AWS servicios para su CodeBuild proyecto.
2. Siguiendo las instrucciones, añade la siguiente política de IAM a tu rol en el CodeBuild proyecto para conceder el acceso a la conexión.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "codeconnections:GetConnectionToken",
                "codeconnections:GetConnection"
            ],
            "Resource": [
```

```
                "arn:aws:iam::*:role/Service*"
            ]
        }
    ]
}
```

Paso 3: CodeBuild Configúralo para usar la nueva conexión

Puede configurar una conexión como credencial de cuenta y usarla en un proyecto.

Consola de administración de AWS

Para configurar una conexión como una credencial a nivel de cuenta en el Consola de administración de AWS

1. En Source provider (Proveedor de código fuente), elija Bitbucket.
2. En Credencial, realice una de las siguientes acciones:
   - Elija Credencial de origen predeterminada para aplicar la credencial de origen predeterminada de su cuenta a todos los proyectos.
     a. Si no está conectado a Bitbucket, elija Administrar la credencial de origen predeterminada.
     b. En el tipo de credencial, elija. AWS CodeConnections
     c. En Conexión, elija usar una conexión existente o cree una nueva.
   - Elija Credencial de origen personalizada a fin de usar una credencial de este tipo para anular la configuración predeterminada de su cuenta.
     a. En Tipo de credencial, elija. AWS CodeConnections
     b. En Conexión, elija usar una conexión existente o cree una nueva.

AWS CLI

Para configurar una conexión como una credencial a nivel de cuenta en el AWS CLI

- Abra un terminal (Linux, macOS o Unix) o un símbolo del sistema (Windows). Utilice el AWS CLI para ejecutar el import-source-credentials comando, especificando el `--auth-type`-- `server-type`, y `--token` para la conexión.

Utilice el siguiente comando :

```
aws codebuild import-source-credentials --auth-type CODECONNECTIONS --server-
type BITBUCKET --token <connection-arn>
```

Para obtener más información sobre la configuración de varios tokens en su CodeBuild proyecto, consulteConfiguración de varios tokens como credenciales de origen.

Tokens de actualización rotativos de OAuth de Bitbucket

Atlassian ahora aplica los tokens de actualización rotativos de un solo uso para Bitbucket OAuth. Cada vez que utilizas un token de actualización, Bitbucket lo invalida y devuelve uno nuevo. Para obtener más información, consulta los tokens de actualización OAuth de un solo uso de Bitbucket (CHANGE-3052) en el sitio web para desarrolladores de Atlassian.

Acción Manager-stored obligatoria para las credenciales de Secrets

Si guardas las credenciales OAuth de Bitbucket en Secrets Manager, añade el `secretsmanager:PutSecretValue` permiso a tu CodeBuild rol de servicio. Con este permiso, CodeBuild puedes volver a escribir el token de actualización actualizado en tu secreto después de cada actualización del token. Sin este permiso, se produce un error en la siguiente compilación tras la actualización del token.

El siguiente ejemplo de declaración de política de IAM concede el permiso necesario:

```
{
    "Effect": "Allow",
    "Action": "secretsmanager:PutSecretValue",
    "Resource": "arn:aws:secretsmanager:*:*:secret:codebuild/*"
}
```

Si utiliza CodeBuild-managed esta opción, no es necesario realizar cambios. CodeBuild gestiona la rotación de los tokens automáticamente.

Migre a AWS CodeConnections (recomendado)

Te recomendamos migrar a AWS CodeConnections para integrarte con Bitbucket. AWS CodeConnections utiliza un mecanismo de autorización independiente al que no afecta la rotación de los tokens de actualización. No es necesario administrar los tokens ni añadir permisos.

Para migrar a AWS CodeConnections, complete los siguientes pasos:

1. Abra la consola de herramientas para desarrolladores de . Seleccione Configuración, Conexiones y, a continuación, Crear conexión. Selecciona Bitbucket como proveedor.
2. Autoriza la AWS CodeConnections aplicación en tu espacio de trabajo de Bitbucket. Esto requiere el permiso de administración del espacio de trabajo en Bitbucket.
3. Añade los siguientes permisos a tu función CodeBuild de servicio: `codeconnections:GetConnectionToken` y`codeconnections:GetConnection`.
4. Configura la conexión como tu credencial predeterminada:

```
aws codebuild import-source-credentials --auth-type CODECONNECTIONS --server-type
 BITBUCKET --token <connection-arn>
```

5. Todos los proyectos que utilizan la credencial predeterminada cambian automáticamente. Actualice individualmente los proyectos que utilizan credenciales personalizadas para cada proyecto. Usa la consola o la `UpdateProject` API.

Contraseña, token de API o token de acceso de la aplicación Bitbucket

Las contraseñas de las aplicaciones de Bitbucket están en desuso

Atlassian ha dejado en desuso las contraseñas de las aplicaciones de Bitbucket en favor de los tokens de API. Para seguir utilizando la autenticación básica CodeBuild, crea un token de API en la configuración de tu cuenta de Atlassian. Cuando lo conectes a CodeBuild, utiliza la dirección de correo electrónico asociada a tu cuenta de Atlassian como nombre de usuario. CodeBuild utiliza automáticamente las credenciales correctas para las operaciones de Git y las llamadas a la API de Bitbucket.
Para obtener más información, consulta Cómo usar los tokens de API en el sitio web de Atlassian.

Rotación de los tokens OAuth de Bitbucket

Si utilizas las credenciales OAuth de Bitbucket, te recomendamos migrar a. Conexiones de aplicación de Bitbucket Para obtener información sobre los cambios necesarios para seguir utilizando OAuth con Secrets Manager, consulte. Aplicación OAuth de Bitbucket

Requisitos previos

Antes de empezar, debes añadir los ámbitos de permiso adecuados a la contraseña, el token de API o el token de acceso de tu aplicación de Bitbucket.

En el caso de Bitbucket, la contraseña de tu aplicación, el token de API o el token de acceso 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 su proyecto tiene un webhook de Bitbucket, entonces la contraseña de su aplicación o el token de acceso debe tener este ámbito.
- webhook: Concede acceso a webhooks. Si su proyecto tiene una operación de webhook, entonces la contraseña de su aplicación o el token de acceso debe tener este ámbito.
- cuenta: otorga acceso de lectura a la información de la cuenta del usuario.

Para obtener más información, consulte Scopes for Bitbucket Cloud REST API y OAuth on Bitbucket Cloud en el sitio web de Bitbucket.

Connect Bitbucket con una contraseña de aplicación o un token de API (consola)

Para usar la consola para conectar tu proyecto a Bitbucket mediante una contraseña de aplicación o un token de API, haz lo siguiente al crear un proyecto. Para obtener información, consulte Creación de un proyecto de compilación (consola).

> **Note**
>
> Si utilizas un token de API, introduce la dirección de correo electrónico asociada a tu cuenta de Atlassian como nombre de usuario.

1. En Source provider (Proveedor de código fuente), elija Bitbucket.
2. En Credencial, realice una de las siguientes acciones:
   - Elija utilizar credenciales de cuenta para aplicar la credencial de origen predeterminada de su cuenta a todos los proyectos.

     a. Si no está conectado a Bitbucket, elija Administrar la credencial de cuenta.

b. Para el tipo de credencial, elige el token de API (que anteriormente se mostraba como contraseña de la aplicación).

- Si ha elegido utilizar credenciales de nivel de cuenta para Servicio, elija qué servicio desea usar para almacenar el token y haga lo siguiente:

  a. Si decide usar Secrets Manager, puede elegir entre usar una conexión de secreto existente o crear un secreto nuevo y elegir Guardar. Para obtener más información sobre cómo crear un secreto nuevo, consulteCreación y almacenamiento de un token en un secreto de Secrets Manager.

  b. Si decides usarlo CodeBuild, introduce tu nombre de usuario y contraseña de la aplicación de Bitbucket y, a continuación, selecciona Guardar. Para un token de API de Bitbucket, introduce la dirección de correo electrónico y el token de API de tu cuenta de Atlassian en su lugar.

- Seleccione Usar credenciales de anulación solo para este proyecto para utilizar una credencial de origen personalizada para anular la configuración de credencial de la cuenta.

  a. En la lista de credenciales rellenada, elige una de las opciones de la contraseña de la aplicación o del token de API.

  b. También puede crear una nueva conexión de contraseña de aplicación o token de API seleccionando el enlace de creación de conexión en la descripción.

Conexión de Bitbucket con un token de acceso (consola)

Si quiere utilizar la consola para conectar el proyecto a Bitbucket con un token de acceso, haga lo siguiente a la hora de crear un proyecto. Para obtener información, consulte Creación de un proyecto de compilación (consola).

1. En Source provider (Proveedor de código fuente), elija Bitbucket.
2. En Credencial, realice una de las siguientes acciones:
   - Elija utilizar credenciales de cuenta para aplicar la credencial de origen predeterminada de su cuenta a todos los proyectos.

     a. Si no está conectado a Bitbucket, elija Administrar la credencial de cuenta.

     b. En Tipo de credencial, elija Token de acceso personal.

   - Si ha elegido utilizar credenciales de nivel de cuenta para Servicio, elija qué servicio desea usar para almacenar el token y haga lo siguiente:

a. Si decide usar Secrets Manager, puede elegir entre usar una conexión de secreto existente o crear un secreto nuevo y elegir Guardar. Para obtener más información sobre cómo crear un secreto nuevo, consulteCreación y almacenamiento de un token en un secreto de Secrets Manager.

b. Si decides usarlo CodeBuild, introduce tu token de acceso personal de Bitbucket y, a continuación, selecciona Guardar.

- Seleccione Usar credenciales de anulación solo para este proyecto para utilizar una credencial de origen personalizada para anular la configuración de credencial de la cuenta.

a. En la lista de credenciales rellenada, elija una de las opciones de Token de acceso personal.

b. También puedes crear un nuevo token de acceso personal seleccionando crear una nueva conexión de token de acceso personal en la descripción.

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 laReferencia de la línea de comandos.

1. Ejecute el comando import-source-credentials:

```
aws codebuild import-source-credentials --generate-cli-skeleton
```

JSON-formatted los datos aparecen en la salida. Copie los datos a 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:

- *server-type*: valor obligatorio. El proveedor de código fuente utilizado para esta credencial. Los valores válidos son GITHUB, BITBUCKET, GITHUB_ENTERPRISE, GITLAB y GITLAB_SELF_MANAGED.
- *auth-type*: Valor obligatorio. El tipo de autenticación que se utiliza para conectarse a un repositorio. Los valores válidos son OAUTH, BASIC_AUTH, PERSONAL_ACCESS_TOKEN, CODECONNECTIONS y SECRETS_MANAGER. Para GitHub, solo se permite PERSONAL_ACCESS_TOKEN. BASIC_AUTH solo está permitido con la contraseña de la aplicación Bitbucket o el token de API.
- *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. En el caso de Bitbucket, se trata del token de acceso personal, la contraseña de la aplicación o el token de API. Para el tipo de autenticación CODECONNECTIONS, es el ARN de la conexión. Para el tipo de autenticación SECRETS_MANAGER, es el ARN del secreto.
- *username*: valor opcional. Este parámetro se omite para GitHub los proveedores de origen de GitHub Enterprise Server. En el caso de la contraseña de una aplicación de Bitbucket, este es tu nombre de usuario de Bitbucket. En el caso de un token de API de Bitbucket, esta es la dirección de correo electrónico asociada a tu cuenta de Atlassian.

2. Para conectar tu cuenta con una contraseña de aplicación, un token de API 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
```

JSON-formatted los datos aparecen en la salida 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 su cuenta esté conectada con una contraseña de aplicación o un token de API, podrá utilizarlos `create-project` para crear su CodeBuild proyecto. Para obtener más información, consulte Cree un proyecto de compilación (AWS CLI).

3. Para ver las contraseñas de aplicación o los tokens de acceso conectados, ejecute el comando list-source-credentials.

```
aws codebuild list-source-credentials
```

Aparece un JSON-formatted `sourceCredentialsInfos` objeto en la salida:

```
{
        "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`, `PERSONAL_ACCESS_TOKEN`, `CODECONNECTIONS` o `SECRETS_MANAGER`.
- El `serverType` es el tipo de proveedor de código fuente. Este valor puede ser `GITHUB`, `GITHUB_ENTERPRISE`, `BITBUCKET`, `GITLAB` o `GITLAB_SELF_MANAGED`.
- El `arn` es el ARN del token.

4. Para desconectarse de un proveedor de origen y eliminar sus tokens de acceso o contraseña de aplicación, ejecute el comando delete-source-credentials con su ARN.

```
aws codebuild delete-source-credentials --arn arn-of-your-credentials
```

JSON-formatted los datos se devuelven con un ARN de las credenciales eliminadas.

```
{
    "arn": "arn:aws:codebuild:region:account-id:token/server-type"
    }
```

Aplicación OAuth de Bitbucket

> **Acción necesaria para la rotación del token OAuth de Bitbucket**
>
> Atlassian ahora aplica los tokens de actualización rotativos de un solo uso para Bitbucket OAuth. Recomendamos migrar a, ya que elimina por completo la administración de los tokens. Conexiones de aplicación de Bitbucket Para obtener instrucciones sobre la migración, consulte. Migre a AWS CodeConnections (recomendado)
> Si prefieres seguir usando OAuth con Secrets Manager, añade el `secretsmanager:PutSecretValue` permiso a tu rol de CodeBuild servicio. Para obtener más información, consulte Acción Manager-stored obligatoria para las credenciales de Secrets.
> Si utilizas CodeBuild-managed esta opción, no es necesario realizar cambios. CodeBuild gestiona la rotación de los tokens automáticamente.

Conexión de Bitbucket mediante OAuth (consola)

Si quiere utilizar la consola para conectar el proyecto a Bitbucket mediante una aplicación OAuth, haga lo siguiente a la hora de crear un proyecto. Para obtener información, consulte Creación de un proyecto de compilación (consola).

1. En Source provider (Proveedor de código fuente), elija Bitbucket.
2. En Credencial, realice una de las siguientes acciones:
   - Elija utilizar credenciales de cuenta para aplicar la credencial de origen predeterminada de su cuenta a todos los proyectos.

a. Si no está conectado a Bitbucket, elija Administrar la credencial de cuenta.

b. En Tipo de credencial, elija Aplicación de OAuth.

- Si ha elegido utilizar credenciales de nivel de cuenta para Servicio, elija qué servicio desea usar para almacenar el token y haga lo siguiente:

  a. Si decide usar Secrets Manager, puede elegir entre usar una conexión de secreto existente o crear un secreto nuevo y elegir Guardar. Para obtener más información sobre la creación de un secreto nuevo, consulteCreación y almacenamiento de un token en un secreto de Secrets Manager.

  b. Si elige usar CodeBuildy, a continuación, selecciona Guardar.

- Seleccione Usar credenciales de anulación solo para este proyecto para utilizar una credencial de origen personalizada para anular la configuración de credencial de la cuenta.

  a. En la lista de credenciales rellenada, elija una de las opciones de Aplicación de OAuth.

  b. También puedes crear un nuevo token de aplicación OAuth seleccionando crear una nueva conexión de token de aplicación OAuth en la descripción.

Para revisar sus aplicaciones OAuth autorizadas, vaya a Autorizaciones de aplicaciones en Bitbucket y compruebe que se enumera una aplicación llamada `AWS CodeBuild (region)`.

GitLab acceder en CodeBuild

Para GitLab, utiliza una GitLab conexión para acceder al proveedor de origen.

Temas

- Conectarse CodeBuild a GitLab

Conectarse CodeBuild a GitLab

Las conexiones le permiten autorizar y establecer configuraciones que asocian a su proveedor externo con AWS los recursos que utiliza AWS CodeConnections. Para asociar su repositorio de terceros como origen del proyecto de compilación, debe usar una conexión.

Para añadir un proveedor de fuentes GitLab autogestionado GitLab o autogestionado CodeBuild, puede 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 GitLabopción de proveedor GitLab autogestionado. Consulte Crea una conexión a GitLab (consola) para agregar el proveedor de origen. La consola le ayuda a crear un recurso de conexiones.
- Use 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 GitLab, concedes a nuestro servicio permisos para procesar tus datos accediendo a tu cuenta, y puedes revocar los permisos en cualquier momento desinstalando la aplicación.

Crea una conexión a GitLab

En esta sección se describe cómo conectarse GitLab a CodeBuild. Para obtener más información acerca de las conexiones de GitLab , consulte Conectarse CodeBuild a GitLab.

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.

> **Note**
>
> En lugar de crear o usar una conexión existente en tu cuenta, puedes usar una conexión compartida desde otra AWS cuenta. Para obtener más información, consulta Compartir conexiones con AWS cuentas.

Para crear o editar el proyecto de compilación

1. Inicia sesión en la CodeBuild consola.
2. Elija una de las siguientes opciones.
   - Elija crear un proyecto de compilación. Sigue 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, selecciona GitLab.
   - Elija editar un proyecto de compilación existente. Elija Editar y, después, Origen. En la página Editar fuente, en Proveedor de fuentes, selecciona GitLab.

3. Seleccione una de las siguientes opciones:

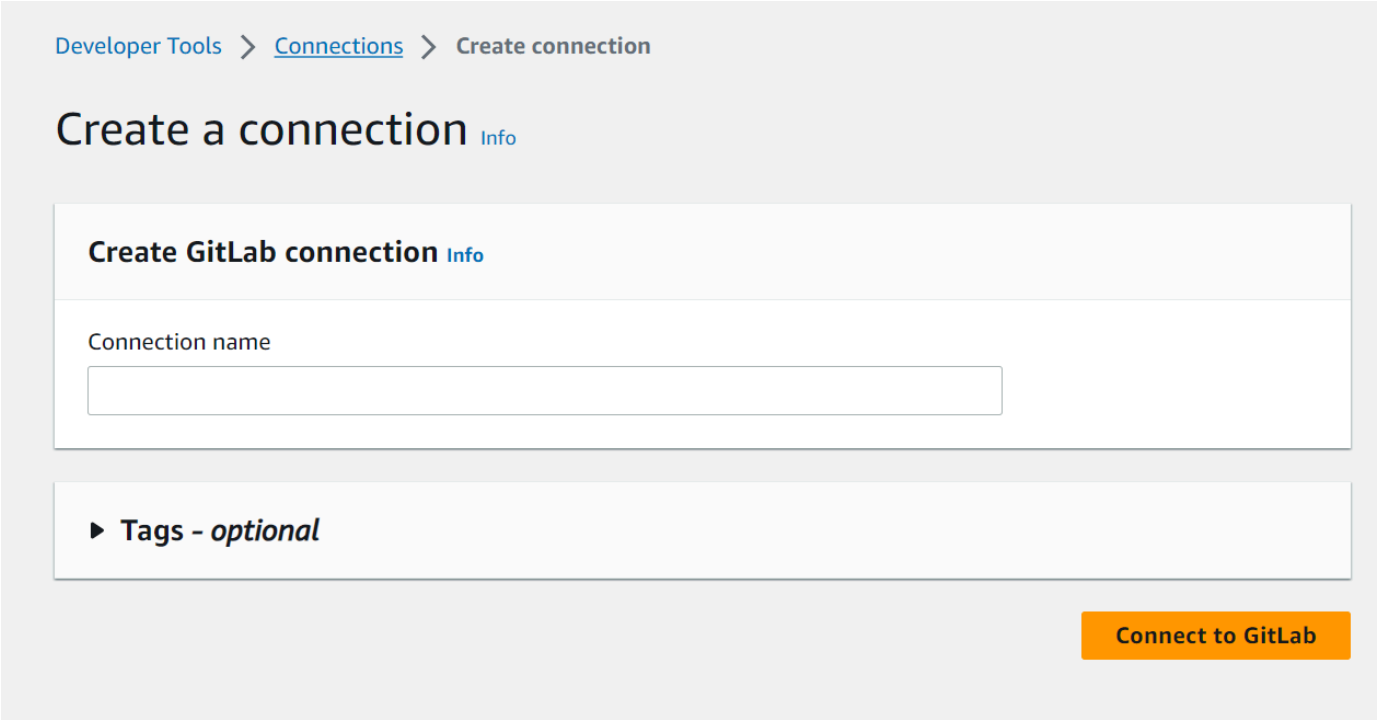
   - En Conexión, elija Conexión predeterminada. La conexión predeterminada aplica una GitLab conexión predeterminada en todos los proyectos.
   - En Conexión, elija Conexión personalizada. La conexión personalizada aplica una GitLab conexión personalizada que anula la configuración predeterminada de tu cuenta.

4. Realice una de las siguientes acciones:

   - 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.

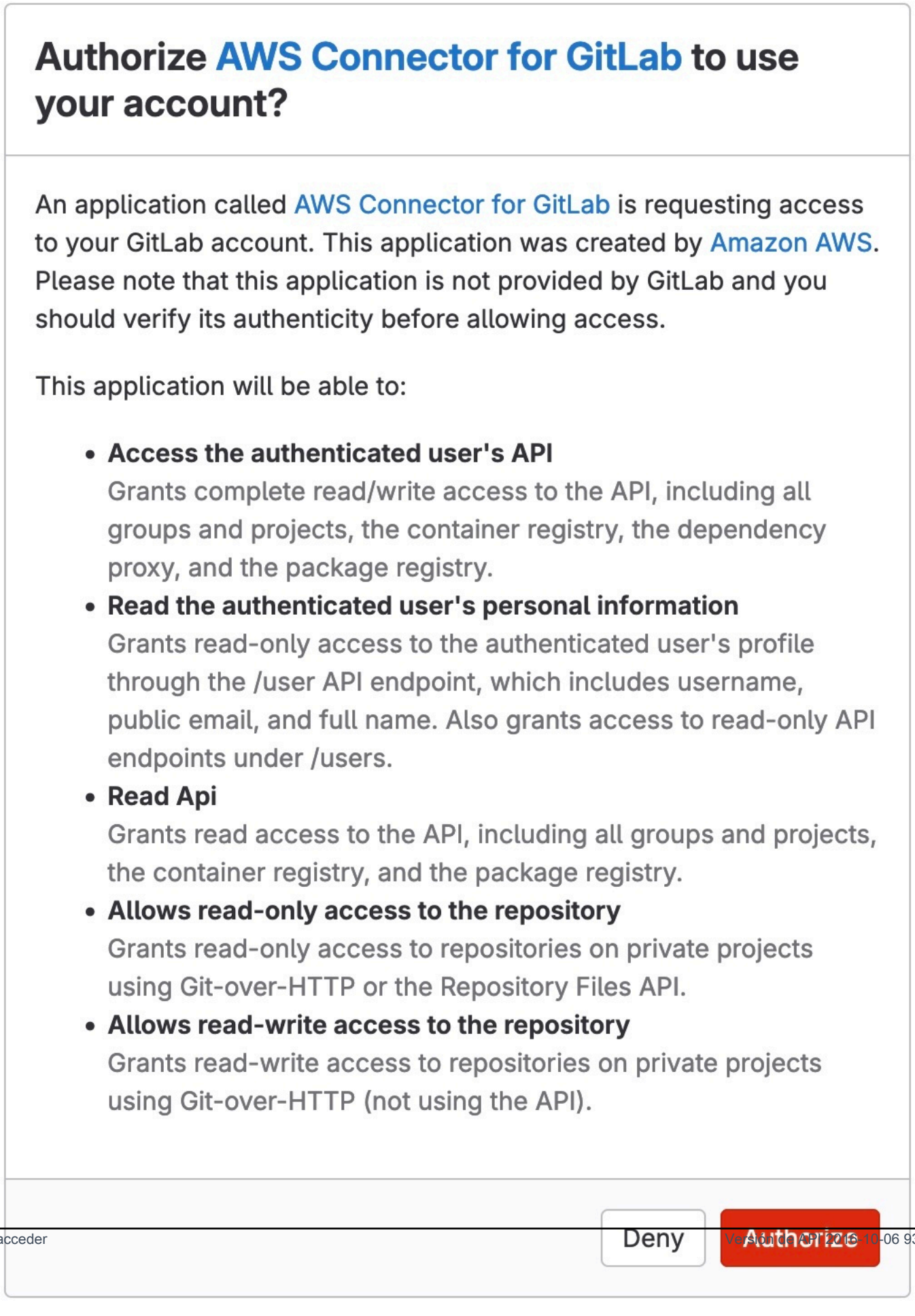
   > **Note**
   >
   > Si cierras la ventana emergente antes de crear una GitLab conexión, tendrás que actualizar la página.

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. Elija Connect to (Conectar a 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).
Deny
Authorize

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 el proyecto de compilación, en la lista desplegable Conexión predeterminada o Conexión personalizada, asegúrese de que aparezca el ARN de su conexión. De lo contrario, elija el botón de actualización para que aparezca.
11. En el repositorio, elige el nombre de tu proyecto GitLab especificando la ruta del proyecto con el espacio de nombres. Por ejemplo, para un repositorio 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 de #get -single-project. https://docs.gitlab.com/ee/api/projects.html Para obtener más información sobre el espacio de nombres de, consulte. GitLab https://docs.gitlab.com/ee/user/namespace/

    > **Note**
    >
    > Para los grupos de GitLab, 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

12. En Versión de origen (opcional), especifique un ID de solicitud de extracción, una ramificación, un ID de confirmación, un etiqueta o referencia y un ID de confirmación. Para obtener más información, consulte Ejemplo de 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 Profundidad del clon de Git (opcional), puede crear un clon superficial con un historial truncado en el número de confirmaciones especificado. Si desea un clon completo, elija Completo.
14. En Estado de la compilación (opcional), seleccione Informe acerca de los estados de compilación al proveedor de origen al inicio y al final de sus compilaciones si quiere que se notifique el estado del inicio y finalización de la compilación al proveedor de origen.

    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 CloudFormation, utilice la consola para editar la conexión y establecer su estado`AVAILABLE`.

Creación de una conexión

- Siga las instrucciones de la Guía del usuario de la consola Developer Tools para crear una conexión a GitLab (CLI).

Cross-service prevención confusa de diputados

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. Cross-service la suplantación de identidad puede producirse cuando un servicio (el servicio de 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`y las claves de contexto en las políticas de recursos para limitar los permisos que se AWS CodeBuild otorgan a otro servicio al recurso. Utiliza `aws:SourceArn` si desea que solo se asocie un recurso al acceso entre servicios. Utiliza `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 se pueden 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.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "codebuild.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceArn": "arn:aws:codebuild:us-
east-1:111122223333:project/MyProject"
```

```
                }
            }
        }
    ]
}
```

Temas avanzados

Esta sección contiene varios temas avanzados que son útiles para los usuarios de AWS CodeBuild con más experiencia.

Temas

- Permita a los usuarios interactuar con CodeBuild
- CodeBuild Permitir interactuar con otros AWS servicios
- Cifrado de las salidas de compilación con una clave administrada por el cliente
- Interactúe con el CodeBuild uso del AWS CLI
- Referencia de línea de comandos para AWS CodeBuild
- AWS Referencia de SDK y herramientas para AWS CodeBuild
- Uso de este servicio con un AWS SDK
- Especifique el AWS CodeBuild punto de conexión
- Uso AWS CodeBuild por AWS CodePipeline para probar el código y ejecutar compilaciones
- Uso AWS CodeBuild con Codecov
- Uso AWS CodeBuild con Jenkins
- Uso AWS CodeBuild con aplicaciones sin servidor
- Avisos de terceros para AWS CodeBuild para Windows
- Utilice claves de CodeBuild condición como variables de rol del servicio de IAM para controlar el acceso a la compilación
- AWS CodeBuild claves de condición

Permita a los usuarios interactuar con CodeBuild

Si sigue los pasos Primeros pasos con la consola AWS CodeBuild para acceder por primera vez, lo más probable es que no necesite la información de este tema. Sin embargo, a medida que continúe usándolo CodeBuild, es posible que desee hacer cosas como ofrecer a otros usuarios y grupos de su organización la posibilidad de interactuar con ellos CodeBuild.

Para permitir que un usuario o un grupo de IAM interactúe con ellos AWS CodeBuild, debes concederles permisos de acceso para CodeBuild hacerlo. En esta sección, se describe cómo hacer esto con la consola de IAM o la AWS CLI.

Si vas a acceder CodeBuild con tu cuenta AWS raíz (no se recomienda) o con un usuario administrador de tu AWS cuenta, no es necesario que sigas estas instrucciones.

Para obtener información sobre las cuentas AWS raíz y los usuarios administradores, consulte El usuario Cuenta de AWS raíz y Cómo crear su primer usuario y grupo Cuenta de AWS raíz en la Guía del usuario.

Para añadir permisos de CodeBuild acceso a un grupo o usuario de IAM (consola)

1. Abra la consola de IAM en https://console.aws.amazon.com/iam/.

   Ya deberías haber iniciado sesión en el Consola de administración de AWS mediante una de las siguientes opciones:

   - Tu cuenta AWS root. No se recomienda. Para obtener más información, consulte El usuario raíz de Cuenta de AWS en la Guía del usuario.
   - Un usuario administrador de tu AWS cuenta. Para obtener más información, consulte Crear su primer usuario y grupo Cuenta de AWS raíz en la Guía del usuario.
   - Un usuario de su AWS cuenta con permiso para realizar el siguiente conjunto mínimo de acciones:

     ```
     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.
3. Para añadir un conjunto personalizado de permisos de AWS CodeBuild acceso a un grupo o usuario de IAM, vaya directamente al paso 4 de este procedimiento.

   Para añadir un conjunto predeterminado de permisos de CodeBuild acceso a un grupo de IAM o a un usuario de IAM, elija Tipo de política, AWS Administrado y, a continuación, haga lo siguiente:

- Para añadir permisos de acceso total CodeBuild, selecciona la casilla denominada, elige Acciones de política y AWSCodeBuildAdminAccess, a continuación, selecciona Adjuntar. 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 AmazonS3ReadOnlyAccess e IAMFullAccess.
- Para añadir permisos de acceso a CodeBuild todo, excepto a la administración de proyectos de construcción, selecciona la casilla denominada AWSCodeBuildDeveloperAccess, elige Acciones de política y, a continuación, selecciona Adjuntar. 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 acceso de solo lectura CodeBuild, selecciona las casillas denominadas. 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 agregado un conjunto predeterminado de permisos de CodeBuild acceso a un grupo o usuario de IAM. Omita el resto de los pasos de este procedimiento.

4. Seleccione 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).

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "CodeBuildAccessPolicy",
            "Effect": "Allow",
            "Action": [
                "codebuild:*"
```

```
            ],
            "Resource": "*"
        },
        {
            "Sid": "CodeBuildRolePolicy",
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": "arn:aws:iam::111122223333: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 CodeBuild las acciones y a un número potencialmente elevado de AWS recursos. Para restringir los permisos a CodeBuild acciones específicas, cambie el valor de `codebuild:*` en la declaración CodeBuild de política. Para obtener más información, consulte Identity and Access Management. Para restringir el acceso a AWS recursos específicos, cambie el valor del `Resource` objeto. Para obtener más información, consulte Identity and Access Management.

8. En el panel de navegación, elija Groups o Users.
9. En la lista de grupos o usuarios, elija el nombre del grupo o usuario de IAM al que quiere añadir los permisos de CodeBuild acceso.
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. Para un grupo, en la página Adjuntar política, seleccione y, a continuación CodeBuildAccessPolicy, elija Adjuntar política.

    Para un usuario, en la página Añadir permisos, elija Asociar políticas existentes directamente. Seleccione CodeBuildAccessPolicy Siguiente: Revisar y, a continuación, elija Agregar permisos.

Para añadir permisos de CodeBuild acceso a un grupo o usuario de IAM (AWS CLI)

1. Asegúrese de haberlo configurado AWS CLI con la clave de AWS acceso y la clave de acceso AWS secreta correspondientes a una de las entidades de IAM, tal y como se describe 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 añadir un conjunto personalizado de permisos de AWS CodeBuild acceso a un grupo o usuario de IAM, vaya al paso 3 de este procedimiento.

Para añadir un conjunto predeterminado de permisos de CodeBuild acceso 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, sustituyendo *group-name* o sustituyendo por el nombre del grupo o *user-name* el nombre de usuario de IAM y sustituyéndolo *policy-arn* una vez por cada uno de los siguientes nombres de recursos de Amazon (ARN) de la política:

- Para añadir permisos de acceso total CodeBuild, utilice los siguientes ARN de la política:
  - `arn:aws:iam::aws:policy/AWSCodeBuildAdminAccess`
  - `arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess`
  - `arn:aws:iam::aws:policy/IAMFullAccess`
- Para añadir permisos de acceso a todo tipo de CodeBuild aplicaciones, excepto a la administración de proyectos de construcción, usa los siguientes ARN de política:
  - `arn:aws:iam::aws:policy/AWSCodeBuildDeveloperAccess`
  - `arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess`
- Para añadir permisos de acceso de solo lectura CodeBuild, usa los siguientes ARN de política:
  - `arn:aws:iam::aws:policy/AWSCodeBuildReadOnlyAccess`
  - `arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess`

Ahora ha agregado un conjunto predeterminado de permisos de CodeBuild acceso a un grupo o usuario de IAM. Omita el resto de los pasos de este procedimiento.

3. En un directorio vacío de la estación de trabajo local o instancia en la que AWS CLI esté instalado, cree un archivo llamado `put-group-policy.json` o. `put-user-policy.json` Si elige otro nombre de archivo, no olvide utilizarlo durante todo este procedimiento.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "CodeBuildAccessPolicy",
            "Effect": "Allow",
            "Action": [
                "codebuild:*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CodeBuildRolePolicy",
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": "arn:aws:iam::111122223333: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 CodeBuild las acciones y a una cantidad potencialmente grande de AWS recursos. Para restringir los permisos a CodeBuild acciones específicas, cambie el valor de `codebuild:*` en la declaración CodeBuild de política. Para obtener más información, consulte Identity and Access Management. Para restringir el acceso a AWS recursos específicos, cambie el valor del `Resource` objeto relacionado. Para obtener más información, consulte Identity and Access Management o la documentación de seguridad del servicio de AWS específico.

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:

   ```
   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 del usuario o grupo de IAM de destino.

CodeBuild Permitir interactuar con otros AWS servicios

Si sigue los pasos Primeros pasos con la consola AWS CodeBuild para acceder por primera vez, lo más probable es que no necesite la información de este tema. Sin embargo, a medida que continúe usándolo CodeBuild, es posible que desee hacer cosas como CodeBuild permitir la interacción con otros AWS servicios.

CodeBuild Para poder interactuar con AWS los servicios dependientes en tu nombre, necesitas un rol AWS CodeBuild de servicio. Puede crear un rol CodeBuild de servicio mediante las AWS CodePipeline consolas CodeBuild o. Para obtener información, consulte:

- Creación de un proyecto de compilación (consola)
- Crear una canalización que utilice CodeBuild (CodePipeline consola)
- Añadir una acción de CodeBuild compilación a una canalización (CodePipeline consola)
- Cambiar la configuración de un proyecto de compilación (consola)

Si no tiene previsto utilizar estas consolas, en esta sección se describe cómo crear un rol de CodeBuild servicio con la consola de IAM o la AWS CLI.

> **Important**
>
> CodeBuild usa la función 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 CodeBuild de servicio (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:

   - Tu cuenta AWS raíz. No se recomienda. Para obtener más información, consulte El usuario raíz de Cuenta de AWS en la Guía del usuario.

- Un usuario administrador de tu AWS cuenta. Para obtener más información, consulte Crear su primer usuario y grupo Cuenta de AWS raíz en la Guía del usuario.
- Un usuario de su AWS cuenta con permiso para realizar el siguiente conjunto mínimo de acciones:

```
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.
3. Seleccione Crear política.
4. En la página Create Policy, elija JSON.
5. En la política JSON, escriba lo siguiente y elija Review Policy (Consultar política):

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": "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 declaraciones que permiten el acceso a una cantidad potencialmente grande de AWS recursos. Para restringir el acceso AWS CodeBuild a AWS recursos específicos, cambie el valor de la `Resource` matriz. Para obtener más información, consulte la documentación de seguridad del AWS servicio.

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 el AWS servicio ya seleccionado, elija y CodeBuild, a continuación, elija Next:Permissions.
10. En la página Adjuntar políticas de permisos, selecciona y CodeBuildServiceRolePolicy, a continuación, elige Siguiente: Revisar.
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).

Para crear un rol CodeBuild de servicio (AWS CLI)

1. Asegúrese de haberlo configurado AWS CLI con la clave de AWS acceso y la clave de acceso AWS secreta que corresponden a una de las entidades de IAM, tal y como se describe 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 local o instancia en la que AWS CLI esté instalado, cree dos archivos con el nombre `create-role.json` y. `put-role-policy.json` Si elige nombres de archivo diferentes, no olvide utilizarlos durante todo este procedimiento.

   `create-role.json`:

   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 CodeBuild proyecto y `aws:SourceArn` es el ARN del CodeBuild proyecto.

   Si quieres restringir tu rol de servicio a una AWS cuenta, `create-role.json` puede tener un aspecto similar al siguiente:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "codebuild.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": [
                        "111122223333"
                    ]
                }
            }
        }
    ]
}
```

Si quieres restringir tu rol de servicio a un CodeBuild proyecto específico, `create-role.json` puede tener un aspecto similar al siguiente:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "codebuild.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
```

```
                    "aws:SourceArn": "arn:aws:codebuild:us-
east-1:111122223333:project/MyProject"
                }
            }
        }
    ]
}
```

> **Note**
>
> Si no sabes un nombre para tu CodeBuild proyecto o aún no lo has decidido y quieres que la política de confianza restrinja un patrón de ARN concreto, puedes reemplazar esa parte del ARN por un comodín (*). Después de crear el proyecto, puede actualizar la política de confianza.

`put-role-policy.json`:

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 declaraciones que permiten el acceso a una cantidad potencialmente grande de recursos. AWS Para restringir el acceso AWS CodeBuild a AWS recursos específicos, cambie el valor de la `Resource` matriz. Para obtener más información, consulte la documentación de seguridad del AWS servicio.

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
```

Cifrado de las salidas de compilación con una clave administrada por el cliente

Si sigue los pasos Primeros pasos con la consola AWS CodeBuild para acceder por primera vez, lo más probable es que no necesite la información de este tema. Sin embargo, a medida que continúe CodeBuild usándolo, es posible que desee hacer cosas como cifrar los artefactos de compilación.

AWS CodeBuild Para cifrar sus artefactos de salida de compilación, necesita acceder a una clave de KMS. De forma predeterminada, CodeBuild utiliza el Clave administrada de AWS para Amazon S3 en su AWS cuenta.

Si no desea utilizarla Clave administrada de AWS, debe crear y configurar usted mismo una clave gestionada por el cliente. 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 .

Para configurar una clave gestionada por el cliente para que la utilice CodeBuild, siga las instrucciones de la sección «Cómo modificar una política de claves» de la Guía para AWS KMS desarrolladores. A continuación, añada las siguientes afirmaciones (entre *### BEGIN ADDING STATEMENTS HERE ###* y*### END ADDING STATEMENTS HERE ###*) a la política de claves. 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.

JSON

```
{
  "Version":"2012-10-17",
```

```
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "kms:ViaService": "s3.us-east-1.amazonaws.com",
          "kms:CallerAccount": "111122223333"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
      ],
      "Resource": "*"
    }
  ]
}
```

- *region-ID*representa el ID de la AWS región en la que se CodeBuild encuentran los buckets de Amazon S3 asociados (por ejemplo,`us-east-1`).
- *account-ID*representa el ID de la AWS cuenta propietaria de la clave gestionada por el cliente.
- *CodeBuild-service-role*representa el nombre del rol de CodeBuild servicio que creó o identificó anteriormente en este tema.

> **Note**
>
> Para crear o configurar una clave gestionada por el cliente a través de la consola de IAM, primero debes iniciar sesión en ella Consola de administración de AWS mediante una de las siguientes opciones:
>
> - Tu cuenta AWS raíz. No se recomienda. Para obtener más información, consulte El usuario raíz de la cuenta en la Guía del usuario.
> - Un usuario administrador de tu AWS cuenta. Para obtener más información, consulte Crear su primer usuario y grupo Cuenta de AWS raíz en la Guía del usuario.
> - Un usuario de su AWS cuenta con permiso para crear o modificar la clave gestionada por el cliente. Para obtener más información, consulta los permisos necesarios para usar la AWS KMS consola en la Guía para AWS KMS desarrolladores.

Interactúe con el CodeBuild uso del AWS CLI

Si sigue los pasos descritos Primeros pasos con la consola AWS CodeBuild para acceder por primera vez, lo más probable es que no necesite la información de este tema. Sin embargo, a medida que continúe usándolo CodeBuild, es posible que desee hacer cosas como permitir que los usuarios lo usen AWS CLI para interactuar con CodeBuild la CodeBuild consola, la consola o los AWS SDK (o además de ellos). CodePipeline

Para instalarlo y configurarlo AWS CLI, consulte Cómo configurarlo AWS Command Line Interface en la Guía del AWS Command Line Interface usuario.

Tras instalarlo AWS CLI, complete las siguientes tareas:

1. Ejecute el siguiente comando para confirmar si la instalación del dispositivo AWS CLI es compatible 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 línea de comandos para AWS CodeBuild

AWS CLI Proporciona comandos para la automatización. 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 quieres usar los AWS SDK para realizar llamadas CodeBuild, consulta la. AWS Referencia de SDK y herramientas

Para utilizar la información de este tema, ya debe haberlo instalado AWS CLI y configurado para su uso con CodeBuild, tal y como se describe enInteractúe con el CodeBuild uso del AWS CLI.

Para usar el AWS CLI para especificar el punto final CodeBuild, consulteEspecifique el AWS CodeBuild punto final (AWS CLI).

Ejecute este comando para obtener una lista de CodeBuild comandos.

```
aws codebuild help
```

Ejecute este comando para obtener información sobre un CodeBuild comando, donde *command-name* está el nombre del comando.

```
aws codebuild command-name help
```

CodeBuild los comandos incluyen:

- `batch-delete-builds`: Elimina una o más compilaciones. 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 Cree 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 Consulta una lista de los identificadores de compilación de un proyecto de compilación (AWS CLI).
- `list-curated-environment-images`: Obtiene una lista de imágenes de Docker gestionadas por él 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 Ejecute 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 Cambia la configuración de un proyecto de compilación (AWS CLI).

AWS Referencia de SDK y herramientas para AWS CodeBuild

Para usar uno de los AWS SDK o herramientas para automatizar AWS CodeBuild, consulte los siguientes recursos.

Si quieres usar el AWS CLI para ejecutar CodeBuild, consulta elReferencia de la línea de comandos.

compatible AWS SDK y herramientas para AWS CodeBuild

Los siguientes AWS SDK y herramientas son compatibles: CodeBuild

- SDK de AWS para C++. Para obtener más información, consulte la sección sobre el espacio de CodeBuild nombres Aws:: de la referencia de la API del AWS SDK para 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.
- El AWS SDK 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.
- El AWS SDK para el navegador y JavaScript el SDK para dentro. AWS JavaScript Node.js Para obtener más información, consulte la clase:. AWS CodeBuildsección del AWS SDK para la referencia de JavaScript API.
- SDK de AWS para .NET. Para obtener más información, consulte las secciones Amazon.CodeBuild y el espacio de Amazon.CodeBuild.Model nombres de la referencia de AWS API del SDK para .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.
- Las AWS herramientas para. PowerShell Para obtener más información, consulte la AWS CodeBuild sección de la referencia sobre AWS herramientas para PowerShell cmdlets.

Uso de este servicio con un AWS SDK

AWS Los kits de desarrollo de software (SDK) están disponibles para muchos lenguajes de programación populares. Cada SDK proporciona una API, ejemplos de código y documentación que facilitan a los desarrolladores la creación de aplicaciones en su lenguaje preferido.

Documentación de SDK	Ejemplos de código
AWS SDK para C++	AWS SDK para C++ ejemplos de código

Documentación de SDK	Ejemplos de código
AWS CLI	AWS CLI ejemplos de código
AWS SDK para Go	AWS SDK para Go ejemplos de código
AWS SDK para Java	AWS SDK para Java ejemplos de código
AWS SDK para JavaScript	AWS SDK para JavaScript ejemplos de código
AWS SDK para Kotlin	AWS SDK para Kotlin ejemplos de código
AWS SDK para .NET	AWS SDK para .NET ejemplos de código
AWS SDK para PHP	AWS SDK para PHP ejemplos de código
Herramientas de AWS para PowerShell	Herramientas de AWS para PowerShell ejemplos de código
AWS SDK para Python v4	AWS Ejemplos de código del SDK para Python v4
AWS SDK para Python (Boto3)	AWS SDK para Python (Boto3) ejemplos de código
AWS SDK para Ruby	AWS SDK para Ruby ejemplos de código
AWS SDK para Rust	AWS SDK para Rust ejemplos de código
AWS SDK para SAP ABAP	AWS SDK para SAP ABAP ejemplos de código
AWS SDK para Swift	AWS SDK para Swift ejemplos de código

Para obtener ejemplos específicos de este servicio, consulte Ejemplos de código para CodeBuild usar los AWS SDK.

> **Ejemplo de disponibilidad**
>
> ¿No encuentra lo que necesita? Solicite un ejemplo de código a través del enlace de Enviar comentarios que se encuentra al final de esta página.

Especifique el AWS CodeBuild punto de conexión

Puede usar el AWS Command Line Interface (AWS CLI) o uno de los AWS SDK para especificar el punto final utilizado por AWS CodeBuild. Hay un punto final para cada región en la que 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 conexión de FIPS, consulte Información general sobre FIPS 140-2.

La especificación de un punto de conexión es opcional. Si no indicas explícitamente CodeBuild qué punto final usar, el servicio usa el punto final asociado a la región que usa tu AWS cuenta. CodeBuildnunca usa un punto final FIPS de forma predeterminada. Si desea utilizar un punto de enlace de FIPS, debe asociarlo a CodeBuild mediante uno de los métodos siguientes.

> **Note**
>
> Puede usar un alias o un nombre de región para especificar un punto final mediante un AWS SDK. Si usa el AWS CLI, debe usar el nombre completo del punto final.

Para ver los puntos finales con los que se puede usar CodeBuild, consulte CodeBuild regiones y puntos finales.

Temas

- Especifique el AWS CodeBuild punto final (AWS CLI)
- Especifique el AWS CodeBuild punto final (AWS SDK)

Especifique el AWS CodeBuild punto final (AWS CLI)

Puede usar el AWS CLI para especificar el punto final a través del cual AWS CodeBuild se accede mediante el `--endpoint-url` argumento de cualquier CodeBuild comando. 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 `https://` al principio del punto final.

El `--endpoint-url` AWS CLI argumento está disponible para todos los AWS servicios. Para obtener más información sobre este y otros AWS CLI argumentos, consulte la referencia de AWS CLI comandos.

Especifique el AWS CodeBuild punto final (AWS SDK)

Puede usar un AWS SDK para especificar el punto final a través del cual AWS CodeBuild se accede. Aunque en este ejemplo se usa el AWS SDK para Java, puede especificar el punto final con los demás AWS SDK.

Utilice el `withEndpointConfiguration` método al crear el AWSCodeBuild cliente. Este es el formato que se debe usar:

```
AWSCodeBuild awsCodeBuild = AWSCodeBuildClientBuilder.standard().
    withEndpointConfiguration(new AwsClientBuilder.EndpointConfiguration("endpoint",
 "region")).
    withCredentials(new AWSStaticCredentialsProvider(sessionCredentials)).
    build();
```

Para obtener información al respecto`AWSCodeBuildClientBuilder`, consulte Class AWSCodeBuildClientBuilder.

Las credenciales utilizadas en `withCredentials` deben ser del tipo `AWSCredentialsProvider`. Para obtener más información, consulte Trabajar con AWS credenciales.

No lo incluya `https://` al principio del punto final.

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
Este de EE. UU. (Norte de Virginia)	us-east-1	codebuild-fips.us-east-1.amazonaws.com	us-east-1-fips
Este de EE. UU. (Ohio)	us-east-2	codebuild-fips.us-east-2.amazonaws.com	us-east-2-fips
Oeste de EE. UU. (Norte de California)	us-west-1	codebuild-fips.us-west-1.amazonaws.com	us-west-1-fips
Oeste de EE. UU. (Oregón)	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();
```

Uso AWS CodeBuild por AWS CodePipeline para probar el código y ejecutar compilaciones

Puedes automatizar tu proceso de lanzamiento usándolo AWS CodePipeline para probar tu código y ejecutar tus compilaciones 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
Crea una canalización de entrega continua (CD) CodePipeline que automatic e las compilaci ones con CodeBuild	• CodePipeline consola • AWS CLI • AWS SDKs	• 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 `create-pi peline` adecuada para su lenguaje de programac ión en la sección SDK de Herramientas de Amazon Web Services o consulte `CreatePipeline` en la Referencia de la API de AWS CodePipeline .

Tarea	Enfoques disponibles	Enfoques descritos en este tema
Agregue la automatización de pruebas y compilaciones CodeBuild a una canalizac ión existente en CodePipeline	• CodePipeline consola • AWS CLI • AWS SDKs	• Uso de la consola de CodePipeline para añadir la automatización de compilaciones • Uso de la consola de CodePipeline para añadir la automatización de pruebas • Para ello AWS CLI, puede adaptar la información de este tema para crear una canalización que contenga una acción de CodeBuild compilación o una acción de prueba. Para obtener más información, consulte Editar una canalización (AWS CLI) y la referencia a la estructura de 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 `update-pipeline` 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 .

Temas

- Requisitos previos
- Crear una canalización que utilice CodeBuild (CodePipeline consola)
- Cree una canalización que utilice CodeBuild (AWS CLI)
- Añadir una acción de CodeBuild compilación a una canalización (CodePipeline consola)
- Añadir una acción CodeBuild de prueba a una canalización (CodePipeline consola)

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 el nombre de la política gestionada `AWSCodePipelineFullAccess`

al usuario (o al grupo de IAM al que pertenece el usuario). No se recomienda usar una cuenta AWS raíz. 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 usa esta función de servicio para interactuar con otros AWS servicios AWS CodeBuild, incluso en tu nombre. Por ejemplo, para usar el AWS CLI para crear un rol de CodePipeline 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 este rol de CodePipeline servicio debe tener permiso en IAM para crear roles de servicio.

4. Tras crear un rol de CodePipeline servicio o identificar uno existente, debe añadir la política de roles de CodePipeline servicio predeterminada al rol de servicio, tal y como se describe

en Revisar la política de roles de CodePipeline servicio predeterminada en la Guía del AWS CodePipeline usuario, si aún no forma parte de la política del rol.

> **Note**
>
> La entidad de IAM que agrega esta política de roles de CodePipeline servicio debe tener permiso en IAM para agregar políticas de roles de servicio a los roles de servicio.

5. Crea y sube el código fuente a 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 la 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 ello OpsWorks, consulte Fuente de la aplicación y Uso CodePipeline con OpsWorks en la Guía del OpsWorks usuario.

Crear una canalización que utilice CodeBuild (CodePipeline consola)

Usa el siguiente procedimiento para crear una canalización que se utilice CodeBuild para compilar e implementar tu 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 Añadir una acción CodeBuild de prueba a una canalización (CodePipeline consola) 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 Añadir una acción CodeBuild de prueba a una canalización (CodePipeline consola) de este tema para añadir a la canalización una acción de prueba que use CodeBuild.

Para usar el asistente de creación de canalizaciones CodePipeline para crear una canalización que utilice CodeBuild

1. Inicia sesión en el Consola de administración de AWS mediante:

   - Tu cuenta AWS root. No se recomienda. Para obtener más información, consulte El usuario raíz de la cuenta en la Guía del usuario.
   - Un usuario administrador de tu AWS cuenta. Para obtener más información, consulte Crear el primer usuario y grupo Cuenta de AWS raíz en la Guía del usuario.
   - Un usuario de su AWS cuenta con permiso para usar el siguiente conjunto mínimo de acciones:

   ```
   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. Abre la AWS CodePipeline consola en https://console.aws.amazon.com/codesuite/codepipeline/home.
3. En el selector de AWS regiones, elige la AWS región en la que se encuentran AWS los recursos de tu proyecto de construcción. Esta debe ser una AWS región en la que CodeBuild se admita. Para obtener más información, consulte AWS CodeBuild en la Referencia general de Amazon Web Services.

4. Crear una canalización. Si aparece una página de CodePipeline información, elija 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 Existing service role (Rol de servicio existente)y elija después el rol de servicio de CodePipeline que creó o identificó en los requisitos previos de este tema.
7. En Artifact store (Almacén de artefactos), haga lo siguiente:

   - Elige la ubicación predeterminada para usar el almacén de artefactos predeterminado, como el depósito de artefactos S3 designado como predeterminado, para tu oleoducto en la AWS región que has seleccionado para tu oleoducto.
   - Elige una ubicación personalizada si ya tienes un almacén de artefactos creado (por ejemplo, un depósito de artefactos de S3) en la misma AWS región que tu oleoducto.

   > **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 requiere un almacén de artefactos independiente, como un depósito de S3, para cada canalización, en la misma AWS región que la canalización.
8. Elija Siguiente.
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 el caso de la clave de objeto S3, introduce el nombre del archivo que contiene el código fuente (por ejemplo,*`file-name`*`.zip`). Elija Siguiente.

- Si el código fuente está almacenado en un AWS CodeCommit repositorio, elija 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 Siguiente.
- Si el código fuente está almacenado en un GitHub repositorio, elija GitHub. Selecciona Conectarse a GitHub y sigue las instrucciones para autenticarte 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 Siguiente.

10. En la página Step 3: Add build stage (Paso 3: Agregar una fase de compilación), para Build provider (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 necesitas crear un nuevo proyecto de CodeBuild construcción, sigue las instrucciones que aparecen en la página Creación de un proyecto de compilación (consola) y vuelve a este procedimiento.

    Si eliges un proyecto de construcción existente, este debe tener ya definidos los ajustes de los artefactos de salida de compilación (aunque los CodePipeline anule). Para obtener más información, consulte Cambiar la configuración de un proyecto de compilación (consola).

    > **Important**
    >
    > Si habilitas los webhooks para un CodeBuild proyecto y el proyecto se usa como paso de compilación CodePipeline, se crean 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 estás usando CodePipeline, te recomendamos que deshabilites 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 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 CodePipeline consola, en la acción Crear, selecciona la información sobre herramientas. Anota el valor del artefacto de salida (por ejemplo, MyAppBuild).

    > **Note**
    >
    > También puedes obtener el artefacto resultante de la compilación seleccionando el enlace Construir artefactos en la página de detalles de la compilación de la CodeBuild consola. 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`*. Puedes usarlo AWS CLI para ejecutar el CodePipeline get-pipeline comando y obtener el nombre del bucket, donde *`my-pipeline-name`* se muestra el nombre de tu 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 has dado instrucciones CodePipeline para implementar el artefacto resultante de la compilación, sigue las instrucciones del proveedor de implementación para acceder al artefacto resultante de la compilación en los objetivos de la implementación.

Cree una canalización que utilice CodeBuild (AWS CLI)

Utilice el siguiente procedimiento para crear una canalización que se utilice CodeBuild para crear el código fuente.

AWS CLI Para crear una canalización que despliegue el código fuente creado o que solo lo pruebe, puedes adaptar las instrucciones de Editar una canalización (AWS CLI) y la referencia a la estructura de CodePipeline canalización de la guía del AWS CodePipeline usuario.

1. Crea o identifica 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 Cree un proyecto de compilación (AWS CLI).

2. Asegúrese de haberlo configurado AWS CLI con la clave de AWS acceso y la clave de acceso AWS secreta que corresponden 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. Cree un JSON-formatted archivo que represente la estructura de la canalización. Asigne al archivo el nombre `create-pipeline.json` o uno similar. Por ejemplo, esta JSON-formatted estructura crea una canalización con una acción de origen que hace referencia a un depósito 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 JSON-formatted datos:

- El valor de `roleArn` debe coincidir con el ARN del rol de CodePipeline servicio que creó o identificó 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 CodeBuild construcción que creó 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 solo describen una acción de origen y una acción de compilación, puede agregar acciones para actividades relacionadas con las pruebas, el despliegue del artefacto de salida de la compilación, la invocación de AWS Lambda funciones y más. 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 CodePipeline create-pipeline comando especificando el nombre del archivo:

```
aws codepipeline create-pipeline --cli-input-json file://create-pipeline.json
```

> **Note**
>
> Debe crear la canalización en una AWS región que CodeBuild sea compatible. Para obtener más información, consulte AWS CodeBuild en la Referencia general de Amazon Web Services.

Los JSON-formatted datos aparecen en la salida y CodePipeline crean la canalización.

5. Para obtener información sobre el estado de la canalización, ejecute el CodePipeline get-pipeline-state comando 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.

6. 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 Build artifacts (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).

7. 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>`. Puedes obtener el nombre del depósito desde el `create-pipeline.json` archivo o puedes ejecutar el CodePipeline get-pipeline comando para obtener el nombre del depósito.

```
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.

8. Abra la carpeta que coincide con el nombre de la canalización (por ejemplo, *<pipeline-name>*).

9. 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.

Añadir una acción de CodeBuild compilación a una canalización (CodePipeline consola)

1. Inicia sesión en el Consola de administración de AWS mediante:

   - Tu cuenta AWS root. No se recomienda. Para obtener más información, consulte El usuario raíz de la cuenta en la Guía del usuario.
   - Un usuario administrador de tu AWS cuenta. Para obtener más información, consulte Crear el primer usuario y grupo Cuenta de AWS raíz en la Guía del usuario.
   - Un usuario de su AWS cuenta con permiso para realizar el siguiente conjunto mínimo de acciones:

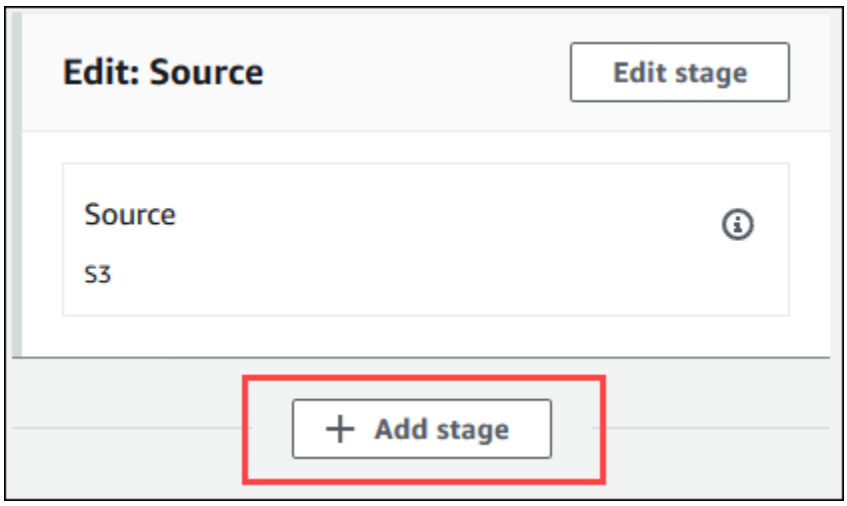
```
codepipeline:*
iam:ListRoles
iam:PassRole
s3:CreateBucket
s3:GetBucketPolicy
s3:GetObject
s3:ListAllMyBuckets
s3:ListBucket
s3:PutBucketPolicy
codecommit:ListBranches
codecommit:ListRepositories
codedeploy:GetApplication
codedeploy:GetDeploymentGroup
codedeploy:ListApplications
codedeploy:ListDeploymentGroups
elasticbeanstalk:DescribeApplications
elasticbeanstalk:DescribeEnvironments
lambda:GetFunctionConfiguration
lambda:ListFunctions
opsworks:DescribeStacks
opsworks:DescribeApps
opsworks:DescribeLayers
```

2. Abre la CodePipeline consola en https://console.aws.amazon.com/codesuite/codepipeline/home.
3. En el selector de AWS regiones, elige la AWS región en la que se encuentra tu gasoducto. Esta debe ser una región en la que CodeBuild esté admitido. Para obtener más información, consulte 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 del 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).

6. Elija Edit (Edición de).
7. 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. En Action provider (Proveedor de acción), seleccione 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 necesitas crear un nuevo proyecto de CodeBuild construcción, sigue las instrucciones que aparecen en la página Creación de un proyecto de compilación (consola) y vuelve a este procedimiento.

    Si eliges un proyecto de construcción existente, este debe tener ya definidos los ajustes de los artefactos de salida de compilación (aunque los CodePipeline 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 habilitas los webhooks para un CodeBuild proyecto y el proyecto se usa como paso de compilación CodePipeline, se crean 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 estás usando CodePipeline, te recomendamos que deshabilites los webhooks en. CodeBuild En la CodeBuild consola, desmarca la casilla Webhook. Para obtener más información, consulte 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. Seleccione Agregar 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 CodePipeline consola, en la acción Crear, selecciona la descripción emergente. Anota el valor del artefacto de salida (por ejemplo, MyAppBuild).

> **Note**
>
> También puedes obtener el artefacto resultante de la compilación seleccionando el enlace Construir artefactos en la página de detalles de la compilación de la CodeBuild consola. 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`*. Puedes usarlo AWS CLI para ejecutar el CodePipeline get-pipeline comando y 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 has dado instrucciones CodePipeline para implementar el artefacto resultante de la compilación, sigue las instrucciones del proveedor de implementación para acceder al artefacto resultante de la compilación en los objetivos de la implementación.

Añadir una acción CodeBuild de prueba a una canalización (CodePipeline consola)

1. Inicia sesión en el Consola de administración de AWS mediante:

    - Tu cuenta AWS root. No se recomienda. Para obtener más información, consulte El usuario raíz de la cuenta en la Guía del usuario.
    - Un usuario administrador de tu AWS cuenta. Para obtener más información, consulte Crear el primer usuario y grupo Cuenta de AWS raíz en la Guía del usuario.
    - Un usuario de su AWS cuenta con permiso para realizar el siguiente conjunto mínimo de acciones:

    ```
    codepipeline:*
    iam:ListRoles
    iam:PassRole
    s3:CreateBucket
    ```

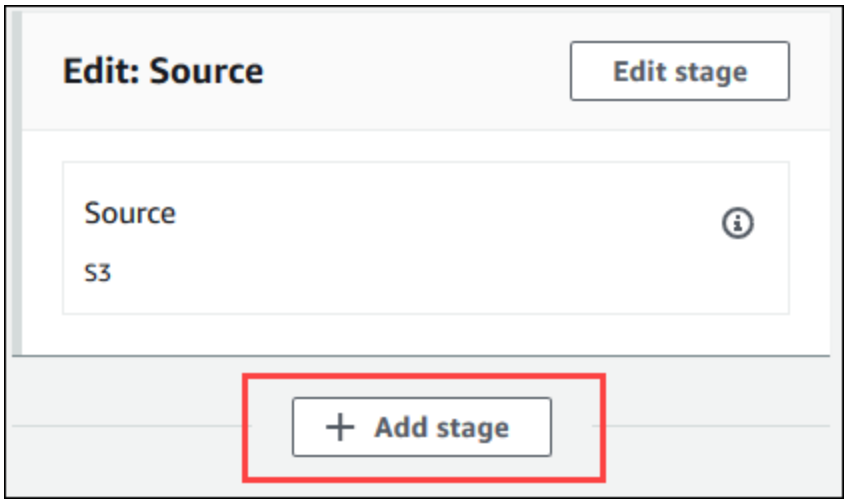
```
s3:GetBucketPolicy
s3:GetObject
s3:ListAllMyBuckets
s3:ListBucket
s3:PutBucketPolicy
codecommit:ListBranches
codecommit:ListRepositories
codedeploy:GetApplication
codedeploy:GetDeploymentGroup
codedeploy:ListApplications
codedeploy:ListDeploymentGroups
elasticbeanstalk:DescribeApplications
elasticbeanstalk:DescribeEnvironments
lambda:GetFunctionConfiguration
lambda:ListFunctions
opsworks:DescribeStacks
opsworks:DescribeApps
opsworks:DescribeLayers
```

2. Abre la CodePipeline consola en https://console.aws.amazon.com/codesuite/codepipeline/home.
3. En el selector de AWS regiones, elige la AWS región en la que se encuentra tu gasoducto. Esta debe ser una AWS región en la que CodeBuild esté admitido. 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 del 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 (Edición de).
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 Action provider (Proveedor de acción), en Test (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 necesitas crear un nuevo proyecto de CodeBuild construcción, sigue las instrucciones que aparecen en la página Creación de un proyecto de compilación (consola) y vuelve a este procedimiento.

    > **Important**
    >
    > Si habilitas los webhooks para un CodeBuild proyecto y el proyecto se usa como paso de compilación CodePipeline, se crean 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 estás usando CodePipeline, te recomendamos que deshabilites los webhooks en. CodeBuild En la CodeBuild consola, desmarca la casilla Webhook. Para obtener más información, consulte 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 de prueba del proceso, elige el CodeBuild hipervínculo para abrir la página del proyecto de compilación relacionado en la CodeBuild consola.
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, selecciona el hiperenlace 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 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 para tres de los repositorios de código fuente compatibles CodeBuild con Enterprise Server y Bitbucket. GitHub GitHub Si tu proyecto de compilación usa GitHub Enterprise Server, debes usar Codecov Enterprise.

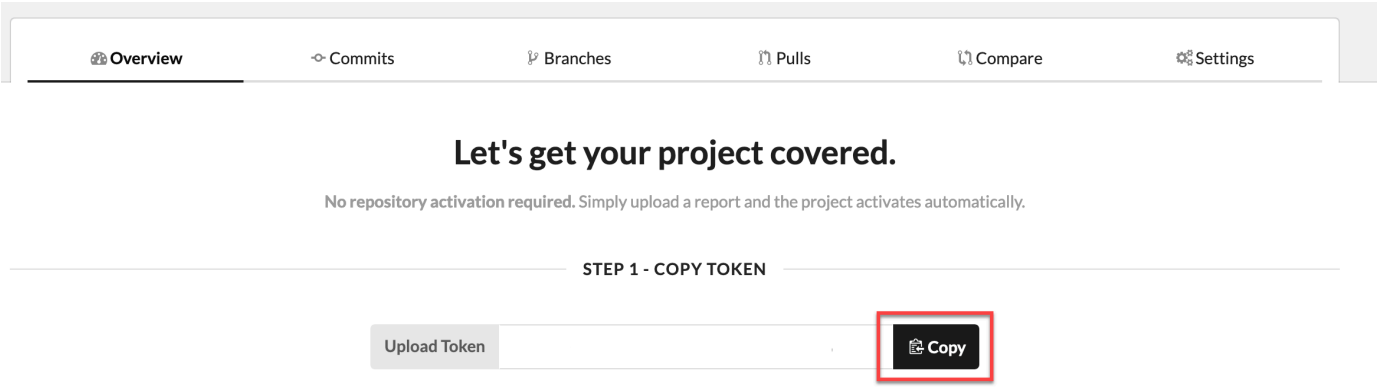
Cuando ejecutas una compilación de un CodeBuild proyecto que está integrado con Codecov, los informes de Codecov que analizan el código de tu repositorio se cargan en Codecov. Los registros de compilación incluyen un enlace a los informes. En este ejemplo, se muestra cómo se integra un proyecto de compilación 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

Usa el siguiente procedimiento para integrar Codecov en un proyecto de compilación.

Para integrar Codecov en un proyecto de compilación

1. Ve a un repositorio fuente GitHub o a Bitbucket https://codecov.io/signup y regístrate en él. Si utilizas GitHub Enterprise, consulta 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 JaCoCo complemento a `pom.xml` su 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 Ejecute AWS CodeBuild compilaciones manualmente y Registro AWS CodeBuild Llamadas a la API con AWS CloudTrail. La información de Codecov en los registros de compilación tiene el siguiente aspecto:

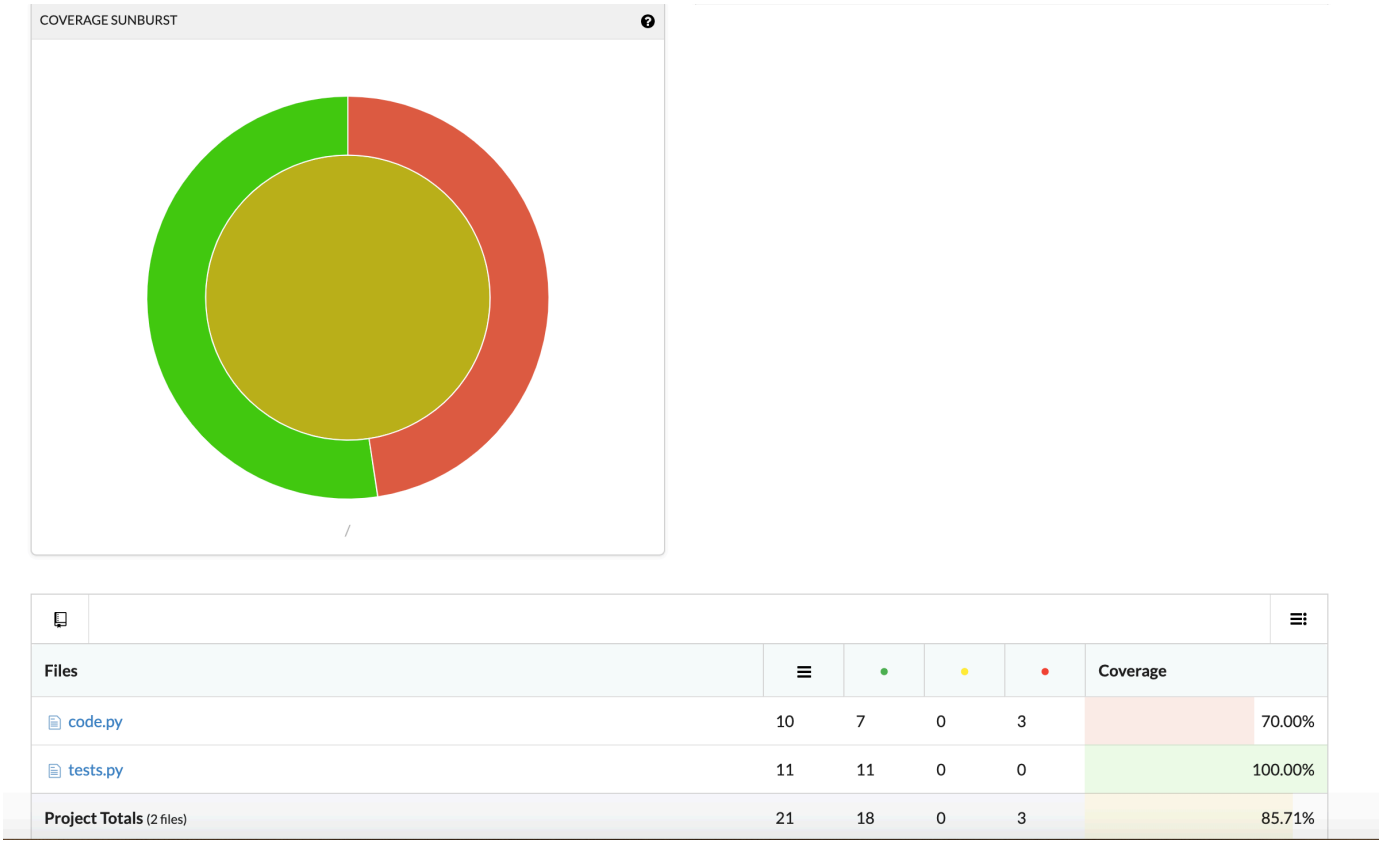
```
[Container] 2020/03/09 16:31:04 Running command bash my_script.sh


  _____          _
 / ____|        | |
| |     ___   __| | ___  ___ _____   __
| |    / _ \ / _` |/ _ \/ __/ _ \ \ / /
| |___| (_) | (_| |  __/ (_| (_) \ V /
 \_____\___/ \__,_|\___|\___\___/ \_/
                            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:



Uso AWS CodeBuild con Jenkins

Puedes usar el complemento de Jenkins para AWS CodeBuild integrarlo CodeBuild con tus trabajos de compilación de Jenkins. En lugar de enviar sus trabajos de compilación a los nodos de compilación de Jenkins, puede usar el complemento para enviar sus trabajos de compilación a CodeBuild. De esta manera, ya no tendrá que aprovisionar, configurar ni administrar nodos de compilación de Jenkins.

Temas

- Configuración de Jenkins
- Instalación del complemento
- Uso del complemento

Configuración de Jenkins

Para obtener información sobre cómo configurar Jenkins con el AWS CodeBuild complemento y descargar el código fuente del complemento, consulte. https://github.com/awslabs/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 fuentes externas a una VPC

1. Crea un proyecto en la CodeBuild consola. Para obtener más información, consulte Creación de un proyecto de compilación (consola).
   - Elige la AWS región en la que quieres ejecutar la compilación.
   - (Opcional) Establezca la configuración de Amazon VPC para permitir que el contenedor de CodeBuild compilación acceda a los recursos de su VPC.
   - Anote el nombre de su proyecto. Lo necesitará en el paso 3.
   - (Opcional) Si su repositorio de origen no es compatible de forma nativa con Amazon S3 CodeBuild, puede configurar Amazon S3 como el tipo de fuente de entrada para su proyecto.
2. En la consola de IAM, cree un usuario para que lo utilice el complemento de 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.

     JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Resource": [
                "arn:aws:logs:us-east-1:111122223333: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:us-east-1:111122223333:project/
{{projectName}}"
            ],
            "Action": [
                "codebuild:StartBuild",
                "codebuild:BatchGetBuilds",
                "codebuild:BatchGetProjects"
            ]
        }
```

```
    ]
}
```

3. Cree un proyecto de estilo libre en Jenkins.

   - En la página Configure, elija Add build step y después Run build on 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 tengas que instalar el GitHub complemento (o el complemento de Jenkins para tu proveedor de repositorios de origen) en tu 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 AWS CodeBuild complemento con el complemento Jenkins Pipeline

- En la página de tu proyecto de canalización de Jenkins, usa el generador de fragmentos para generar un script de canalización que se añada CodeBuild como un paso más en tu canalización. Debe generar un script similar a este:

```
awsCodeBuild projectName: 'project', credentialsType: 'keys', region: 'us-west-2',
 sourceControlType: 'jenkins'
```

Uso AWS CodeBuild con aplicaciones sin servidor

The AWS Serverless Application Model (AWS SAM) es un marco de código abierto para crear aplicaciones sin servidor. Para obtener más información, consulte el repositorio de modelos de aplicaciones AWS sin servidor en. GitHub

Se puede utilizar AWS CodeBuild para empaquetar e implementar aplicaciones sin servidor que sigan el AWS SAM estándar. Para el paso de implementación, CodeBuild puede usar AWS CloudFormation. Para automatizar la creación y el despliegue de aplicaciones sin servidor con CodeBuild y 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 sobre cómo empezar AWS CodeBuild, consulteIntroducción al AWS CodeBuild usando la consola.
- Para obtener información sobre la solución de problemas en CodeBuild, consulteResolución de problemas AWS CodeBuild.
- Para obtener información sobre las cuotas en CodeBuild, consulteCuotas para AWS CodeBuild.

Avisos de terceros para AWS CodeBuild para Windows

Cuando lo usas CodeBuild para compilaciones de Windows, tienes 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 base de Windows - microsoft-build-tools --versión 15.0.26320.2
- 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 - visualfsharptools, v 4.0
- 9) Imagen de Docker base de Windows - netfx-pcl-reference-assemblies-4.6
- 10) Imagen de Docker base de Windows - visualcppbuildtools, v 14.0.25420.1

- 11) Imagen de Docker base de Windows - microsoft-windows-netfx3-ondemand-package.cab
- 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 DE LICENCIA ADICIONALES (DERECHOS AND/OR DE USO)

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 complementarios de esta imagen de sistema operativo contenedor para contenedores Windows, también está sujeto a los términos de licencia subyacentes del software anfitrión 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 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

(los términos de la licencia están disponibles en: https://chocolatey.org/packages/git/2.16.2)

Licenciado bajo la Licencia Pública General GNU, versión 2, disponible en: https://www.gnu.org/licenses/old-licenses/gpl-2.0.html.

4) Imagen de Docker base de Windows - microsoft-build-tools --versión 15.0.26320.2

(los términos de la licencia están 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 dichos elementos, si se incluyen con el software, en sus equipos o en los de terceros, para depurar e implementar sus aplicaciones y las bases de datos que ha 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 que acompaña 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 hacer que una copia del código fuente esté 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 en 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 a la exportación, visite (también conocido como. 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 de los suplementos, actualizaciones, Internet-based servicios y servicios de soporte que utilice constituyen el acuerdo completo para el software y los servicios de soporte.
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 ESTÁ LICENCIADO "»AS-IS.» USTED CORRE EL RIESGO DE USARLO. MICROSOFT NO PROPORCIONA NINGUNA GARANTÍA EXPRESA, ASÍ COMO TAMPOCO AVALES O CONDICIONES. EN LA MEDIDA EN QUE LO PERMITAN LAS LEYES LOCALES, MICROSOFT EXCLUYE LAS GARANTÍAS IMPLÍCITAS DE COMERCIABILIDAD, IDONEIDAD PARA UN PROPÓSITO PARTICULAR Y. NON-INFRINGEMENT

11 LIMITACIÓN Y EXCLUSIÓN DE INDEMNIZACIONES. PUEDE RECUPERAR DE MICROSOFT Y SUS PROVEEDORES SOLO LOS DAÑOS DIRECTOS HASTA U.S. 5,00 DÓLARES. 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: VS2015_Update3_ _ ShellsRedist <ENU>

5) Imagen de Docker base de Windows - nuget.commandline --version 4.5.1

(los términos de licencia están disponibles en:) https://github.com/NuGet/Home/blob/dev/LICENSE.txt

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 servicios de soporte para este software tal y como se describe en www.support.microsoft. com/common/international.aspx.

8) Imagen de Docker base de Windows - visualfsharptools, 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 base de Windows - netfx-pcl-reference-assemblies-4.6

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,
- Internet-based servicios, 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 DE LICENCIA ADICIONALES, DERECHOS DE AND/OR USO.
   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 de los suplementos, actualizaciones, Internet-based servicios y servicios de soporte que utilice constituyen el acuerdo completo para el software y cualquier servicio de soporte que proporcionemos.
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. EFECTO 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 ESTÁ LICENCIADO "»AS-IS.» USTED CORRE EL RIESGO DE USARLO. 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 PERMITAN LAS LEYES LOCALES, MICROSOFT EXCLUYE LAS GARANTÍAS IMPLÍCITAS DE COMERCIABILIDAD, IDONEIDAD PARA UN PROPÓSITO PARTICULAR Y. NON-INFRINGEMENT

    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 SOLO LOS DAÑOS DIRECTOS HASTA U.S. 5,00 DÓLARES. 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

(los términos de licencia están 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 de Build Server listados en BuildServer.TXT archivos y and/or todos los archivos incluidos en la BuildeServer lista se encuentran de acuerdo con estos términos de licencia del 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 parar 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 que acompaña 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 a la 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 de los suplementos, actualizaciones, Internet-based servicios y servicios de soporte que utilice constituyen el acuerdo completo para el software y los servicios de soporte.
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.

10. EFECTO 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 ESTÁ LICENCIADO "»AS-IS.» USTED CORRE EL RIESGO DE USARLO. MICROSOFT NO PROPORCIONA NINGUNA GARANTÍA EXPRESA, ASÍ COMO TAMPOCO AVALES O CONDICIONES. EN LA MEDIDA EN QUE LO PERMITAN LAS LEYES LOCALES, MICROSOFT EXCLUYE LAS GARANTÍAS IMPLÍCITAS DE COMERCIABILIDAD, IDONEIDAD PARA UN PROPÓSITO PARTICULAR Y. NON-INFRINGEMENT

12. LIMITACIÓN Y EXCLUSIÓN DE INDEMNIZACIONES. PUEDE RECUPERAR DE MICROSOFT Y SUS PROVEEDORES SOLO LOS DAÑOS DIRECTOS HASTA U.S. 5,00 DÓLARES. 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 base de Windows - microsoft-windows-netfx3-ondemand-package.cab

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 servicios de soporte para este software tal y como se describe 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 comparativa 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 pueda tener con Microsoft, si divulga los resultados de dichas pruebas comparativas, Microsoft tendrá derecho a divulgar los resultados de las

pruebas comparativas que lleve a cabo con sus productos que compitan con el componente.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

Por la presente se concede permiso, de forma gratuita, a cualquier persona que obtenga una copia de este software y los archivos de documentación asociados (el «Software») para comerciar con el Software sin restricciones, incluidos, entre otros, los derechos de uso, copia, modificación, fusión, publicación, distribución, sublicencia y and/or venta de copias del Software y para permitir que las personas a las que se suministre el Software lo hagan, con sujeción 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 COMERCIABILIDAD, 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, EXTRACONTRACTUAL O DE OTRA ÍNDOLE, QUE SE DERIVEN DEL SOFTWARE O ESTÉN RELACIONADOS CON ÉL, CON SU UTILIZACIÓN O CON OTROS USOS DE ESTE.

Utilice claves de CodeBuild condición como variables de rol del servicio de IAM para controlar el acceso a la compilación

Con el ARN de CodeBuild compilación, puede restringir el acceso a los recursos de compilación mediante claves de contexto para limitar el acceso a los recursos en su función de CodeBuild servicio. Pues CodeBuild, las claves que se pueden usar para controlar el comportamiento de acceso a la compilación son `codebuild:buildArn` y`codebuild:projectArn`. Con el ARN del proyecto de compilación, puede verificar si una llamada al recurso proviene de un proyecto de

compilación específico. Para comprobarlo, utilice las claves de condición `codebuild:buildArn` o `codebuild:projectArn` en una política basada en identidades de IAM.

Para utilizar las claves de condición `codebuild:buildArn` o `codebuild:projectArn` en la política, inclúyalas como condición en cualquiera de los operadores de condición de ARN. El valor de la clave debe ser una variable de IAM que se resuelva en un ARN válido. En la política de ejemplo siguiente, el único acceso permitido será al proyecto de compilación con el ARN del proyecto para la variable de IAM `${codebuild:projectArn}` .

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::bucket-name/${codebuild:projectArn}/*"
        }
    ]
}
```

AWS CodeBuild claves de condición

AWS CodeBuild proporciona un conjunto de claves de condición que puede usar en las políticas de IAM para hacer cumplir las políticas de su organización sobre CodeBuild recursos como proyectos y flotas. Las claves de condición abarcan la mayoría de los contextos de solicitud de CodeBuild API, incluidos los ajustes de red, las configuraciones de credenciales y las restricciones informáticas.

Temas

- Aplicación de configuración de conectividad de VPC a proyectos y flotas
- Evitación de modificaciones no autorizadas en buildspec de un proyecto
- Restricción de tipos de computación para compilaciones
- Control de configuración de variables de entorno
- Uso de variables en nombres de claves de condición
- Comprobación de la existencia de atributos en solicitudes de API

Aplicación de configuración de conectividad de VPC a proyectos y flotas

Esta política permite a la persona que llama usar las VPC, las subredes y los grupos de seguridad seleccionados al crear CodeBuild proyectos y flotas. Para obtener más información sobre las claves de contexto multivalor, consulte Comparación entre claves de contexto multivalor. Single-valued

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "codebuild:CreateProject",
            "codebuild:CreateFleet"
        ],
        "Resource": "*",
        "Condition": {
            "ForAnyValue:StringEquals": {
                "codebuild:vpcConfig.vpcId": [
                    "vpc-01234567890abcdef",
                    "vpc-abcdef01234567890"
                ],
                "codebuild:vpcConfig.subnets": [
                    "subnet-1234abcd",
                    "subnet-5678abcd"
                ],
                "codebuild:vpcConfig.securityGroupIds": [
                    "sg-12345678abcdefghij",
                    "sg-01234567abcdefghij"
                ]
            }
        }
    }]
}
```

Evitación de modificaciones no autorizadas en buildspec de un proyecto

Esta política no permite a la persona que llama anular la buildspec en el campo `buildspecOverride`.

> **Note**
>
> La clave de condición de `codebuild:source.buildspec` solo admite el operador Null para comprobar la existencia del campo de la API. No evalúa el contenido de la especificación de compilación.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": "codebuild:StartBuild",
        "Resource": "*"
    }, {
        "Effect": "Deny",
        "Action": "codebuild:StartBuild",
        "Resource": "*",
        "Condition": {
            "Null": {
                "codebuild:source.buildspec": "false"
            }
        }
    }]
}
```

Restricción de tipos de computación para compilaciones

Esta política permite crear flotas que pueden compilarse únicamente con un tipo de instancia de computación `c5.large` o `m5.large`.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": "codebuild:CreateFleet",
```

```
        "Resource": "*",
        "Condition": {
            "ForAnyValue:StringEquals": {
                "codebuild:computeConfiguration.instanceType": ["c5.large",
 "m5.large"]
            }
        }
    }]
}
```

Control de configuración de variables de entorno

Esta política permite a la persona que llama anular la variable de entorno `STAGE` para que sea `BETA` o `GAMMA`. También deniega explícitamente anular `STAGE` por `PRODUCTION` y deniega la anulación de la variable de entorno `MY_APP_VERSION`. Para ver las claves de contexto de varios valores, consulta la sección Single-valued vs. claves de contexto de varios valores.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "codebuild:StartBuild"
            ],
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "codebuild:environment.environmentVariables/STAGE.value": [
                        "BETA",
                        "GAMMA"
                    ]
                }
            }
        },
        {
            "Effect": "Deny",
            "Action": [
                "codebuild:StartBuild"
```

```
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "codebuild:environment.environmentVariables/STAGE.value":
 "PRODUCTION"
                },
                "ForAnyValue:StringEquals": {
                    "codebuild:environment.environmentVariables.name": [
                        "MY_APP_VERSION"
                    ]
                }
            }
        }
    ]
}
```

Uso de variables en nombres de claves de condición

Puede usar variables en nombres de claves de condición, como `secondarySources/${sourceIdentifier}.location` y `secondaryArtifacts/${artifactIdentifier}.location`, donde puede especificar el origen secundario o el identificador de artefacto secundario en la política de IAM. La siguiente política permite a la persona que llama crear un proyecto con una ubicación de código fuente específica para el código fuente secundario `mySecondSource`.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "codebuild:CreateProject",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "codebuild:secondarySources/mySecondSource.location": "my-
source-location"
                }
```

```
            }
        }
    ]
}
```

Comprobación de la existencia de atributos en solicitudes de API

CodeBuild admite claves de condición para comprobar la existencia de algunos campos en la solicitud de API. La política aplica el requisito de VPC al crear o actualizar proyectos.

```
{
    "Version": "2012-10-17"             ,
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "codebuild:CreateProject",
            "codebuild:UpdateProject"
        ],
        "Resource": "*",
        "Condition": {
            "Null": {
                "codebuild:vpcConfig.vpcId": "false"
            }
        }
    }]
}
```

Ejemplos de código para CodeBuild usar los AWS SDK

Los siguientes ejemplos de código muestran cómo usarlo CodeBuild con un kit de desarrollo de AWS software (SDK).

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de este servicio con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Ejemplos de código

- Ejemplos básicos de CodeBuild uso de los AWS SDK
  - Acciones para CodeBuild usar los SDK AWS
    - Úselo CreateProject con un AWS SDK o CLI
    - Úselo ListBuilds con un AWS SDK o CLI
    - Úselo ListProjects con un AWS SDK o CLI
    - Úselo StartBuild con un AWS SDK o CLI

Ejemplos básicos de CodeBuild uso de los AWS SDK

Los siguientes ejemplos de código muestran cómo utilizar los aspectos básicos de AWS CodeBuild los AWS SDK.

Ejemplos

- Acciones para CodeBuild usar los SDK AWS
  - Úselo CreateProject con un AWS SDK o CLI
  - Úselo ListBuilds con un AWS SDK o CLI
  - Úselo ListProjects con un AWS SDK o CLI
  - Úselo StartBuild con un AWS SDK o CLI

Acciones para CodeBuild usar los SDK AWS

Los siguientes ejemplos de código muestran cómo realizar CodeBuild acciones individuales con los AWS SDK. Cada ejemplo incluye un enlace a GitHub, donde puede encontrar instrucciones para configurar y ejecutar el código.

Los siguientes ejemplos incluyen solo las acciones que se utilizan con mayor frecuencia. Para ver una lista completa, consulte la Referencia de la API de AWS CodeBuild.

Ejemplos

- Úselo CreateProject con un AWS SDK o CLI
- Úselo ListBuilds con un AWS SDK o CLI
- Úselo ListProjects con un AWS SDK o CLI
- Úselo StartBuild con un AWS SDK o CLI

Úselo **`CreateProject`** con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar `CreateProject`.

CLI

AWS CLI

Ejemplo 1: Para crear un proyecto de AWS CodeBuild compilación

El siguiente `create-project` ejemplo crea un proyecto de CodeBuild compilación con archivos fuente de un bucket de S3

```
aws codebuild create-project \
    --name "my-demo-project" \
    --source "{\"type\": \"S3\",\"location\": \"codebuild-us-west-2-123456789012-
input-bucket/my-source.zip\"}" \
    --artifacts {"\"type\": \"S3\",\"location\": \"codebuild-us-
west-2-123456789012-output-bucket\""} \
    --environment "{\"type\": \"LINUX_CONTAINER\",\"image\": \"aws/codebuild/
standard:1.0\",\"computeType\": \"BUILD_GENERAL1_SMALL\"}" \
    --service-role "arn:aws:iam::123456789012:role/service-role/my-codebuild-
service-role"
```

Salida:

```
{
    "project": {
        "arn": "arn:aws:codebuild:us-west-2:123456789012:project/my-demo-
project",
        "name": "my-cli-demo-project",
        "encryptionKey": "arn:aws:kms:us-west-2:123456789012:alias/aws/s3",
        "serviceRole": "arn:aws:iam::123456789012:role/service-role/my-codebuild-
service-role",
        "lastModified": 1556839783.274,
        "badge": {
            "badgeEnabled": false
        },
        "queuedTimeoutInMinutes": 480,
        "environment": {
            "image": "aws/codebuild/standard:1.0",
            "computeType": "BUILD_GENERAL1_SMALL",
            "type": "LINUX_CONTAINER",
            "imagePullCredentialsType": "CODEBUILD",
            "privilegedMode": false,
            "environmentVariables": []
        },
        "artifacts": {
            "location": "codebuild-us-west-2-123456789012-output-bucket",
            "name": "my-cli-demo-project",
            "namespaceType": "NONE",
            "type": "S3",
            "packaging": "NONE",
            "encryptionDisabled": false
        },
        "source": {
            "type": "S3",
            "location": "codebuild-us-west-2-123456789012-input-bucket/my-
source.zip",
            "insecureSsl": false
        },
        "timeoutInMinutes": 60,
        "cache": {
            "type": "NO_CACHE"
        },
        "created": 1556839783.274
    }
```

```
}
```

Ejemplo 2: Para crear un proyecto de AWS CodeBuild compilación con un archivo de entrada JSON para los parámetros

En el siguiente `create-project` ejemplo, se crea un proyecto de CodeBuild compilación pasando todos los parámetros necesarios a un archivo de entrada JSON. Cree la plantilla del archivo de entrada ejecutando el comando solo con `--generate-cli-skeleton parameter`.

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

El archivo JSON de entrada `create-project.json` contiene lo siguiente:

```
{
    "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:1.0",
        "computeType": "BUILD_GENERAL1_SMALL"
    },
    "serviceRole": "serviceIAMRole"
}
```

Salida:

```
{
    "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:1.0",
            "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"
    }
}
```

Para obtener más información, consulte Crear un proyecto de compilación (AWS CLI) en la Guía del AWS CodeBuild usuario.

- Para obtener más información sobre la API, consulte CreateProjectla Referencia de AWS CLI comandos.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Cree un proyecto.

```
import {
```

```
  ArtifactsType,
  CodeBuildClient,
  ComputeType,
  CreateProjectCommand,
  EnvironmentType,
  SourceType,
} from "@aws-sdk/client-codebuild";

// Create the AWS CodeBuild project.
export const createProject = async (
  projectName = "MyCodeBuilder",
  roleArn = "arn:aws:iam::xxxxxxxxxxxx:role/CodeBuildAdmin",
  buildOutputBucket = "xxxx",
  githubUrl = "https://...",
) => {
  const codeBuildClient = new CodeBuildClient({});

  const response = await codeBuildClient.send(
    new CreateProjectCommand({
      artifacts: {
        // The destination of the build artifacts.
        type: ArtifactsType.S3,
        location: buildOutputBucket,
      },
      // Information about the build environment. The combination of
 "computeType" and "type" determines the
      // requirements for the environment such as CPU, memory, and disk space.
      environment: {
        // Build environment compute types.
        // https://docs.aws.amazon.com/codebuild/latest/userguide/build-env-ref-
compute-types.html
        computeType: ComputeType.BUILD_GENERAL1_SMALL,
        // Docker image identifier.
        // See https://docs.aws.amazon.com/codebuild/latest/userguide/build-env-
ref-available.html
        image: "aws/codebuild/standard:7.0",
        // Build environment type.
        type: EnvironmentType.LINUX_CONTAINER,
      },
      name: projectName,
      // A role ARN with permission to create a CodeBuild project, write to the
 artifact location, and write CloudWatch logs.
      serviceRole: roleArn,
      source: {
```

```
       // The type of repository that contains the source code to be built.
       type: SourceType.GITHUB,
       // The location of the repository that contains the source code to be
built.
       location: githubUrl,
     },
   }),
 );
 console.log(response);
 //   {
 //     '$metadata': {
 //       httpStatusCode: 200,
 //       requestId: 'b428b244-777b-49a6-a48d-5dffedced8e7',
 //       extendedRequestId: undefined,
 //       cfId: undefined,
 //       attempts: 1,
 //       totalRetryDelay: 0
 //     },
 //     project: {
 //       arn: 'arn:aws:codebuild:us-east-1:xxxxxxxxxxxx:project/MyCodeBuilder',
 //       artifacts: {
 //         encryptionDisabled: false,
 //         location: 'xxxxxx-xxxxxxx-xxxxxx',
 //         name: 'MyCodeBuilder',
 //         namespaceType: 'NONE',
 //         packaging: 'NONE',
 //         type: 'S3'
 //       },
 //       badge: { badgeEnabled: false },
 //       cache: { type: 'NO_CACHE' },
 //       created: 2023-08-18T14:46:48.979Z,
 //       encryptionKey: 'arn:aws:kms:us-east-1:xxxxxxxxxxxx:alias/aws/s3',
 //       environment: {
 //         computeType: 'BUILD_GENERAL1_SMALL',
 //         environmentVariables: [],
 //         image: 'aws/codebuild/standard:7.0',
 //         imagePullCredentialsType: 'CODEBUILD',
 //         privilegedMode: false,
 //         type: 'LINUX_CONTAINER'
 //       },
 //       lastModified: 2023-08-18T14:46:48.979Z,
 //       name: 'MyCodeBuilder',
 //       projectVisibility: 'PRIVATE',
 //       queuedTimeoutInMinutes: 480,
```

```
  //       serviceRole: 'arn:aws:iam::xxxxxxxxxxxx:role/CodeBuildAdmin',
  //       source: {
  //         insecureSsl: false,
  //         location: 'https://...',
  //         reportBuildStatus: false,
  //         type: 'GITHUB'
  //       },
  //       timeoutInMinutes: 60
  //     }
  //   }
  return response;
};
```

- Para obtener más información, consulte la Guía para desarrolladores de AWS SDK para JavaScript.
- Para obtener más información sobre la API, consulta CreateProjectla Referencia AWS SDK para JavaScript de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de este servicio con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`ListBuilds`** con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar `ListBuilds`.

C++

SDK para C++

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
//! List the CodeBuild builds.
/*!
  \param sortType: 'SortOrderType' type.
```

```
  \param clientConfiguration: AWS client configuration.
  \return bool: Function succeeded.
 */
bool AwsDoc::CodeBuild::listBuilds(Aws::CodeBuild::Model::SortOrderType sortType,
                                   const Aws::Client::ClientConfiguration
 &clientConfiguration) {
    Aws::CodeBuild::CodeBuildClient codeBuildClient(clientConfiguration);

    Aws::CodeBuild::Model::ListBuildsRequest listBuildsRequest;
    listBuildsRequest.SetSortOrder(sortType);

    Aws::String nextToken; // Used for pagination.

    do {
        if (!nextToken.empty()) {
            listBuildsRequest.SetNextToken(nextToken);
        }

        Aws::CodeBuild::Model::ListBuildsOutcome listBuildsOutcome =
 codeBuildClient.ListBuilds(
                listBuildsRequest);

        if (listBuildsOutcome.IsSuccess()) {
            const Aws::Vector<Aws::String> &ids =
 listBuildsOutcome.GetResult().GetIds();
            if (!ids.empty()) {

                std::cout << "Information about each build:" << std::endl;
                Aws::CodeBuild::Model::BatchGetBuildsRequest getBuildsRequest;
                getBuildsRequest.SetIds(listBuildsOutcome.GetResult().GetIds());
                Aws::CodeBuild::Model::BatchGetBuildsOutcome getBuildsOutcome =
 codeBuildClient.BatchGetBuilds(
                        getBuildsRequest);

                if (getBuildsOutcome.IsSuccess()) {
                    const Aws::Vector<Aws::CodeBuild::Model::Build> &builds =
 getBuildsOutcome.GetResult().GetBuilds();
                    std::cout << builds.size() << " build(s) found." <<
 std::endl;
                    for (auto val: builds) {
                        std::cout << val.GetId() << std::endl;
                    }
                } else {
                    std::cerr << "Error getting builds"
```

```
                                << getBuildsOutcome.GetError().GetMessage() <<
 std::endl;
                    return false;
                }
            } else {
                std::cout << "No builds found." << std::endl;
            }

            // Get the next token for pagination.

            nextToken = listBuildsOutcome.GetResult().GetNextToken();
        } else {
            std::cerr << "Error listing builds"
                      << listBuildsOutcome.GetError().GetMessage()
                      << std::endl;
            return false;
        }

    } while (!nextToken.

            empty()

            );

    return true;
}
```

- Para obtener más información sobre la API, consulta ListBuildsla Referencia AWS SDK para C++ de la API.

CLI

AWS CLI

Para obtener una lista de los ID de AWS CodeBuild compilación.

En el siguiente `list-builds` ejemplo, se obtiene una lista de CodeBuild identificadores ordenados en orden ascendente.

```
aws codebuild list-builds --sort-order ASCENDING
```

La salida incluye un valor `nextToken` que indica que hay más salidas disponibles.

```
{
    "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"
    ]
}
```

Vuelva a ejecutar este comando y proporcione el valor `nextToken` de la respuesta anterior como parámetro para obtener la siguiente parte de la salida. Repita el procedimiento hasta que no reciba ningún valor `nextToken` en la respuesta.

```
aws codebuild list-builds --sort-order ASCENDING --next-
token 4AEA6u7J...The full token has been omitted for brevity...MzY2OA==
```

Siguiente parte de 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"
    ]
}
```

Para obtener más información, consulte Ver una lista de ID de compilación (AWS CLI) en la Guía del AWS CodeBuild usuario

- Para obtener más información sobre la API, consulte ListBuildsla Referencia de AWS CLI comandos.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de este servicio con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`ListProjects`** con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar `ListProjects`.

C++

SDK para C++

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
//! List the CodeBuild projects.
/*!
  \param sortType: 'SortOrderType' type.
  \param clientConfiguration: AWS client configuration.
  \return bool: Function succeeded.
 */
bool AwsDoc::CodeBuild::listProjects(Aws::CodeBuild::Model::SortOrderType
 sortType,
                                     const Aws::Client::ClientConfiguration
 &clientConfiguration) {
    Aws::CodeBuild::CodeBuildClient codeBuildClient(clientConfiguration);

    Aws::CodeBuild::Model::ListProjectsRequest listProjectsRequest;
    listProjectsRequest.SetSortOrder(sortType);

    Aws::String nextToken; // Next token for pagination.
    Aws::Vector<Aws::String> allProjects;

    do {
        if (!nextToken.empty()) {
            listProjectsRequest.SetNextToken(nextToken);
        }

        Aws::CodeBuild::Model::ListProjectsOutcome outcome =
 codeBuildClient.ListProjects(
                listProjectsRequest);

        if (outcome.IsSuccess()) {
```

```
            const Aws::Vector<Aws::String> &projects =
 outcome.GetResult().GetProjects();
            allProjects.insert(allProjects.end(), projects.begin(),
 projects.end());
            nextToken = outcome.GetResult().GetNextToken();
        }

        else {
            std::cerr << "Error listing projects" <<
 outcome.GetError().GetMessage()
                      << std::endl;
        }

    } while (!nextToken.empty());

    std::cout << allProjects.size() << " project(s) found." << std::endl;
    for (auto project: allProjects) {
        std::cout << project << std::endl;
    }

    return true;
}
```

- Para obtener más información sobre la API, consulta ListProjectsla Referencia AWS SDK para C++ de la API.

CLI

AWS CLI

Para obtener una lista de los nombres de los proyectos de AWS CodeBuild compilación.

En el siguiente `list-projects` ejemplo, se obtiene una lista de proyectos de CodeBuild construcción ordenados por nombre en orden ascendente.

```
aws codebuild list-projects --sort-by NAME --sort-order ASCENDING
```

La salida incluye un valor `nextToken` que indica que hay más salidas disponibles.

```
{
```

```
    "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"
    ]
}
```

Vuelva a ejecutar este comando y proporcione el valor `nextToken` de la respuesta anterior como parámetro para obtener la siguiente parte de la salida. Repita el procedimiento hasta que no reciba ningún valor `nextToken` en la respuesta.

```
aws codebuild list-projects  --sort-by NAME --sort-order ASCENDING --next-
token Ci33ACF6...The full token has been omitted for brevity...U+AkMx8=

{
    "projects": [
        "codebuild-demo-project100",
        "codebuild-demo-project101",

  ... The full list of build project names has been omitted for brevity ...
        "codebuild-demo-project122"

  ]
}
```

Para obtener más información, consulte Ver una lista de nombres de proyectos de compilación (AWS CLI) en la Guía del AWS CodeBuild usuario.

- Para obtener más información sobre la API, consulte ListProjectsla Referencia de AWS CLI comandos.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de este servicio con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`StartBuild`** con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar `StartBuild`.

C++

SDK para C++

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
//! Start an AWS CodeBuild project build.
/*!
  \param projectName: A CodeBuild project name.
  \param clientConfiguration: AWS client configuration.
  \return bool: Function succeeded.
 */
bool AwsDoc::CodeBuild::startBuild(const Aws::String &projectName,
                                   const Aws::Client::ClientConfiguration
 &clientConfiguration) {
    Aws::CodeBuild::CodeBuildClient codeBuildClient(clientConfiguration);

    Aws::CodeBuild::Model::StartBuildRequest startBuildRequest;
    startBuildRequest.SetProjectName(projectName);

    Aws::CodeBuild::Model::StartBuildOutcome outcome =
 codeBuildClient.StartBuild(
            startBuildRequest);

    if (outcome.IsSuccess()) {
        std::cout << "Successfully started build" << std::endl;
        std::cout << "Build ID: " << outcome.GetResult().GetBuild().GetId()
                  << std::endl;
    }

    else {
        std::cerr << "Error starting build" << outcome.GetError().GetMessage()
                  << std::endl;
    }

    return outcome.IsSuccess();
}
```

- Para obtener más información sobre la API, consulta StartBuildla Referencia AWS SDK para C++ de la API.

CLI

AWS CLI

Para empezar a ejecutar una compilación de un proyecto de AWS CodeBuild compilación.

En el siguiente `start-build` ejemplo, se inicia una compilación para el CodeBuild proyecto especificado. La compilación anula tanto la configuración del proyecto para el número de minutos que se permite poner en cola la compilación antes de que se agote el tiempo de espera como la configuración de artefacto del proyecto.

```
aws codebuild start-build \
    --project-name "my-demo-project" \
    --queued-timeout-in-minutes-override 5 \
    --artifacts-override {"\"type\": \"S3\",\"location\":
 \"arn:aws:s3:::artifacts-override\",\"overrideArtifactName\":true"}
```

Salida:

```
{
    "build": {
        "serviceRole": "arn:aws:iam::123456789012:role/service-role/my-codebuild-
service-role",
        "buildStatus": "IN_PROGRESS",
        "buildComplete": false,
        "projectName": "my-demo-project",
        "timeoutInMinutes": 60,
        "source": {
            "insecureSsl": false,
            "type": "S3",
            "location": "codebuild-us-west-2-123456789012-input-bucket/my-
source.zip"
        },
        "queuedTimeoutInMinutes": 5,
        "encryptionKey": "arn:aws:kms:us-west-2:123456789012:alias/aws/s3",
        "currentPhase": "QUEUED",
        "startTime": 1556905683.568,
        "environment": {
```

```
            "computeType": "BUILD_GENERAL1_MEDIUM",
            "environmentVariables": [],
            "type": "LINUX_CONTAINER",
            "privilegedMode": false,
            "image": "aws/codebuild/standard:1.0",
            "imagePullCredentialsType": "CODEBUILD"
        },
        "phases": [
            {
                "phaseStatus": "SUCCEEDED",
                "startTime": 1556905683.568,
                "phaseType": "SUBMITTED",
                "durationInSeconds": 0,
                "endTime": 1556905684.524
            },
            {
                "startTime": 1556905684.524,
                "phaseType": "QUEUED"
            }
        ],
        "logs": {
            "deepLink": "https://console.aws.amazon.com/cloudwatch/home?
region=us-west-2#logEvent:group=null;stream=null"
        },
        "artifacts": {
            "encryptionDisabled": false,
            "location": "arn:aws:s3:::artifacts-override/my-demo-project",
            "overrideArtifactName": true
        },
        "cache": {
            "type": "NO_CACHE"
        },
        "id": "my-demo-project::12345678-a1b2-c3d4-e5f6-11111EXAMPLE",
        "initiator": "my-aws-account-name",
        "arn": "arn:aws:codebuild:us-west-2:123456789012:build/my-demo-
project::12345678-a1b2-c3d4-e5f6-11111EXAMPLE"
    }
}
```


Para obtener más información, consulte Ejecutar una compilación (AWS CLI) en la Guía del AWS CodeBuild usuario.

- Para obtener más información sobre la API, consulte StartBuildla Referencia de AWS CLI comandos.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de este servicio con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

API de compilación pública

La AWS CodeBuild consola y el sitio web de resultados de compilación pública utilizan la API de compilación pública para obtener información sobre los proyectos de construcción pública. Para obtener más información, consulte the section called “Obtención de direcciones URL de proyectos de compilación públicos”.

> **Note**
>
> La API de compilación CodeBuild pública no está incluida en los AWS CLI AWS SDK.

Temas

- Acciones de construcción pública
- Tipos de datos de construcción pública

Acciones de construcción pública

> **Note**
>
> La API de compilación CodeBuild pública no está incluida en los AWS CLI AWS SDK.

La API de compilación CodeBuild pública admite las siguientes acciones:

Acciones

- DescribeBuildBatchesForPublicProject
- DescribeBuildsForPublicProject
- GetCloudWatchLogsForPublicBuild
- GetPresignedUrlsForPublicBuild
- GetPublicBuild
- GetPublicBuildBatch
- GetPublicProject

DescribeBuildBatchesForPublicProject

> **Note**
>
> Este elemento de API no está incluido en los AWS CLI AWS SDK.

Sintaxis de la solicitud

```
{
   "filter": {
      "status": "string"
   },
   "maxResults": number,
   "nextToken": "string",
   "publicProjectAlias": "string",
   "sortOrder": "string"
}
```

Parámetros de la solicitud

Para obtener información sobre los parámetros comunes a todas las acciones, consulte Parámetros comunes.

La solicitud acepta los siguientes datos en formato JSON.

> **Note**
>
> Los parámetros obligatorios se describen primero en la siguiente tabla.

público ProjectAlias

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Patrón: `^[0-9a-zA-Z%+=]+$`

Obligatorio: sí

filtrar

Especifica los filtros al recuperar compilaciones por lotes.

Tipo: objeto BuildBatchFilter

Obligatorio: no

MaxResults

Tipo: entero

Rango válido: valor mínimo de 1. Valor máximo de 100.

Obligatorio: no

NextToken

Tipo: cadena

Requerido: no

SortOrder

Tipo: cadena

Valores válidos: `ASCENDING | DESCENDING`

Obligatorio: no

Sintaxis de la respuesta

```
{
   "nextToken": "string",
   "publicBuildBatches": [
      {
         "buildBatchNumber": number,
         "buildBatchStatus": "string",
         "endTime": number,
         "publicBuildBatchAlias": "string",
         "sourceVersion": "string",
         "startTime": number
      }
   ]
```

```
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

NextToken

Tipo: cadena

público BuildBatches

Tipo: matriz de objetos BuildBatchForDescribeBuildBatchesPublic

Errores

Para obtener información sobre los errores comunes a todas las acciones, consulte Errores comunes.

InvalidInputException

Código de estado HTTP: 400

ResourceNotFoundException

Código de estado HTTP: 400

DescribeBuildsForPublicProject

> **Note**
>
> Este elemento de API no está incluido en los AWS CLI AWS SDK.

Sintaxis de la solicitud

```
{
   "maxResults": number,
   "nextToken": "string",
   "publicProjectAlias": "string",
   "sortOrder": "string"
}
```

Parámetros de la solicitud

Para obtener información sobre los parámetros comunes a todas las acciones, consulte Parámetros comunes.

La solicitud acepta los siguientes datos en formato JSON.

> **Note**
>
> Los parámetros obligatorios se describen primero en la siguiente tabla.

público ProjectAlias

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Patrón: `^[0-9a-zA-Z%+=]+$`

Obligatorio: sí

MaxResults

Tipo: entero

Rango válido: valor mínimo de 1. Valor máximo de 100.

Obligatorio: no

NextToken

Tipo: cadena

Requerido: no

SortOrder

Tipo: cadena

Valores válidos: `ASCENDING | DESCENDING`

Obligatorio: no

Sintaxis de la respuesta

```
{
   "builds": [
      {
         "artifacts": {
            "artifactIdentifier": "string",
            "packaging": "string",
            "type": "string"
         },
         "buildComplete": boolean,
         "buildNumber": number,
         "buildStatus": "string",
         "endTime": number,
         "environment": {
            "computeType": "string",
            "environmentVariables": [
               {
                  "name": "string",
                  "type": "string",
                  "value": "string"
               }
            ],
            "image": "string",
            "type": "string"
         },
         "id": "string",
         "initiator": "string",
```

```
        "logsStatus": {
           "cloudWatchLogsStatus": "string",
           "s3LogsStatus": "string"
        },
        "phases": [
           {
              "contexts": [
                 {
                    "message": "string",
                    "statusCode": "string"
                 }
              ],
              "durationInSeconds": number,
              "endTime": number,
              "phaseStatus": "string",
              "phaseType": "string",
              "startTime": number
           }
        ],
        "projectName": "string",
        "queuedTimeoutInMinutes": number,
        "resolvedSourceVersion": "string",
        "secondaryArtifacts": [
           {
              "artifactIdentifier": "string",
              "packaging": "string",
              "type": "string"
           }
        ],
        "secondarySources": [
           {
              "buildspec": "string",
              "gitCloneDepth": number,
              "gitSubmodulesConfig": {
                 "fetchSubmodules": boolean
              },
              "location": "string",
              "sourceIdentifier": "string",
              "type": "string"
           }
        ],
        "secondarySourceVersions": [
           {
              "sourceIdentifier": "string",
```

```
                "sourceVersion": "string"
            }
        ],
        "source": {
            "buildspec": "string",
            "gitCloneDepth": number,
            "gitSubmodulesConfig": {
                "fetchSubmodules": boolean
            },
            "location": "string",
            "sourceIdentifier": "string",
            "type": "string"
        },
        "sourceVersion": "string",
        "startTime": number,
        "timeoutInMinutes": number
    }
  ],
  "nextToken": "string"
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

construye

Tipo: matriz de objetos PublicBuild

NextToken

Tipo: cadena

Errores

Para obtener información sobre los errores comunes a todas las acciones, consulte Errores comunes.

InvalidInputException

Código de estado HTTP: 400

ResourceNotFoundException

Código de estado HTTP: 400

GetCloudWatchLogsForPublicBuild

> **Note**
>
> Este elemento de API no está incluido en los AWS CLI AWS SDK.

Sintaxis de la solicitud

```
{
   "maxResult": number,
   "nextToken": "string",
   "publicBuildAlias": "string"
}
```

Parámetros de la solicitud

Para obtener información sobre los parámetros comunes a todas las acciones, consulte Parámetros comunes.

La solicitud acepta los siguientes datos en formato JSON.

> **Note**
>
> Los parámetros obligatorios se describen primero en la siguiente tabla.

público BuildAlias

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Patrón: `^[0-9a-zA-Z%+=]+:\p{XDigit}{8}(-\p{XDigit}{4}){3}-\p{XDigit}{12}$`

Obligatorio: sí

Resultado máximo

Tipo: entero

Rango válido: valor mínimo de 1. Valor máximo de 10000.

Obligatorio: no

NextToken

Tipo: cadena

Requerido: no

Sintaxis de la respuesta

```
{
   "logs": [ "string" ],
   "nextToken": "string"
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

registros

Tipo: matriz de cadenas

NextToken

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1.

Errores

Para obtener información sobre los errores comunes a todas las acciones, consulte Errores comunes.

InvalidInputException

Código de estado HTTP: 400

ResourceNotFoundException

Código de estado HTTP: 400

GetPresignedUrlsForPublicBuild

> **Note**
>
> Este elemento de API no está incluido en los AWS CLI AWS SDK.

Sintaxis de la solicitud

```
{
   "publicBuildAlias": "string"
}
```

Parámetros de la solicitud

Para obtener información sobre los parámetros comunes a todas las acciones, consulte Parámetros comunes.

La solicitud acepta los siguientes datos en formato JSON.

> **Note**
>
> Los parámetros obligatorios se describen primero en la siguiente tabla.

público BuildAlias

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Patrón: `^[0-9a-zA-Z%+=]+:\p{XDigit}{8}(-\p{XDigit}{4}){3}-\p{XDigit}{12}$`

Obligatorio: sí

Sintaxis de la respuesta

```
{
   "artifacts": {
```

```
      "expiredAt": number,
      "identifier": "string",
      "md5Checksum": "string",
      "presignedUrl": "string",
      "s3Arn": "string",
      "sha256Checksum": "string"
   },
   "log": {
      "expiredAt": number,
      "identifier": "string",
      "md5Checksum": "string",
      "presignedUrl": "string",
      "s3Arn": "string",
      "sha256Checksum": "string"
   },
   "secondaryArtifacts": [
      {
         "expiredAt": number,
         "identifier": "string",
         "md5Checksum": "string",
         "presignedUrl": "string",
         "s3Arn": "string",
         "sha256Checksum": "string"
      }
   ]
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

artefactos

Tipo: objeto Descargable para S3

registro

Tipo: objeto Descargable para S3

SecondaryArtifacts

Tipo: matriz de objetos Descargable para S3

Errores

Para obtener información sobre los errores comunes a todas las acciones, consulte Errores comunes.

InvalidInputException

Código de estado HTTP: 400

ResourceNotFoundException

Código de estado HTTP: 400

GetPublicBuild

> **Note**
>
> Este elemento de API no está incluido en los AWS CLI AWS SDK.

Sintaxis de la solicitud

```
{
   "publicBuildAlias": "string"
}
```

Parámetros de la solicitud

Para obtener información sobre los parámetros comunes a todas las acciones, consulte Parámetros comunes.

La solicitud acepta los siguientes datos en formato JSON.

> **Note**
>
> Los parámetros obligatorios se describen primero en la siguiente tabla.

público BuildAlias

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Patrón: `^[0-9a-zA-Z%+=]+:\p{XDigit}{8}(-\p{XDigit}{4}){3}-\p{XDigit}{12}$`

Obligatorio: sí

Sintaxis de la respuesta

```
{
   "build": {
      "artifacts": {
         "artifactIdentifier": "string",
```

```
            "packaging": "string",
            "type": "string"
        },
        "buildComplete": boolean,
        "buildNumber": number,
        "buildStatus": "string",
        "endTime": number,
        "environment": {
            "computeType": "string",
            "environmentVariables": [
                {
                    "name": "string",
                    "type": "string",
                    "value": "string"
                }
            ],
            "image": "string",
            "type": "string"
        },
        "id": "string",
        "initiator": "string",
        "logsStatus": {
            "cloudWatchLogsStatus": "string",
            "s3LogsStatus": "string"
        },
        "phases": [
            {
                "contexts": [
                    {
                        "message": "string",
                        "statusCode": "string"
                    }
                ],
                "durationInSeconds": number,
                "endTime": number,
                "phaseStatus": "string",
                "phaseType": "string",
                "startTime": number
            }
        ],
        "projectName": "string",
        "queuedTimeoutInMinutes": number,
        "resolvedSourceVersion": "string",
        "secondaryArtifacts": [
```

```
            {
               "artifactIdentifier": "string",
               "packaging": "string",
               "type": "string"
            }
         ],
         "secondarySources": [
            {
               "buildspec": "string",
               "gitCloneDepth": number,
               "gitSubmodulesConfig": {
                  "fetchSubmodules": boolean
               },
               "location": "string",
               "sourceIdentifier": "string",
               "type": "string"
            }
         ],
         "secondarySourceVersions": [
            {
               "sourceIdentifier": "string",
               "sourceVersion": "string"
            }
         ],
         "source": {
            "buildspec": "string",
            "gitCloneDepth": number,
            "gitSubmodulesConfig": {
               "fetchSubmodules": boolean
            },
            "location": "string",
            "sourceIdentifier": "string",
            "type": "string"
         },
         "sourceVersion": "string",
         "startTime": number,
         "timeoutInMinutes": number
      }
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

construir

Tipo: objeto PublicBuild

Errores

Para obtener información sobre los errores comunes a todas las acciones, consulte Errores comunes.

InvalidInputException

Código de estado HTTP: 400

ResourceNotFoundException

Código de estado HTTP: 400

GetPublicBuildBatch

> **Note**
>
> Este elemento de API no está incluido en los AWS CLI AWS SDK.

Sintaxis de la solicitud

```
{
   "publicBuildBatchAlias": "string"
}
```

Parámetros de la solicitud

Para obtener información sobre los parámetros comunes a todas las acciones, consulte Parámetros comunes.

La solicitud acepta los siguientes datos en formato JSON.

> **Note**
>
> Los parámetros obligatorios se describen primero en la siguiente tabla.

público BuildBatchAlias

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Patrón: `^[0-9a-zA-Z%+=]+:[a-z0-9]{8}(-[a-z0-9]{4}){3}-[a-z0-9]{12}$`

Obligatorio: sí

Sintaxis de la respuesta

```
{
   "publicBuildBatch": {
      "artifacts": {
         "artifactIdentifier": "string",
```

```
      "packaging": "string",
      "type": "string"
   },
   "buildBatchConfig": {
      "combineArtifacts": boolean,
      "restrictions": {
         "computeTypesAllowed": [ "string" ],
         "maximumBuildsAllowed": number
      },
      "timeoutInMins": number
   },
   "buildBatchNumber": number,
   "buildBatchStatus": "string",
   "buildGroups": [
      {
         "currentBuildSummary": {
            "buildStatus": "string",
            "primaryArtifact": {
               "identifier": "string",
               "location": "string",
               "type": "string"
            },
            "publicBuildAlias": "string",
            "requestedOn": number,
            "secondaryArtifacts": [
               {
                  "identifier": "string",
                  "location": "string",
                  "type": "string"
               }
            ]
         },
         "dependsOn": [ "string" ],
         "identifier": "string",
         "ignoreFailure": boolean,
         "priorBuildSummaryList": [
            {
               "buildStatus": "string",
               "primaryArtifact": {
                  "identifier": "string",
                  "location": "string",
                  "type": "string"
               },
               "publicBuildAlias": "string",
```

```
            "requestedOn": number,
            "secondaryArtifacts": [
               {
                  "identifier": "string",
                  "location": "string",
                  "type": "string"
               }
            ]
         }
      ]
   }
],
"buildTimeoutInMinutes": number,
"complete": boolean,
"currentPhase": "string",
"endTime": number,
"environment": {
   "computeType": "string",
   "environmentVariables": [
      {
         "name": "string",
         "type": "string",
         "value": "string"
      }
   ],
   "image": "string",
   "type": "string"
},
"id": "string",
"initiator": "string",
"logsStatus": {
   "cloudWatchLogsStatus": "string",
   "s3LogsStatus": "string"
},
"phases": [
   {
      "contexts": [
         {
            "message": "string",
            "statusCode": "string"
         }
      ],
      "durationInSeconds": number,
      "endTime": number,
```

```
            "phaseStatus": "string",
            "phaseType": "string",
            "startTime": number
         }
      ],
      "projectName": "string",
      "publicBuildBatchAlias": "string",
      "queuedTimeoutInMinutes": number,
      "resolvedSourceVersion": "string",
      "secondaryArtifacts": [
         {
            "artifactIdentifier": "string",
            "packaging": "string",
            "type": "string"
         }
      ],
      "secondarySources": [
         {
            "buildspec": "string",
            "gitCloneDepth": number,
            "gitSubmodulesConfig": {
               "fetchSubmodules": boolean
            },
            "location": "string",
            "sourceIdentifier": "string",
            "type": "string"
         }
      ],
      "secondarySourceVersions": [
         {
            "sourceIdentifier": "string",
            "sourceVersion": "string"
         }
      ],
      "source": {
         "buildspec": "string",
         "gitCloneDepth": number,
         "gitSubmodulesConfig": {
            "fetchSubmodules": boolean
         },
         "location": "string",
         "sourceIdentifier": "string",
         "type": "string"
      },
```

```
      "sourceVersion": "string",
      "startTime": number
   }
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

público BuildBatch

Tipo: objeto PublicBuildBatch

Errores

Para obtener información sobre los errores comunes a todas las acciones, consulte Errores comunes.

InvalidInputException

Código de estado HTTP: 400

ResourceNotFoundException

Código de estado HTTP: 400

GetPublicProject

> **Note**
>
> Este elemento de API no está incluido en los AWS CLI AWS SDK.

Sintaxis de la solicitud

```
{
   "publicProjectAlias": "string"
}
```

Parámetros de la solicitud

Para obtener información sobre los parámetros comunes a todas las acciones, consulte Parámetros comunes.

La solicitud acepta los siguientes datos en formato JSON.

> **Note**
>
> Los parámetros obligatorios se describen primero en la siguiente tabla.

público ProjectAlias

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Patrón: `^[0-9a-zA-Z%+=]+$`

Obligatorio: sí

Sintaxis de la respuesta

```
{
   "project": {
      "artifacts": {
         "artifactIdentifier": "string",
```

```
      "location": "string",
      "type": "string"
   },
   "buildBatchConfig": {
      "combineArtifacts": boolean,
      "restrictions": {
         "computeTypesAllowed": [ "string" ],
         "maximumBuildsAllowed": number
      },
      "timeoutInMins": number
   },
   "concurrentBuildLimit": number,
   "description": "string",
   "environment": {
      "computeType": "string",
      "environmentVariables": [
         {
            "name": "string",
            "type": "string",
            "value": "string"
         }
      ],
      "image": "string",
      "type": "string"
   },
   "name": "string",
   "queuedTimeoutInMinutes": number,
   "secondaryArtifacts": [
      {
         "artifactIdentifier": "string",
         "location": "string",
         "type": "string"
      }
   ],
   "secondarySources": [
      {
         "buildspec": "string",
         "gitCloneDepth": number,
         "gitSubmodulesConfig": {
            "fetchSubmodules": boolean
         },
         "location": "string",
         "sourceIdentifier": "string",
         "type": "string"
```

```
            }
         ],
         "secondarySourceVersions": [
            {
               "sourceIdentifier": "string",
               "sourceVersion": "string"
            }
         ],
         "source": {
            "buildspec": "string",
            "gitCloneDepth": number,
            "gitSubmodulesConfig": {
               "fetchSubmodules": boolean
            },
            "location": "string",
            "sourceIdentifier": "string",
            "type": "string"
         },
         "sourceVersion": "string",
         "timeoutInMinutes": number,
         "webhook": {
            "branchFilter": "string",
            "buildType": "string",
            "filterGroups": [
               [
                  {
                     "excludeMatchedPattern": boolean,
                     "pattern": "string",
                     "type": "string"
                  }
               ]
            ],
            "payloadUrl": "string",
            "url": "string"
         }
      }
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

proyecto

Tipo: objeto PublicProject

Errores

Para obtener información sobre los errores comunes a todas las acciones, consulte Errores comunes.

InvalidInputException

Código de estado HTTP: 400

ResourceNotFoundException

Código de estado HTTP: 400

Tipos de datos de construcción pública

> **Note**
>
> La API de compilación CodeBuild pública no está incluida en los AWS CLI AWS SDK.

La API de compilación CodeBuild pública admite los siguientes tipos de datos:

Tipos de datos

- BuildBatchForDescribeBuildBatchesPublic
- PublicBuild
- PublicBuildArtifacts
- PublicBuildBatch
- PublicBuildGroup
- PublicBuildSummary
- PublicLogsStatus
- PublicProject
- PublicProjectArtifacts
- PublicProjectBuildBatchConfig

- PublicProjectEnvironment
- PublicProjectSource
- PublicWebhook
- Descargable para S3

BuildBatchForDescribeBuildBatchesPublic

> **Note**
>
> Este elemento de API no está incluido en los AWS CLI AWS SDK.

Contenido

> **Note**
>
> Los parámetros obligatorios se describen primero en la siguiente tabla.

buildBatchNumber

Tipo: largo

Obligatorio: no

buildBatchStatus

Tipo: cadena

Valores válidos: `PENDING | SUCCEEDED | FAILED | FAULT | TIMED_OUT | IN_PROGRESS | STOPPED`

Obligatorio: no

endTime

Tipo: marca temporal

Obligatorio: no

público BuildBatchAlias

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Patrón: `^[0-9a-zA-Z%+=]+:[a-z0-9]{8}(-[a-z0-9]{4}){3}-[a-z0-9]{12}$`

Obligatorio: no

sourceVersion

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1.

Obligatorio: no

startTime

Tipo: marca temporal

Obligatorio: no

PublicBuild

> **Note**
>
> Este elemento de API no está incluido en los AWS CLI AWS SDK.

Contenido

> **Note**
>
> Los parámetros obligatorios se describen primero en la siguiente tabla.

artifacts

Tipo: objeto PublicBuildArtifacts

Obligatorio: no

BuildComplete

Tipo: Booleano

Obligatorio: no

Número de compilación

Tipo: largo

Obligatorio: no

Estado de compilación

Tipo: cadena

Valores válidos: `PENDING | SUCCEEDED | FAILED | FAULT | TIMED_OUT | IN_PROGRESS | STOPPED`

Obligatorio: no

endTime

Tipo: marca temporal

Obligatorio: no

environment

Tipo: objeto PublicProjectEnvironment

Obligatorio: no

id

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1.

Obligatorio: no

iniciador

Tipo: cadena

Requerido: no

Estado de los registros

Tipo: objeto PublicLogsStatus

Obligatorio: no

fases

Tipo: matriz de objetos BuildPhase

Obligatorio: no

ProjectName

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1.

Obligatorio: no

en cola TimeoutInMinutes

Tipo: entero

Obligatorio: no

resuelto SourceVersion

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1.

Obligatorio: no

SecondaryArtifacts

Tipo: matriz de objetos PublicBuildArtifacts

Obligatorio: no

secondarySources

Tipo: matriz de objetos PublicProjectSource

Miembros de la matriz: número mínimo de 0 artículos. Número máximo de 12 artículos.

Obligatorio: no

secundario SourceVersions

Tipo: matriz de objetos ProjectSourceVersion

Miembros de la matriz: número mínimo de 0 artículos. Número máximo de 12 artículos.

Obligatorio: no

origen

Tipo: objeto PublicProjectSource

Obligatorio: no

sourceVersion

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1.

Obligatorio: no

startTime

Tipo: marca temporal

Obligatorio: no

timeoutInMinutes

Tipo: entero

Obligatorio: no

PublicBuildArtifacts

> **Note**
>
> Este elemento de API no está incluido en los AWS CLI AWS SDK.

Contenido

> **Note**
>
> Los parámetros obligatorios se describen primero en la siguiente tabla.

Identificador de artefacto

Tipo: cadena

Requerido: no

embalaje

Tipo: cadena

Requerido: no

type

Tipo: cadena

Valores válidos: `CODEPIPELINE | S3 | NO_ARTIFACTS | DOCKER_IMAGE`

Obligatorio: no

PublicBuildBatch

> **Note**
>
> Este elemento de API no está incluido en los AWS CLI AWS SDK.

Contenido

> **Note**
>
> Los parámetros obligatorios se describen primero en la siguiente tabla.

artifacts

Tipo: objeto PublicBuildArtifacts

Obligatorio: no

buildBatchConfig

Tipo: objeto PublicProjectBuildBatchConfig

Obligatorio: no

buildBatchNumber

Tipo: largo

Obligatorio: no

buildBatchStatus

Tipo: cadena

Valores válidos: `PENDING | SUCCEEDED | FAILED | FAULT | TIMED_OUT | IN_PROGRESS | STOPPED`

Obligatorio: no

Cree grupos

Tipo: matriz de objetos PublicBuildGroup

Obligatorio: no

buildTimeoutInMinutes

Tipo: entero

Obligatorio: no

completo

Tipo: Booleano

Obligatorio: no

Fase actual

Tipo: cadena

Requerido: no

endTime

Tipo: marca temporal

Obligatorio: no

environment

Tipo: objeto PublicProjectEnvironment

Obligatorio: no

id

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1.

Obligatorio: no

iniciador

Tipo: cadena

Requerido: no

Estado de los registros

Tipo: objeto PublicLogsStatus

Obligatorio: no

fases

Tipo: matriz de objetos BuildBatchPhase

Obligatorio: no

ProjectName

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1.

Obligatorio: no

público BuildBatchAlias

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Patrón: `^[0-9a-zA-Z%+=]+:[a-z0-9]{8}(-[a-z0-9]{4}){3}-[a-z0-9]{12}$`

Obligatorio: no

en cola TimeoutInMinutes

Tipo: entero

Obligatorio: no

resuelto SourceVersion

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1.

Obligatorio: no

SecondaryArtifacts

Tipo: matriz de objetos PublicBuildArtifacts

Obligatorio: no

secondarySources

Tipo: matriz de objetos PublicProjectSource

Miembros de la matriz: número mínimo de 0 artículos. Número máximo de 12 artículos.

Obligatorio: no

secundario SourceVersions

Tipo: matriz de objetos ProjectSourceVersion

Miembros de la matriz: número mínimo de 0 artículos. Número máximo de 12 artículos.

Obligatorio: no

origen

Tipo: objeto PublicProjectSource

Obligatorio: no

sourceVersion

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1.

Obligatorio: no

startTime

Tipo: marca temporal

Obligatorio: no

PublicBuildGroup

> **Note**
>
> Este elemento de API no está incluido en los AWS CLI AWS SDK.

Contenido

> **Note**
>
> Los parámetros obligatorios se describen primero en la siguiente tabla.

currentBuildSummary

Tipo: objeto PublicBuildSummary

Obligatorio: no

Depende de

Tipo: matriz de cadenas

Limitaciones de longitud: longitud mínima de 1.

Obligatorio: no

identifier

Tipo: cadena

Requerido: no

Ignorar el error

Tipo: Booleano

Obligatorio: no

anterior BuildSummaryList

Tipo: matriz de objetos PublicBuildSummary

Obligatorio: no

PublicBuildSummary

> **Note**
>
> Este elemento de la API no está incluido en los AWS CLI AWS SDK.

Contenido

> **Note**
>
> Los parámetros obligatorios se describen primero en la siguiente tabla.

BuildStatus

Tipo: cadena

Requerido: no

Artefacto principal

Representa un artefacto de construcción resuelto. Un artefacto resuelto es un artefacto que se crea e implementa en el destino, como Amazon S3.

Tipo: objeto ResolvedArtifact

Obligatorio: no

público BuildAlias

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Patrón: `^[0-9a-zA-Z%+=]+:\p{XDigit}{8}(-\p{XDigit}{4}){3}-\p{XDigit}{12}$`

Obligatorio: no

Solicitado el

Tipo: marca temporal

Obligatorio: no

Artefacto secundario

Tipo: matriz de objetos ResolvedArtifact

Obligatorio: no

PublicLogsStatus

> **Note**
>
> Este elemento de API no está incluido en los AWS CLI AWS SDK.

Contenido

> **Note**
>
> Los parámetros obligatorios se describen primero en la siguiente tabla.

nube WatchLogsStatus

Tipo: cadena

Valores válidos: `ENABLED | DISABLED`

Obligatorio: no

s3LogsStatus

Tipo: cadena

Valores válidos: `ENABLED | DISABLED`

Obligatorio: no

PublicProject

> **Note**
>
> Este elemento de API no está incluido en los AWS CLI AWS SDK.

Contenido

> **Note**
>
> Los parámetros obligatorios se describen primero en la siguiente tabla.

artifacts

Tipo: objeto PublicProjectArtifacts

Obligatorio: no

buildBatchConfig

Tipo: objeto PublicProjectBuildBatchConfig

Obligatorio: no

simultáneo BuildLimit

Tipo: entero

Obligatorio: no

description

Tipo: cadena

Limitaciones de longitud: longitud mínima de 0. La longitud máxima es de 255 caracteres.

Obligatorio: no

environment

Tipo: objeto PublicProjectEnvironment

Obligatorio: no

name

Tipo: cadena

Limitaciones de longitud: longitud mínima de 2. La longitud máxima es de 150 caracteres.

Patrón: `[A-Za-z0-9][A-Za-z0-9\-_]{1,254}`

Obligatorio: no

en cola TimeoutInMinutes

Tipo: entero

Rango válido: valor mínimo de 5. Valor máximo de 480.

Obligatorio: no

SecondaryArtifacts

Tipo: matriz de objetos PublicProjectArtifacts

Miembros de la matriz: número mínimo de 0 artículos. Número máximo de 12 artículos.

Obligatorio: no

secondarySources

Tipo: matriz de objetos PublicProjectSource

Miembros de la matriz: número mínimo de 0 artículos. Número máximo de 12 artículos.

Obligatorio: no

secundario SourceVersions

Tipo: matriz de objetos ProjectSourceVersion

Miembros de la matriz: número mínimo de 0 artículos. Número máximo de 12 artículos.

Obligatorio: no

origen

Tipo: objeto PublicProjectSource

Obligatorio: no

sourceVersion

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1.

Obligatorio: no

timeoutInMinutes

Tipo: entero

Rango válido: valor mínimo de 5. Valor máximo de 480.

Obligatorio: no

webhook

Tipo: objeto PublicWebhook

Obligatorio: no

PublicProjectArtifacts

> **Note**
>
> Este elemento de API no está incluido en los AWS CLI AWS SDK.

Contenido

> **Note**
>
> Los parámetros obligatorios se describen primero en la siguiente tabla.

type

Tipo: cadena

Valores válidos: `CODEPIPELINE | S3 | NO_ARTIFACTS | DOCKER_IMAGE`

Obligatorio: sí

Identificador de artefacto

Tipo: cadena

Requerido: no

location

Tipo: cadena

Obligatorio: no

PublicProjectBuildBatchConfig

> **Note**
>
> Este elemento de API no está incluido en los AWS CLI AWS SDK.

Contenido

> **Note**
>
> Los parámetros obligatorios se describen primero en la siguiente tabla.

combineArtifacts

Tipo: Booleano

Obligatorio: no

restricciones

Especifica restricciones para la compilación por lotes.

Tipo: objeto BatchRestrictions

Obligatorio: no

timeoutInMins

Tipo: entero

Obligatorio: no

PublicProjectEnvironment

> **Note**
>
> Este elemento de API no está incluido en los AWS CLI AWS SDK.

Contenido

> **Note**
>
> Los parámetros obligatorios se describen primero en la siguiente tabla.

ComputEtype

Tipo: cadena

Valores válidos: `BUILD_GENERAL1_SMALL | BUILD_GENERAL1_MEDIUM | BUILD_GENERAL1_LARGE | BUILD_GENERAL1_2XLARGE`

Obligatorio: no

Variables de entorno

Tipo: matriz de objetos EnvironmentVariable

Obligatorio: no

image

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1.

Obligatorio: no

type

Tipo: cadena

Valores válidos: `WINDOWS_CONTAINER | WINDOWS_SERVER_2019_CONTAINER | LINUX_CONTAINER | LINUX_GPU_CONTAINER | ARM_CONTAINER | MAC`

Obligatorio: no

PublicProjectSource

> **Note**
>
> Este elemento de API no está incluido en los AWS CLI AWS SDK.

Contenido

> **Note**
>
> Los parámetros obligatorios se describen primero en la siguiente tabla.

type

Tipo: cadena

Valores válidos: `NO_SOURCE | CODECOMMIT | CODEPIPELINE | GITHUB | S3 | BITBUCKET | GITHUB_ENTERPRISE | GITLAB | GITLAB_SELF_MANAGED`

Obligatorio: sí

buildspec

Tipo: cadena

Requerido: no

git CloneDepth

Tipo: entero

Rango válido: valor mínimo de 0.

Obligatorio: no

regalo SubmodulesConfig

Información sobre la configuración de los submódulos de Git para un proyecto de AWS CodeBuild compilación.

Tipo: objeto GitSubmodulesConfig

Obligatorio: no

location

Tipo: cadena

Requerido: no

Identificador de origen

Tipo: cadena

Obligatorio: no

PublicWebhook

> **Note**
>
> Este elemento de API no está incluido en los AWS CLI AWS SDK.

Contenido

> **Note**
>
> Los parámetros obligatorios se describen primero en la siguiente tabla.

BranchFilter

Tipo: cadena

Requerido: no

Tipo de compilación

Tipo: cadena

Requerido: no

Grupos de filtros

Tipo: matriz de matrices de objetos WebhookFilter

Obligatorio: no

URL de carga útil

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1.

Obligatorio: no

url

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1.

Obligatorio: no

Descargable para S3

> **Note**
>
> Este elemento de la API no está incluido en los SDK AWS CLI . AWS

Contenido

> **Note**
>
> Los parámetros obligatorios se describen primero en la siguiente tabla.

Expiró el día

Tipo: marca temporal

Obligatorio: no

identifier

Tipo: cadena

Requerido: no

Suma de verificación MD5

Tipo: cadena

Requerido: no

URL prefirmada

Tipo: cadena

Requerido: no

s3ARN

Tipo: cadena

Requerido: no

Suma de comprobación SHA256

Tipo: cadena

Obligatorio: no

Resolución 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, consulteRegistro y supervisión.

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 fallar si los nombres de los archivos no tienen el signo «-U.S». caracteres que no pertenecen al inglés (Estados Unidos)
- Las compilaciones pueden fallar al obtener parámetros del almacén de parámetros de Amazon EC2
- No se puede acceder al filtro de sucursales de la consola 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: «Falló el clon de Git: no se puede acceder 'URL de tu repositorio': Problema con el certificado 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 de las compilaciones almacenadas en caché local
- AccessDenied: El propietario del bucket del grupo de informes no coincide con el propietario del bucket de S3...
- Error: «Sus credenciales carecen de uno o más ámbitos de privilegio obligatorios» al crear un proyecto con CodeBuild AWS CodeConnections
- Error: “Sorry, no terminal at all requested - can't get input” al compilar con el comando de instalación de Ubuntu
- Las compilaciones de Bitbucket fallan debido a errores de autenticación tras usar Secrets Manager OAuth

Apache Maven crea artefactos de referencia en un repositorio incorrecto

Problema: Cuando usas Maven con un entorno de compilación AWS CodeBuild de 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 de compilación de CodeBuild-provided Java incluyen un nombre de archivo `settings.xml` que está preinstalado en el directorio del entorno de compilación. `/root/.m2` Este `settings.xml` archivo contiene las siguientes declaraciones, que indican a Maven que siempre extraiga las dependencias de compilación y complementos del repositorio central seguro de Maven 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 copies tu `settings.xml` archivo al 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 fallar si los nombres de los archivos no tienen el signo «-U.S». caracteres que no pertenecen al inglés (Estados Unidos)

Problema: cuando ejecutas una compilación que usa archivos con nombres de archivo que no contienen ningún valorU.S. que no pertenecen al inglés de Estados Unidos (por ejemplo, caracteres chinos), la compilación no se realiza correctamente.

Causa posible: los entornos de compilación proporcionados por AWS CodeBuild tienen su configuración regional predeterminada establecida en. `POSIX` `POSIX`los ajustes de localización son menos compatibles con CodeBuild los nombres de archivo que no contienen esa información. U.S del inglés de Estados Unidos, lo que puede provocar que las compilaciones relacionadas no se realicen correctamente.

Solución recomendada: añada los siguientes comandos a la sección `pre_build` del archivo buildspec. Estos comandos hacen que el entorno de compilación utilice U.S. El inglés UTF-8 para su configuración de localización, que es más compatible con CodeBuild los nombres de archivo que no contienen la palabra «»U.S. caracteres que no pertenecen al inglés (Estados Unidos)

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 -f noninteractive 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/`:

  JSON

  ```
  {
      "Version":"2012-10-17",
      "Statement": [
          {
              "Action": "ssm:GetParameters",
              "Effect": "Allow",
              "Resource": "arn:aws:ssm:us-east-1:111122223333: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:

  JSON

  ```
  {
  ```

```
    "Version":"2012-10-17",
    "Statement": [
        {
            "Action": "ssm:GetParameters",
            "Effect": "Allow",
            "Resource": "arn:aws:ssm:us-
east-1:111122223333:parameter/PARAMETER_NAME"
        }
    ]
}
```

No se puede acceder al filtro de sucursales de la consola 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 webhooks (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, consulta el informe BuildStatus en la referencia AWS CodeBuild de la API.

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 0.2 de buildspec, que resuelve 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` `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 la 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 se encuentre 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 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 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:

- Para Microsoft Windows, utilice un contenedor de Windows con un sistema operativo contenedor que sea la versión:10.0.x microsoft/windowsservercore (por ejemplo, :10.0.14393.2125). microsoft/windowsservercore
- 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, debe habilitar el modo privilegiado y actualizar la especificación de compilación de acuerdo con las siguientes instrucciones.

Siga estos pasos para ejecutar la compilación en modo privilegiado:

1. https://console.aws.amazon.com/codebuild/Abra la consola en. CodeBuild
2. En el panel de navegación, elija Proyectos de compilación y, a continuación, elija el proyecto de compilación.
3. En Edit (Editar), seleccione Environment (Entorno).
4. Elija Configuración adicional.
5. En Privilegiado, seleccione Habilite esta marca si desea compilar imágenes de Docker o desea que sus compilaciones obtengan privilegios elevados.
6. Seleccione Update environment (Actualizar entorno).
7. Seleccione Start build (Comenzar compilación) para volver a intentar crear la compilación.

También tendrá que iniciar el daemon de Docker dentro de su contenedor. La fase `install` de la especificación de compilación puede ser similar a la siguiente.

```
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` añada 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 AWS CodeBuild, consulteDocker en una muestra de imagen personalizada 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.
- La AWS CodeBuild mayúscula de la función de servicio asociada al proyecto de construcción no coincide con la función de IAM real.

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. Tenga en cuenta que los roles de IAM distinguen entre mayúsculas y minúsculas, por lo que debe comprobar que el uso de mayúsculas y minúsculas del rol de IAM sea correcto.
- 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 CodeBuild Permitir interactuar con otros AWS servicios.

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: «Falló el clon de Git: no se puede acceder **`'URL de tu repositorio'`**: Problema con el certificado 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 AWS CodeBuild proyecto. 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 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, incluya 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 obtener 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 Se ejecuta 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 no `sh` está presente en la imagen de compilación, asegúrate de incluirla antes de iniciar más compilaciones que usen tu 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 e inicie la compilación de nuevo. Para obtener más información, consulte Configuración del proyecto.

Acceder a GitHub los metadatos de las 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.

Error: «Sus credenciales carecen de uno o más ámbitos de privilegio obligatorios» al crear un proyecto con CodeBuild AWS CodeConnections

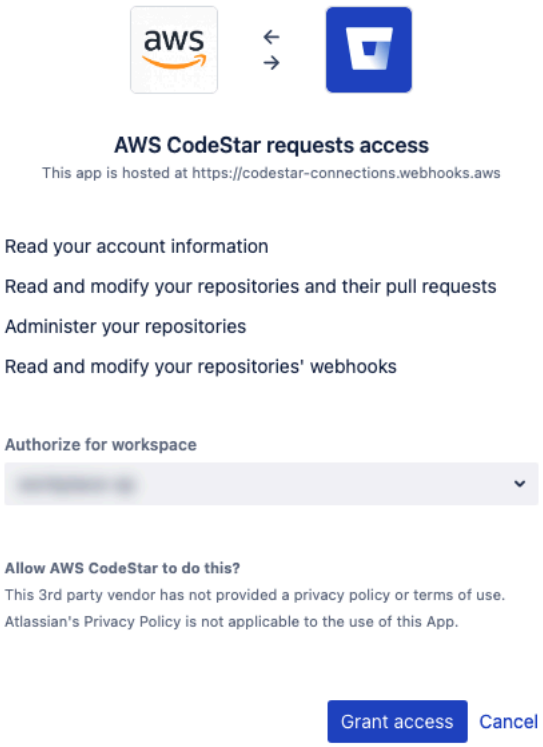
Problema: Al crear un CodeBuild proyecto con AWS CodeConnections, no tienes permiso para instalar un webhook de Bitbucket.

Causas posibles:

- Es posible que su cuenta de Bitbucket no haya aceptado el nuevo ámbito de permisos.

Soluciones recomendadas:

- Para aceptar el nuevo permiso, deberías haber recibido un correo electrónico de Bitbucket con el asunto Action required: scope for AWS CodeStar have change, enviado por Bitbucket. `notifications-noreply@bitbucket.org` El correo electrónico contiene un enlace para conceder al webhook permisos para instalar tu aplicación AWS CodeConnections Bitbucket actual.
- Si no encuentra el correo electrónico, puede conceder el permiso navegando hasta `https://bitbucket.org/site/addons/reauthorize?account=<workspace-name>&addon_key=aws-codestar` o `https://bitbucket.org/site/addons/reauthorize?addon_key=aws-codestar` y seleccionando el espacio de trabajo al que desea conceder el permiso de webhook.



Error: “Sorry, no terminal at all requested - can't get input” al compilar con el comando de instalación de Ubuntu

Problema: Si ejecuta compilaciones con privilegios de contenedores de GPU, es posible que esté instalando el kit de herramientas de contenedores de NVIDIA siguiendo este procedimiento. En la última versión de la CodeBuild imagen, CodeBuild preinstala y configura docker `nvidia-container-toolkit` con la imagen más reciente y seleccionada. `amazonlinux` `ubuntu` Si sigue este procedimiento, las compilaciones con el comando de instalación de Ubuntu fallarán y se generará el siguiente error:

```
Running command curl -fsSL https://nvidia.github.io/libnvidia-container/gpgkey | gpg --
dearmor --no-tty -o /usr/share/keyrings/nvidia-container-toolkit-keyring.gpg
gpg: Sorry, no terminal at all requested - can't get input
curl: (23) Failed writing body
```

Causas posibles: La clave gpg ya existe en la misma ubicación.

Soluciones recomendadas: El `nvidia-container-toolkit` ya está instalado en la imagen. Si ve este error, puede omitir el proceso de instalación y reinicio de docker en su buildspec.

Las compilaciones de Bitbucket fallan debido a errores de autenticación tras usar Secrets Manager OAuth

Problema: las compilaciones que utilizan Bitbucket como proveedor de fuentes fallan y se produce un error en las credenciales de origen. La conexión de Bitbucket se configuró con OAuth como servicio de almacenamiento. AWS Secrets Manager

Causa posible: Atlassian impone el uso de tokens de actualización rotativos de un solo uso para Bitbucket OAuth. Cada vez que utilizas un token de actualización, Bitbucket lo invalida y devuelve uno nuevo. Para obtener más información, consulta los tokens de actualización OAuth de un solo uso de Bitbucket (CHANGE-3052) en el sitio web para desarrolladores de Atlassian.

Solución recomendada: añade el permiso a tu rol de servicio`secretsmanager:PutSecretValue`. CodeBuild Para obtener más información sobre el permiso de IAM necesario, consulteAcción Manager-stored obligatoria para las credenciales de Secrets.

Cuotas para AWS CodeBuild

En las tablas siguientes se muestran las cuotas actuales en AWS CodeBuild. Estas cuotas son para cada AWS región compatible de cada AWS cuenta, a menos que se especifique lo contrario.

Cuotas de servicio

Las cuotas predeterminadas del AWS CodeBuild servicio son las siguientes.

Name	Predeterminado	Ajusta	Description (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: 2160	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 cualquier momento mediante la AWS CLI o un AWS SDK.
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 en cualquier momento mediante la AWS CLI o un AWS SDK.

Name	Predeterminado	Ajusta	Description (Descripción)
Compilaciones en ejecución simultánea para el entorno ARM Lambda/10GB	Cada región admitida: 1	Sí	Número máximo de compilaciones que se ejecutan simultáne amente para el entorno ARM Lambda/10GB
Compilaciones en ejecución simultánea para el entorno ARM Lambda/1GB	Cada región admitida: 1	Sí	Número máximo de compilaciones que se ejecutan simultáne amente para el entorno ARM Lambda/1GB
Compilaciones en ejecución simultánea para el entorno ARM Lambda/2GB	Cada región admitida: 1	Sí	Número máximo de compilaciones que se ejecutan simultáne amente para el entorno ARM Lambda/2GB
Compilaciones en ejecución simultánea para el entorno ARM Lambda/4GB	Cada región admitida: 1	Sí	Número máximo de compilaciones que se ejecutan simultáne amente para el entorno ARM Lambda/4GB
Compilaciones en ejecución simultánea para el entorno ARM Lambda/8GB	Cada región admitida: 1	Sí	Número máximo de compilaciones que se ejecutan simultáne amente para el entorno ARM Lambda/8GB
Compilaciones en ejecución simultánea para el entorno ARM/2XLarge	Cada región admitida: 1	Sí	Número máximo de compilaciones que se ejecutan simultáne amente para el entorno ARM/2XLarge

Name	Predeterminado	Ajusta	Description (Descripción)
Compilaciones para el entorno que se ejecutan simultáneamente ARM/Large	Cada región admitida: 1	Sí	Número máximo de compilaciones que se ejecutan simultáneamente para el entorno ARM/Large
Compilaciones para el entorno que se ejecutan simultáneamente ARM/Medium	Cada región admitida: 1	Sí	Número máximo de compilaciones que se ejecutan simultáneamente para el entorno ARM/Medium
Compilaciones para el entorno que se ejecutan simultáneamente ARM/Small	Cada región admitida: 1	Sí	Número máximo de compilaciones que se ejecutan simultáneamente para el entorno ARM/Small
Compilaciones para el entorno que se ejecutan simultáneamente ARM/XLarge	Cada región admitida: 1	Sí	Número máximo de compilaciones que se ejecutan simultáneamente para el entorno ARM/XLarge
Compilaciones en ejecución simultánea en un entorno con GPU Linux grande	Cada región admitida: 0	Sí	Número máximo de compilaciones que se ejecutan simultáneamente para un entorno Linux GPU/Large
Compilaciones en ejecución simultánea en un entorno con GPU Linux pequeño	Cada región admitida: 0	Sí	Número máximo de compilaciones que se ejecutan simultáneamente para un entorno Linux GPU/Small

Name	Predeterminado	Ajusta	Description (Descripción)
Compilaciones en ejecución simultánea para un entorno Linux Lambda/10GB	Cada región admitida: 1	Sí	Número máximo de compilaciones que se ejecutan simultáneamente para un entorno Linux Lambda/10GB
Compilaciones en ejecución simultánea para un entorno Linux Lambda/1GB	Cada región admitida: 1	Sí	Número máximo de compilaciones que se ejecutan simultáneamente para un entorno Linux Lambda/1GB
Compilaciones en ejecución simultánea para un entorno Linux Lambda/2GB	Cada región admitida: 1	Sí	Número máximo de compilaciones que se ejecutan simultáneamente para un entorno Linux Lambda/2GB
Compilaciones en ejecución simultánea para un entorno Linux Lambda/4GB	Cada región admitida: 1	Sí	Número máximo de compilaciones que se ejecutan simultáneamente para un entorno Linux Lambda/4GB
Compilaciones en ejecución simultánea para un entorno Linux Lambda/8GB	Cada región admitida: 1	Sí	Número máximo de compilaciones que se ejecutan simultáneamente para un entorno Linux Lambda/8GB
Compilaciones para el entorno que se ejecutan simultáneamente Linux/2XLarge	Cada región admitida: 0	Sí	Número máximo de compilaciones que se ejecutan simultáneamente para el entorno Linux/2XLarge

Name	Predeterminado	Ajusta	Description (Descripción)
Compilaciones para el entorno que se ejecutan simultáneamente Linux/Large	Cada región admitida: 1	Sí	Número máximo de compilaciones que se ejecutan simultáneamente para el entorno Linux/Large
Compilaciones para el entorno que se ejecutan simultáneamente Linux/Medium	Cada región admitida: 1	Sí	Número máximo de compilaciones que se ejecutan simultáneamente para el entorno Linux/Medium
Compilaciones para el entorno que se ejecutan simultáneamente Linux/Small	Cada región admitida: 1	Sí	Número máximo de compilaciones que se ejecutan simultáneamente para el entorno Linux/Small
Compilaciones para el entorno que se ejecutan simultáneamente Linux/XLarge	Cada región admitida: 1	Sí	Número máximo de compilaciones que se ejecutan simultáneamente para el entorno Linux/XLarge
Compilaciones en ejecución simultánea para el entorno Windows Server 2019/Large	Cada región admitida: 1	Sí	Número máximo de compilaciones que se ejecutan simultáneamente para el entorno Windows Server 2019/Large

Name	Predeterminado	Ajusta	Description (Descripción)
Compilaciones en ejecución simultánea para el entorno Windows Server 2019/ Medium	Cada región admitida: 1	Sí	Número máximo de compilaciones que se ejecutan simultáne amente para el entorno Windows Server 2019/ Medium
Compilaciones en ejecución simultáne a para el entorno Windows Server 2022/2XLarge	Cada región admitida: 1	Sí	Número máximo de compilaciones que se ejecutan simultáne amente para el entorno Windows Server 2022/2XLarge
Compilaciones en ejecución simultánea para el entorno Windows Server 2022/ Large	Cada región admitida: 1	Sí	Número máximo de compilaciones que se ejecutan simultáne amente para el entorno Windows Server 2022/ Large
Compilaciones en ejecución simultánea para el entorno Windows Server 2022/ Medium	Cada región admitida: 1	Sí	Número máximo de compilaciones que se ejecutan simultáne amente para el entorno Windows Server 2022/ Medium
Compilaciones en ejecución simultánea para el entorno Windows Server 2022/ XLarge	Cada región admitida: 1	Sí	Número máximo de compilaciones que se ejecutan simultáne amente para el entorno Windows Server 2022/ XLarge

Name	Predeterminado	Ajusta	Description (Descripción)
Compilaciones en ejecución simultánea para el entorno Windows/Large	Cada región admitida: 1	Sí	Número máximo de compilaciones que se ejecutan simultáne amente para el entorno Windows/Large
Compilaciones para el entorno que se ejecutan simultáneamente Windows/M edium	Cada región admitida: 1	Sí	Número máximo de compilaciones que se ejecutan simultáne amente para el entorno Windows/Medium
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 configura ción 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

Recurso	Predeterminado
Caracteres permitidos en una descripción de proyecto de compilación	Cualquiera
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 150 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 2160 (36 horas)

Compilaciones

Recurso	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 2160 (36 horas)

Flotas de computación

Recurso	Predeterminado
Número simultáneo de flotas de computación	10
Instancias en ejecución simultánea para flotas de entornos ARM/Small	1
Instancias en ejecución simultánea para flotas de entornos ARM/Large	1
Instancias en ejecución simultánea para flotas de entornos Linux/Small	1
Instancias en ejecución simultánea para flotas de entornos Linux/Medium	1
Instancias en ejecución simultánea para flotas de entornos Linux/Large	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 Linux GPU/Small	0
Instancias en ejecución simultánea para flotas de entornos Linux GPU/Large	0
Instancias en ejecución simultánea para flotas de entornos Windows Server 2019/Medium	1
Instancias en ejecución simultánea para flotas de entornos Windows Server 2019/Large	1

Recurso	Predeterminado
Instancias en ejecución simultánea para flotas de entornos Windows Server 2022/Medium	1
Instancias en ejecución simultánea para flotas de entornos Windows Server 2022/Large	1
Instancias en ejecución simultánea para flotas de entornos Mac ARM/Medium	1
Instancias en ejecución simultánea para flotas de entornos Mac ARM/Large	1

Informes

Recurso	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 AWS	5 000
Número máximo de casos de prueba por informe	500

Tags

Los límites de etiquetas se aplican a las etiquetas de los recursos de los proyectos de CodeBuild construcción y de los grupos de CodeBuild informes.

Recurso	Predeterminado
Nombres de claves de etiquetas de recursos	Cualquier combinación de letras, números, espacios y caracteres permitidos de Unicode con una longitud de UTF-8 entre 1 y 127 caracteres. Los caracteres permitidos son `+ - = . _ : / @` 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: • comenzar por `aws:` • contener únicamente espacios • terminar con un espacio • contener emojis ni ninguno de los siguiente s caracteres: `? ^ * [ \ ~ ! # $ % & * ( ) > < | " ' ` [ ] { } ;`
Valores de las etiquetas de recursos	Cualquier combinación de letras Unicode, números, espacios y caracteres permitidos de UTF-8 entre 0 y 255 caracteres de longitud. Los caracteres permitidos son `+ - = . _ : / @` 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: `? ^ * [ \ ~ ! # $ % & * ( ) > < | " ' ` [ ] { } ;`

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: tiempos de ejecución disponibles para imágenes CodeBuild estándar	Se agregaron Node.js 26 Runtime y Rust 1.97 para imágenes estándar x86_64.	22 de septiembre de 2026
Contenido actualizado: tiempos de ejecución disponibles para imágenes CodeBuild estándar	Se agregó la imagen AArch64 standard:4.0 de Amazon Linux 2023 a los tiempos de ejecución compatibles.	22 de septiembre de 2026
Contenido actualizado: sintaxis de expresiones regulares del filtro de webhooks	Se aclaró que los patrones de filtro de webhooks utilizan la sintaxis de expresiones regulares RE2 y qué construcciones no son compatibles.	3 de agosto de 2026
Contenido actualizado: Imágenes de computación de Lambda	Se agregaron imágenes de computación Lambda para .NET 10Go 1.25,Corretto 25,Node.js 24, yPython 3.14.	29 de julio de 2026
Contenido actualizado: Contraseña, token de API o token de acceso de la aplicación Bitbucket	Se agregó compatibilidad con los tokens de API de Bitbucket, que sustituyen a las	19 de julio de 2026

	contraseñas de aplicaciones obsoletas.	
Asignación personalizada de roles GitHub empresariales y corrección de nombres de roles MAINTAIN	Se agregó documentación sobre cómo las funciones de los repositorios personalizados de GitHub Enterprise se asignan a las funciones estándar en función de la jerarquía de permisos. Se ha corregido el nombre del rol de nivel de mantenimiento para que aparezca `GITHUB_MAINTAIN` en los valores y ejemplos predeterminados.	13 de abril de 2026
Contenido actualizado: tiempos de ejecución disponibles para imágenes CodeBuild estándar	Se han actualizado las versiones de tiempo de ejecución disponibles y se han añadido nuevas imágenes.	7 de abril de 2026
Contenido actualizado: Imágenes de computación de EC2	Se agregaron las imágenes de computación Amazon Linux 2023 x86_64 standard:6.0 y Ubuntu 24.04:8.0 compute images.	7 de abril de 2026
Contenido actualizado: comportamiento de selección de versiones en tiempo de ejecución	Se agregó una guía que indica que las versiones predeterminadas en tiempo de ejecución pueden cambiar cuando una versión predeterminada anterior llega al final de su vida útil (EOL).	25 de marzo de 2026

Contenido actualizado: tiempos de ejecución disponibles para imágenes CodeBuild estándar	Se actualizaron las versiones de tiempo de ejecución disponibles y se agregaron nuevas imágenes.	25 de marzo de 2026
Contenido actualizado: políticas AWS administradas (predefinidas) para AWS CodeBuild	Las AWSCodeBuildDeveloperAccess políticas AWSCodeBuildAdminAccess y se han actualizado para incluir el `ssmmessages:OpenDataChannel` permiso necesario para la depuración interactiva de compilaciones de Session Manager.	1 de diciembre de 2025
Aprobación de comentarios de solicitudes de extracción	CodeBuild admite políticas de creación de solicitudes de incorporación de cambios que proporcionan un control adicional sobre las compilaciones activadas por solicitudes de incorporación de cambios.	7 de agosto de 2025
Contenido actualizado: políticas AWS gestionadas (predefinidas) para AWS CodeBuild	Las AWSCodeBuildReadOnlyAccess políticas AWSCodeBuildAdminAccess AWSCodeBuildDeveloperAccess, y se han actualizado. El recurso original `arn:aws:codeconnections:*:*:connection/*` se ha actualizado a `arn:aws:codeconnections:*:*:*` .	10 de junio de 2025

Nueva referencia para las claves de CodeBuild estado	Se agregó una nueva página de referencia con ejemplos de uso de claves de CodeBuild condición. Consulte Claves de condición de AWS CodeBuild.	15 de mayo de 2025
Contenido nuevo: Ejemplo de servidor de creación de imágenes Docker para CodeBuild	CodeBuild ahora permite transferir tus compilaciones de Docker a un servidor de creación de imágenes administrado.	15 de mayo de 2025
Nuevo tipo de computación: `CUSTOM_INSTANCE_TYPE`	CodeBuild ahora te permite crear una flota de capacidad reservada con un tipo de instancia específico mediante el uso de. `CUSTOM_INSTANCE_TYPE`	23 de abril de 2025
Nuevo soporte para CodeBuild sandbox	Se ha añadido información sobre el uso del nuevo CodeBuild sandbox. Consulte Depurar compilaciones con CodeBuild sandbox.	7 de abril de 2025
Nuevos tipos de entornos de Windows	CodeBuild ahora es compatible con los tipos de entorno Windows XL y 2XL. Para obtener más información, consulte Tipos de computación del entorno de compilación.	31 de marzo de 2025
Se ha actualizado el almacenamiento en caché de Amazon S3	CodeBuild ahora admite un nuevo comportamiento de almacenamiento en caché para el almacenamiento en caché de Amazon S3.	28 de marzo de 2025

Contenido nuevo: opciones de configuración del GitHub Actions Runner	CodeBuild ahora admite `CODEBUILD_CONFIG_GITHUB_ACTIONS_ENTERPRISE_REGISTRATION_NAME` el registro a nivel empresarial.	11 de marzo de 2025
Contenido nuevo: Adición de un nuevo tipo de filtro de webhook	Agregue compatibilidad con un nuevo tipo de filtro de webhook (`ORGANIZATION_NAME`).	11 de marzo de 2025
Nuevo contenido: Tutorial de firma de código de Apple con Fastlane mediante S3 para almacenamiento de certificados	Se ha añadido un nuevo tutorial para la firma de código de Apple con Fastlane sobre el CodeBuild uso de S3 para el almacenamiento de certificados	5 de febrero de 2025
Contenido nuevo: tutorial para la firma de código de Apple con Fastlane con GitHub almacenamiento de certificados	Se ha añadido un nuevo tutorial para la firma de código de Apple con Fastlane y su CodeBuild uso GitHub para el almacenamiento de certificados	5 de febrero de 2025
Nuevo contenido: Ejecutor de Buildkite	Agregue nuevo contenido para el ejecutor de Buildkite	31 de enero de 2025
Nuevo contenido: Webhooks manuales de Buildkite	Agregue compatibilidad para webhooks manuales de Buildkite.	31 de enero de 2025
Nuevo contenido: referencia de buildspec de compilación por lotes	Agregue compatibilidad para compilaciones por lotes en flotas de capacidad reservada y entornos Lambda.	8 de enero de 2025

Nuevo contenido: Ejecución de pruebas paralelas en compilaciones por lotes	Agregue nuevo contenido para pruebas paralelas en compilaciones por lotes.	2 de enero de 2025
Contenido nuevo: Cómo reintentar hacer una compilación de forma automática	CodeBuild ahora admite el reintento automático para compilaciones de webhooks.	18 de diciembre de 2024
Nuevo contenido: Configuración de una credencial de registro privada para ejecutores autoalojados	Agregue compatibilidad con la configuración de credenciales de registro utilizando imágenes personalizadas de registros no privados.	13 de diciembre de 2024
Contenido nuevo: opciones de configuración del GitHub Actions Runner	CodeBuild GitHub Los corredores autohospedados de Actions ahora te permiten registrar a tus corredores a nivel de organización y configurar un ID de grupo de corredores específico.	12 de diciembre de 2024
Nuevo contenido: Adición de un atributo RETRY en caso de fallo	CodeBuild ahora te permite configurar un atributo en caso de error para incluirlo RETRY en tus especificaciones de compilación.	12 de diciembre de 2024
Nuevo contenido: webhooks manuales GitLab	Añade soporte para GitLab webhooks manuales.	11 de diciembre de 2024
Contenido actualizado: actualización de alias	Actualiza los alias de las imágenes en tiempo de ejecución Linux-based estándar.	22 de noviembre de 2024

Contenido actualizado: el ejecutor admite la anulación de etiquetas CodeBuild-hosted GitLab	Se ha añadido la posibilidad de anular etiquetas de imagen personalizadas para GitLab los corredores.	22 de noviembre de 2024
Contenido actualizado: el corredor de acciones admite la anulación de CodeBuild-hosted GitHub etiquetas	Se ha añadido la posibilidad de anular etiquetas de imagen personalizadas para los ejecutores de GitHub Actions.	22 de noviembre de 2024
Contenido actualizado: políticas AWS administradas (predefinidas) para AWS CodeBuild	Las AWSCodeBuildReadOnlyAccess políticas AWSCodeBuildAdminAccess AWSCodeBuildDeveloperAccess, y se han actualizado. El recurso original `arn:aws:codebuild:*` se ha actualizado a `arn:aws:codebuild:*:*:project/*` .	15 de noviembre de 2024
Contenido actualizado: Capacidad reservada	Las flotas de capacidad reservada admiten ahora compilaciones que no son de contenedores: ARM EC2, Linux EC2 y Windows EC2.	12 de noviembre de 2024
Contenido actualizado: Capacidad reservada	Las flotas de capacidad reservada son ahora compatibles con computación basada en atributos.	6 de noviembre de 2024
Contenido nuevo: Cómo reintentar hacer una compilación de forma automática	CodeBuild ahora te permite habilitar el reintento automático para tus compilaciones.	25 de octubre de 2024

Contenido nuevo: se ejecuta CodeBuild en un servidor proxy administrado para flotas con capacidad reservada	Agregue compatibilidad con las configuraciones de proxy para las flotas de capacidad reservada.	15 de octubre de 2024
Contenido nuevo: corredores Self-managed GitLab	Añade contenido nuevo para corredores autogestionados GitLab	17 de septiembre de 2024
Nuevo contenido: webhooks GitLab grupales	Añade soporte para webhooks GitLab grupales.	17 de septiembre de 2024
Contenido nuevo: ejecuta los comandos buildspec en las fases INSTALL, PRE_BUILD y POST_BUILD	Agregue compatibilidad con `-with-buildspec` .	20 de agosto de 2024
Contenido actualizado: Capacidad reservada	Las flotas de capacidad reservada ahora son compatibles con macOS.	19 de agosto de 2024
Contenido nuevo: GitHub conexiones de aplicaciones	Añade soporte para las conexiones de GitHub aplicaciones.	14 de agosto de 2024
Contenido nuevo: Conexiones de aplicación de Bitbucket	Agregue compatibilidad con las conexiones de aplicación de Bitbucket.	14 de agosto de 2024
Contenido nuevo: Tokens de acceso múltiple en CodeBuild	Añada la posibilidad de suministrar tokens de acceso a proveedores externos a partir de los secretos de AWS CodeConnections las conexiones AWS Secrets Manager o a través de ellas.	14 de agosto de 2024

Contenido actualizado: Capacidad reservada	Las flotas de capacidad reservada ahora son compatibles con los tipos de computación ARM Medium, ARM XLarge y ARM 2XLarge.	5 de agosto de 2024
Contenido actualizado: Capacidad reservada	CodeBuild ahora admite la conectividad de VPC para flotas de capacidad reservada en Windows.	1 de agosto de 2024
Nuevos tipos de computación de ARM	CodeBuild ahora admite los tipos de procesamiento ARM Medium, ARM XLarge y ARM 2xLarge. Para obtener más información, consulte Tipos de computación del entorno de compilación.	10 de julio de 2024
Contenido actualizado: Firma SHA	Actualice la firma del algoritmo hash seguro (SHA) para x86_64 y ARM.	19 de junio de 2024
Contenido nuevo: webhooks GitHub globales y organizacionales	Añade soporte para GitHub webhooks globales y organizacionales.	17 de junio de 2024
Contenido nuevo: Adición de un nuevo tipo de filtro de webhook	Agregue compatibilidad con un nuevo tipo de filtro de webhook (`REPOSITORY_NAME`).	17 de junio de 2024
Espacio en disco actualizado	Ahora, los tipos de computación `ARM Small` y `ARM Large` tienen más espacio en disco.	4 de junio de 2024
Contenido nuevo: webhooks GitHub manuales	Añade soporte para GitHub webhooks manuales.	23 de mayo de 2024

Contenido actualizado: Capacidad reservada	CodeBuild ahora admite la conectividad de VPC para flotas de capacidad reservada en Amazon Linux.	15 de mayo de 2024
Contenido actualizado: Imágenes de computación de Lambda	Adición de compatibilidad con Lambda para .NET 8 (`al-lambda/aarch64/dotnet8` y `al-lambda/x86_64/dotnet8`)	8 de mayo de 2024
Cuota actualizada: Tiempo de espera de compilación	Actualice la cuota máxima de tiempo de espera de compilación a 2160 minutos (36 horas).	1 de mayo de 2024
Contenido actualizado: políticas AWS administradas (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, token de API o token de acceso de la aplicación Bitbucket	Agregue compatibilidad con los tokens de acceso de Bitbucket.	11 de abril de 2024
Contenido nuevo: Auto-discover informes en CodeBuild	CodeBuild ahora admite la detección automática de informes.	4 de abril de 2024
Nuevo contenido: Self-hosted GitHub Actions Runners	Añade contenido nuevo para los ejecutores de GitHub Actions autohospedados	2 de abril de 2024

Contenido nuevo: conexiones GitLab	Añada soporte para GitLab conexiones GitHub autogestionadas.	25 de marzo de 2024
Contenido nuevo: Adición de nuevos eventos y tipos de filtro de webhook	Agregue compatibilidad con nuevos eventos (`RELEASED` y `PRERELEASED`) y tipos de filtros (`TAG_NAME` y `RELEASE_NAME`) de webhook.	15 de marzo de 2024
Contenido nuevo: Adición de un nuevo evento de webhook: `PULL_REQUEST_CLOSED`	Adición de soporte para un nuevo evento de webhook: `PULL_REQUEST_CLOSED` .	20 de febrero de 2024
Contenido actualizado: Imágenes de Docker proporcionadas por CodeBuild	Adición de compatibilidad para Windows Server Core 2019 (`windows-base:2019-3.0`)	7 de febrero de 2024
Contenido actualizado: Imágenes de Docker proporcionadas por CodeBuild	Adición de compatibilidad con nuevos tiempos de ejecución para Amazon Linux 2023 (`al2/aarch64/standard/3.0`)	29 de enero de 2024
Contenido nuevo: 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 el tipo de procesamiento 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 (`al2/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
Nuevo contenido: computación AWS Lambda	Agregue contenido nuevo para el AWS Lambda procesamiento	6 de noviembre de 2023
Contenido actualizado: Imágenes de Docker proporcionadas por CodeBuild	Se ha añadido la compatibilidad con Amazon Linux 2 (`al2/standard/5.0`)	17 de mayo de 2023
Cambios en las políticas administradas para CodeBuild	Los detalles sobre las actualizaciones de las políticas AWS administradas para ya CodeBuild están disponibles. Para obtener más información, consulte Actualizaciones de CodeBuild a políticas administradas de AWS.	16 de mayo de 2023
Contenido actualizado: Imágenes de Docker proporcionadas por CodeBuild	Se ha eliminado la compatibilidad con Amazon Linux 2 (`al2/standard/3.0`) y se ha añadido la compatibilidad con Amazon Linux 2 (`al2/standard/corretto8`) y Amazon Linux 2 (`al2/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 (`al2/aarch64/standard/1.0`)	31 de marzo de 2023
Contenido actualizado: Eliminación de la limitación de VPC	Eliminar la siguiente limitación: si la configuras CodeBuild para que funcione con una VPC, no se admite el almacenamiento en caché local. A partir de 02/28 /22, la compilación de la 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 (`al2/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 proyecto específico. CodeBuild 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 la configura CodeBuild para que funcione con una VPC, no se admite el almacenamiento en caché local. A partir de 02/28 /22, la compilación de la 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 compilación 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 procesamiento 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 construcción 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 te permite limitar el número de compilaciones simultáneas para un proyecto de compilación. Para obtener más información, consulta Configuración y BuildLimit concurrencia del proyecto.	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 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 se produce un error en una fase de compilación. 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 te permite excluir rutas de tus artefactos de compilación. Para obtener más información, consulte exclude-paths.	9 de febrero de 2021
Nueva propiedad de especificación de compilación: enable-symlinks	CodeBuild ahora proporciona la propiedad `enable-symlinks` buildspec 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 `artifacts/name` propiedad contenga información de ruta. Para obtener más información, consulte name.	9 de febrero de 2021

Informes de cobertura de código	CodeBuild ahora proporcio na 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 de compilaciones simultáneas y coordinadas de un proyecto. Para obtener más información, consulta Creación por lotes. 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 informaci ón, consulte las imágenes de Docker proporcionadas por CodeBuild.	20 de julio de 2020
Session Manager	CodeBuild ahora te permite pausar una compilación en ejecución y luego usar el Administrador de AWS Systems Manager sesiones para conectarte al contenedor de compilación y ver el estado del contenedor. 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 los informes de pruebas ya está disponible de forma general.	21 de mayo de 2020

Temas actualizados	CodeBuild ahora permite crear filtros de creación de webhooks para Github y Bitbucket que activen compilaciones solo cuando el mensaje de confirmación principal coincide con la expresión especificada. Para obtener más información, consulta el ejemplo de solicitud de GitHub extracción y el filtro de webhooks y el ejemplo de solicitud de extracción y filtro de webhooks de Bitbucket.	6 de mayo de 2020
Temas nuevos	CodeBuild ahora permite compartir recursos de grupos de informes y proyectos de creación. 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, consulta Cómo trabajar con informes de pruebas, Crear un informe de prueba y Crear un informe de prueba con el AWS CLI ejemplo.	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` procesamiento. Para obtener más información, consulte Tipos de computación 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 la versión 29 de Android y la versión 1.13 de Go. Para obtener más información, consulte Imágenes de Docker proporcionadas por CodeBuild y Sintaxis de la especificación de compilación.	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 Imágenes de Docker proporcionadas por CodeBuild y Versiones del entorno de ejecución en el ejemplo del archivo buildspec de CodeBuild.	16 de agosto de 2019
Tema actualizado	Al crear un proyecto, ahora puedes optar por deshabilitar el cifrado de los registros de S3 y, si utilizas un repositorio de Git-based origen, incluir los submódulos de Git. Para obtener más información, consulta 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 especific ar uno de los cuatro modos de almacenamiento en caché local. Para obtener más información, consulte Crear almacenamiento en caché en. CodeBuild	21 de febrero de 2019
Temas nuevos	CodeBuild ahora admite grupos de filtros de webhooks para especificar los eventos que activan una compilación. Para obtener más informaci ón, consulta Filtrar eventos de GitHub webhooks y Filtrar eventos de webhooks de Bitbucket.	8 de febrero de 2019
Nuevo tema	La Guía CodeBuild del usuario ahora muestra cómo usarlo 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 el ejemplo de Amazon ECR para CodeBuild, la creación de un proyecto de compilación y la creación de 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, consulte Ejemplo de registro privado con AWS Secrets Manager.	24 de enero de 2019
Tema actualizado	CodeBuild ahora admite el uso de un token de acceso para conectarse a los GitHub repositorios (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 métricas de compilación nuevas que miden la duración de cada fase de una compilación. Para obtener más información, consulte Métricas de CodeBuild CloudWatch.	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 de enlace de la 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 de 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 usas Bitbucket como repositorio. Para obtener más información, consulte el ejemplo de solicitud de extracción de Bitbucket para CodeBuild.	2 de octubre de 2018

Registros de S3	CodeBuild ahora admite los registros de compilación en un bucket de S3. Anteriorm ente, solo se podían crear CloudWatch registros con Logs. Para obtener más información, consulte Creación de un proyecto.	17 de septiembre de 2018
Múltiples fuentes de entrada y múltiples artefactos de salida	CodeBuild ahora admite proyectos que utilizan más de una fuente de entrada y publican más de un conjunto de artefactos. Para obtener más información, consulte Ejemplo de varias fuentes de entrada y artefactos de entrada y un ejemplo de CodePipeline integración 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 ahora incluye un ejemplo basado en casos de uso que muestra cómo usar 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 ahora incluye un ejemplo basado en casos de uso que muestra cómo alojar la salida de 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 asignar nombres a los artefactos de compilación. Esto resulta útil porque un artefacto de compilación con un nombre de codificación fija sobrescri be artefactos de compilaci ón 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
Soporte para CloudWatch métricas y alarmas de Amazon	CodeBuild ahora ofrece integración con CloudWatch métricas y alarmas. Puede usar la CloudWatch consola CodeBuild o la consola 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 sobre el estado del inicio y la finalización de una compilación a tu proveedor de origen. Para obtener más información, consulta Crear un proyecto de compilación en CodeBuild.	10 de julio de 2018
Variables de entorno agregadas a CodeBuild la 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 especific aciones de construcción. Los comandos del bloque finally siempre se ejecutan después de los comandos de su bloque de comandos correspondiente. Para obtener más informaci ón, 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 usar Amazon SNS para recibir notificaciones cuando se publiquen nuevas versiones del CodeBuild agente. Para obtener más información, consulte Recibir notificaciones de 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 compilaci ones 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 preempaqu etado para .NET Core 2.0 en Windows. Para obtener más información, consulte Ejecute ejemplos de Microsoft Windows para CodeBuild .	
Compatibilidad con la idempotencia de la compilaci ón	Cuando ejecutas el `start-build` comando con AWS Command Line Interface (AWS CLI), puedes especific ar que la compilación es idempotente. Para obtener más información, consulte Ejecute 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 anulacion es solo se aplican a esa compilación. Para obtener más información, consulte Ejecute AWS CodeBuild compilaciones manualmente .	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 Crea activadores AWS CodeBuild.	28 de marzo de 2018
Documentación de los puntos de enlace de FIPS	Ahora puede obtener información sobre cómo usar AWS Command Line Interface (AWS CLI) o un AWS SDK para indicar el uso de uno de los cuatro puntos finales de CodeBuild los Estándares Federales de Procesamiento de la Información (FIPS). Para obtener más información, consulte Especifique el AWS CodeBuild punto de conexión .	28 de marzo de 2018
AWS CodeBuild disponible en Asia Pacífico (Bombay), Europa (París) y Sudamérica (São Paulo)	AWS CodeBuild ya 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 Ejecute el ejemplo de GitHub 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	VPC-enabled las compilaciones ahora pueden acceder a los recursos de tu 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 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) que se puede incrustar y se genera 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, consulteAWS Config muestra.	20 de octubre de 2017
Reconstruya automáticamente el código fuente actualizado en los GitHub repositorios	Si tu 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 Ejecuta el ejemplo del GitHub filtro de solicitudes de extracción 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 usar la AWS CodeBuild consola o la 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 utilizar la consola de AWS CodeBuild 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, Cambio de la configuración del proyecto de compilación y Ejecución de compilaciones de forma manual.	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 es más fácil usar una especific ación 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 ejecutand o comandos de compilación para automatizar la AWS CLI. Para obtener más informaci ón, consulte la asignación de `parameter-store` 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 informaci ón, consulte Creación de un proyecto de compilación y Ejecución de compilaciones de forma manual.	10 de agosto de 2017

Cambio	Descripción	Fecha
AWS CodeBuild disponible en el oeste de EE. UU. (norte de California), Europa (Londres) y Canadá (centro)	AWS CodeBuild ya está disponible en las regiones 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 `buildspec.yml` 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 agregado un ejemplo que muestra cómo usar 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
Obtén el código fuente de las solicitudes 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 compilarla. 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 Ejecute 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 soluciona el problema de CodeBuild ejecutar cada comando de compilación en una instancia independiente del shell predeterminado. También, en la versión 0.2, `environment_variables` ahora se llama `env` y `plaintext` ahora se llama `variables` . 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 proporcionadas por AWS CodeBuild están disponibles como Dockerfiles en. GitHub Para obtener más información, consulte la columna "Definition" 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 acción de prueba para CodeBuild	Ahora puedes añadir a una canalización CodePipeline una acción de prueba que utilice CodeBuild. Para obtener más información, consulte Añadir una acción CodeBuild de prueba a una canalización (CodePipeline consola) .	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 te permiten especificar directorios individuales de nivel superior cuyo contenido puedes indicar que se incluya en los artefactos de salida CodeBuild de compilación. Para ello utilice la asignación `base-directory` . Para obtener más información, consulte Sintaxis de buildspec .	8 de febrero de 2017

Cambio	Descripción	Fecha
Built-in variables de entorno	AWS CodeBuild proporciona variables de entorno integradas adicionales para que las utilicen 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 el este de EE. UU. (Ohio)	AWS CodeBuild ya está disponible en la región Este de EE. UU. (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 Resolución 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 AWS CodeBuild con Jenkins .	5 de diciembre de 2016
Versión inicial de la Guía de usuario	Esta es la versión inicial de la Guía de usuario de CodeBuild .	1 de diciembre de 2016

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